Corso di Laurea Magistrale in Psicobiologia e Neuroscienze Cognitive

DISPENSA DI

TECNICHE DI ANALISI DI DATI I

[fino al capitolo 7 compreso]

e

TECNICHE DI ANALISI DI DATI II

[dal capitolo 8 in poi]

a.a. 2022-2023

CALDA RACCOMANDAZIONE

Si raccomanda vivamente di **leggere la dispensa con R aperto** e di rifare **CONTESTUALMENTE** alla lettura **TUTTI** gli ESEMPI proposti.

ESTREMA raccomandazione

Si supplica di accompagnare alla lettura della dispensa e alla replica degli esempi l'esecuzione degli ESERCIZI proposti QUI E SU ELLY, nonché di farli correggere dal docente, SOPRATTUTTO in caso di difficoltà.

Sommario

Premessa
Cosa s'intende per "analisi di dati"?

Come suggerito dal titolo della dispensa e dal vostro piano di studio, dovrete affrontare due esami di Tecniche di Analisi di dati: è buona norma, quindi, definire in primis l'oggetto dei due insegnamenti, ovvero cosa si possa definire analisi di dati o data analysis.

Prendiamo la definizione di uno dei più importanti statistici del Novecento, che incontremo più volte nelle prossime pagine: secondo Tukey[1] (1966, The future of data analysis, pag.2: http://projecteuclid.org/download/pdf_1/euclid.aoms/1177704711), "data analysis [...] I take to include, among other things: **procedures for analyzing data**, **techniques for interpreting the results** of such procedures, **ways of planning the gathering of data** to make its analysis easier, more precise o more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data"

L'analisi di dati, quindi, si serve della statistica, ma non si esaurisce in essa.

[1] Oltre ad essere stato un grande statistico (lo ritroveremo nei test post hoc dell'analisi della varianza) e un pioniere dell'analisi visiva dei dati, a Tukey è attribuita la creazione dei termini "software" (1958) e "bit" (1947).

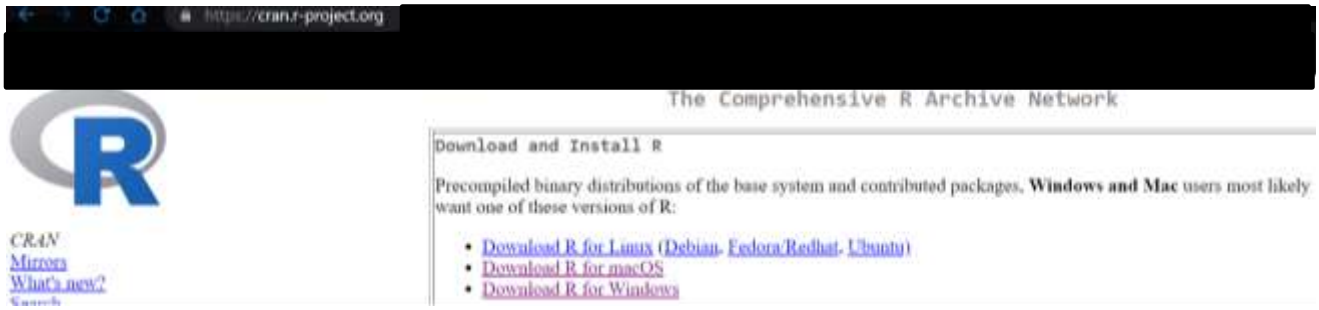
Capitolo 1
Presentazione dell'ambiente R

R è un ambiente per elaborazioni statistiche e rappresentazioni grafiche, gratuito: è un software *open source*, ovvero che non nasconde il codice su cui è basato ed anzi sollecita tutti gli elaboratori a contribuire al suo sviluppo, con nuove funzioni, nuove analisi, ecc. In questo corso, umilmente, ci limiteremo a usare al meglio le funzionalità sviluppate da altri, ma chiunque di voi ambisca a partecipare alla comunità di sviluppatori, sarà accolto a braccia aperte.
R è versatile ed estremamente dinamico: miriadi di utilizzatori e devoti del software sono attivissimi nel rilasciare sempre nuove funzionalità e correggere i propri ed altrui errori (bugs); molte delle cose che fa non sono (ancora) disponibili nei software di elaborazioni con licenza non free (ex., SPSS). Il principale svantaggio di R, invece, è sostanzialmente di "immagine": digitare linee di codice in una pagina bianca, invece di spostare il mouse e cliccare su una comoda finestrella utilizzando un'interfaccia utente grafica (**GUI**: graphic user interface), per ottenere un'analisi o un grafico, sembra quasi un passo indietro contro la modernità; in realtà, esistono almeno un paio di "trucchi" per alleviare l'impatto di questa modalità di inserimento dei comandi (li vedremo) e, una volta imparate ben poche cose di base sui comandi, i comandi scritti di R sono decisamente più efficienti.

R dispone di un pacchetto – base con tantissime funzionalità. Una volta scaricato R dal Web, nelle modalità che vedremo, e installato sul proprio dispositivo[2], si può iniziare a fare un sacco di analisi e a costruire molti grafici. Comunque, le funzionalità di R possono essere ampliate scaricando **package** ad hoc che aggiungono ulteriori funzioni (analisi, grafici, ...) al programma. Come accennato, chiunque, armato di buona volontà e buone competenze di programmazione, può creare un proprio package e metterlo a disposizione di tutti; chi è armato di buona volontà e medio-basse-nulle competenze di programmazione (cioè noi) potrà scaricare package che contengono le funzioni utili per le proprie analisi da un archivio dedicato. L'archivio si chiama **CRAN** (*Comprehensive R Archive Network*): contiene i file di sistema da scaricare per installare R (vedremo esattamente i passaggi), i package e un sacco di file di aiuto. L'archivio CRAN è *mirrored*: a fianco dell'archivio "centrale", sono **replicate identiche versioni dell'archivio** su server in tutto il mondo: quando si tratterà di scaricare cose dall'archivio, **sceglieremo il CRAN geograficamente più vicino** (per l'Italia, a Padova).

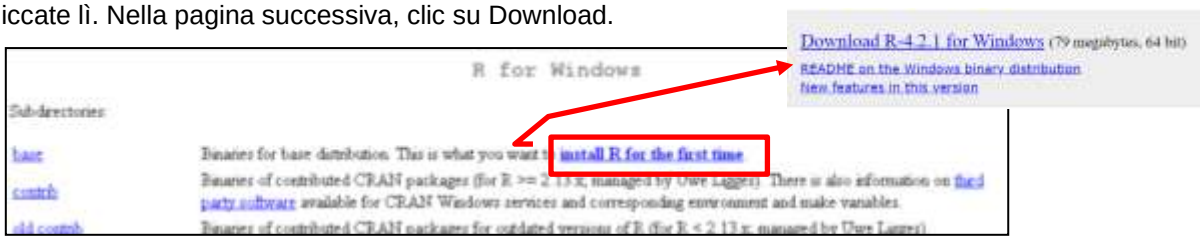
1.1 Scaricare e installare R

Per scaricare R, occorre una connessione internet: il sito del software è: **https://cran.r-project.org/**
Nell'home page, sono presenti i link per scaricare R a seconda del proprio sistema operativo: Windows, Mac, Linus.

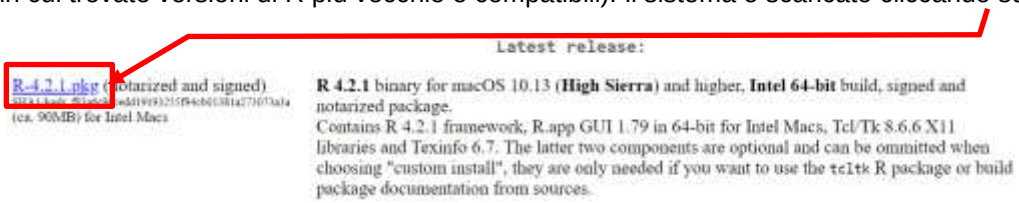


[2] si consiglia caldamente un PC/Mac, ma un buon tablet dovrebbe comunque gestirlo – anche se in maniera decisamente scomoda, dato che in R si usa la tastiera per digitare i comandi, che non è la funzionalità più ergonomica di un tablet.

a) Se usate un sistema Windows, la pagina cui siete condotti vi dice cosa scegliere se installate R **per la prima volta**: cliccate lì. Nella pagina successiva, clic su Download.



b) Se usate Mac, siete portati a una pagina più verbosa, ma l'azione è fondamentalmente la stessa (però, se avete versioni molto antiquate del sistema operativo, un *warning* a inizio pagina vi invita a recarvi in una diversa pagina in cui trovate versioni di R più vecchie e compatibili): il sistema è scaricato cliccando su **R-4.2.1.pkg**



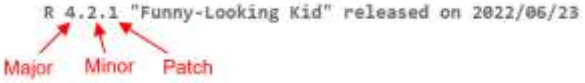
Attenzione: per funzionare su macOS, R richiede che abbiate installato sul vostro device **XQuartz**, programma che serve per varie applicazioni basate su Unix (tra cui, appunto, R), e che non è più installato insieme al sistema operativo macOS. Se non avete mai avuto occasione di scaricarlo prima, fatelo ora, come vi spiega il sito qualche riga sotto le precedenti: basta cliccare sul link nella pagina.

c) Linux è come sempre molto essenziale ☺ :

Una volta salvato il file .exe sul vostro dispositivo, installatelo secondo le solite procedure con cui installate qualsiasi *app* nel vostro specifico sistema.

1.2 Le versioni di R

R si evolve continuamente, e quindi molto spesso vengono rilasciati aggiornamenti. Le sue versioni sono identificate da un codice con tre numeri (e una qualifica testuale – spiritosa, ma talvolta davvero bizzarra):



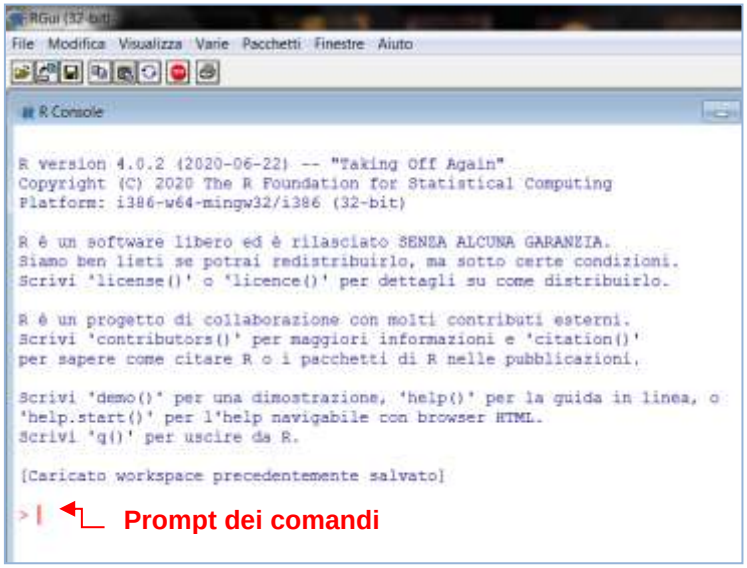
L'ultima cifra (**patch**) indica che in questa versione sono state apportate correzioni a *bugs* di scarsa rilevanza presenti nella precedente; cambia piuttosto frequentemente, ma, perlopiù, non è importante aggiornare la propria versione, a meno che non si intenda proprio usare la funzionalità che è stata corretta. La cifra intermedia (**minor version**) cambia meno spesso (circa un paio di volte all'anno), e indica che la nuova versione raccoglie diverse novità e correzioni. Infine, le variazioni delle versioni identificate dal primo numero (**major version**) sono decisamente più rare: R è passato dalla versione 3.6 all'attuale 4.2.1 ad agosto 2021. In genere, se state usando versioni di R troppo vecchie, è il sistema stesso che vi avvisa: un *warning* nel momento in cui richiedete una funzionalità ormai superata vi sollecita ad aggiornare la versione installata. Poco male: ogni nuova installazione di versioni successive di R, scaricabili come sopra descritto dall'archivio CRAN, affianca, e non sostituisce, le versioni più vecchie: per eliminare le vecchie versioni, dovete proprio fisicamente disinstallarle una a una dal vostro dispositivo.

1.3 Iniziare a usare R

Una volta scaricato e installato, R si apre come un qualsiasi altro programma.

1.3.1 La console

All'apertura, la pagina principale si presenta in maniera decisamente minimal (NB: la sottoscritta usa Windows, quindi la configurazione delle schermate incollate in questa dispensa si riferisce a questo sistema operativo; le differenze grafiche per gli utenti Mac o Linux sono comunque minime, e saranno evidenziate dove necessario)



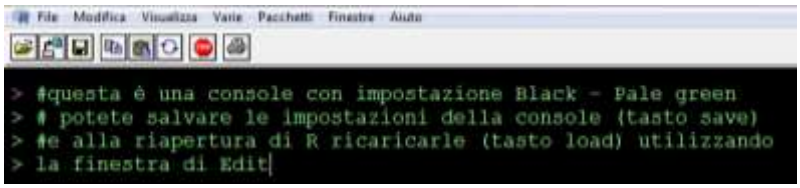
Questa finestra principale si chiama **console:** è qui che digitiamo i comandi e che vediamo i loro risultati. Il segno **>** indica che R è pronto a ricevere vostre istruzioni: si digita il comando e si preme **Invio** (**Enter**). Per esempio:

```
> 2+2
[1] 4
>
```

←

In blu il comando, in nero il suo output. Il numero tra [ ] indica la riga dell'output. Qui abbiamo una sola riga, ma vedremo funzioni complesse con output complessi e articolati su più righe. Una volta restituito l'output, il prompt dei comandi > si ripresenta, pronto per ulteriori funzioni

L'aspetto della Console si può modificare usando il menu **Modifica / Edit** → **Preferenze interfaccia** (Configuration GUI). Per esempio, potete cambiare lo stile e la dimensione del font, oppure i colori di sfondo e testo (di default background – white).



←

Come potete leggere nella figura precedente, se volete far precedere un comando (o seguire un output) da un commento o un promemoria, si fa precedere il testo da #: R interpreta come **commento** e non come comando qualunque cosa sia preceduto da #, indipendentemente dal fatto che sia un testo o un numero:

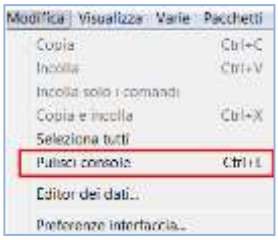
```
> #quanto fa 2+2?
> 2+2
[1] 4
> #la mia prima analisi in R!
```

Se si dimentica #, viene restituito come output un **error:** R si aspetta un comando, e non riconosce quanto scritto come una sua **funzione**:

```
> quanto fa 2+2?
Errore: unexpected symbol in "quanto fa"
```

I messaggi di errore di R, purtroppo, non sempre sono molto espliciti su quale sia esattamente l'errore: che ci sia qualcosa di sbagliato è sempre vero, ma spesso va individuato dall'utente cosa esattamente lo sia.
I comandi si seguono uno dopo l'altro nella pagina.

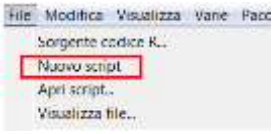
Se scrivete molti comandi inutili, o con diversi tentativi ed errori, e volete tornare ad avere una bella console pulita, potete usare la combinazione di tastiera **Crl+L**, o, nel menu Edit, scegliere **Pulisci Console** (**Clear console**: ma, attenzione: non si torna indietro):



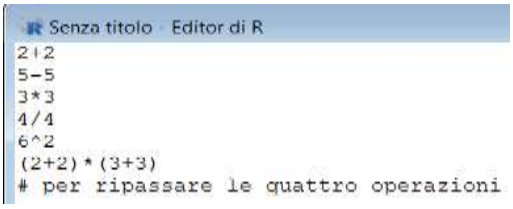
1.3.2 Gli script

Oltre alla finestra console, R ha un'importantissima altra finestra, quella degli **script**: in uno script, vengono scritte e salvate, per gli usi futuri, varie linee di comando.

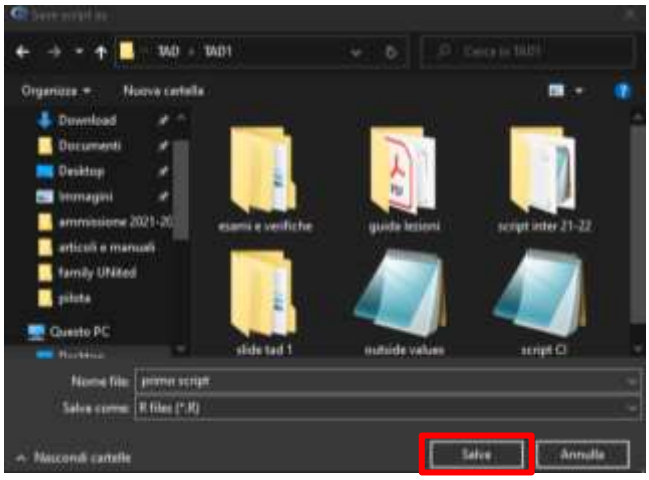
Per creare un **nuovo script**, selezionare `File` → `Nuovo script:`

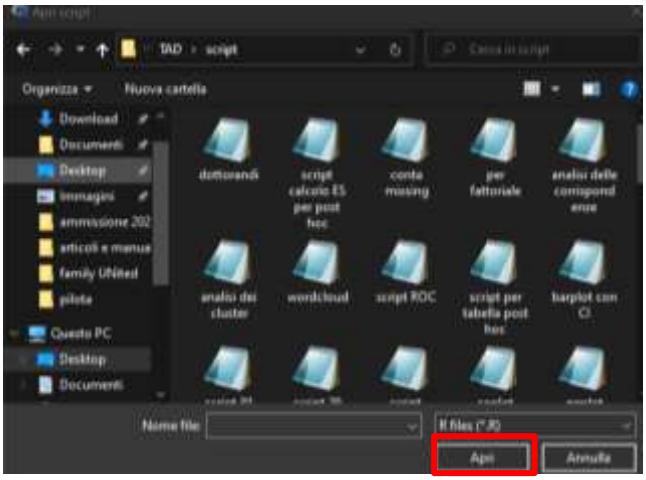


Nella finestra Script, i comandi sono scritti uno dopo l'altro, **senza il prompt `>` della console**. Si possono anche scrivere promemoria utilizzando `#`:



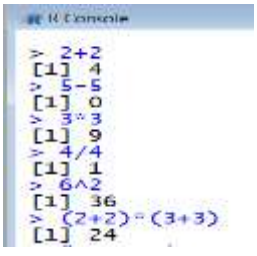
Lo **script viene salvato** (**File → Salva come / Save as** per attribuirgli un nome all'atto della creazione, o **`File → Salva / Save`** per salvare modifiche a uno script già esistente), in una cartella / directory a vostro piacere, come file di R, con estensione .R. Può essere aperto tutte le volte che dovrete rifare le analisi dello script o modificarle per adattarle a nuove esigenze, usando: **`File → Apri script / Open script`**:





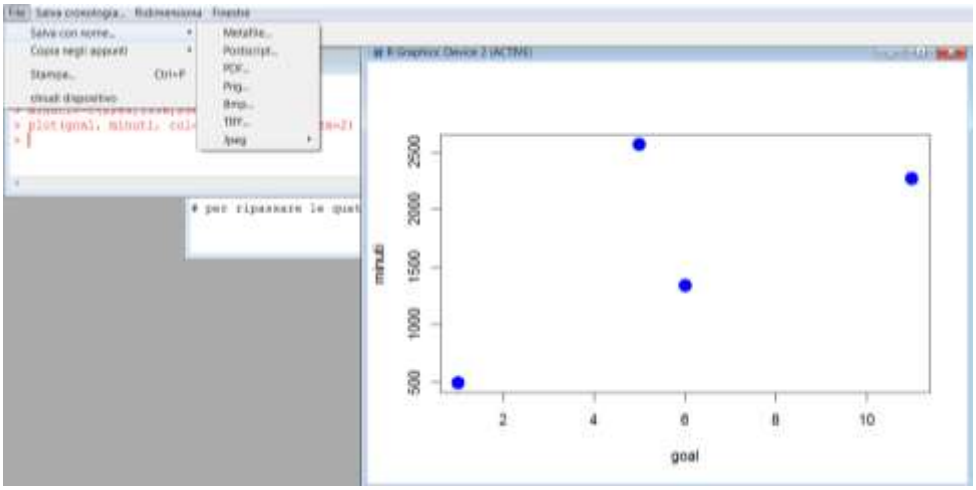
Una volta creato o riaperto uno script, si possono eseguire i comandi che contiene o (per gli utenti Windows) utilizzando il menu **`Modifica/Edit → Esegui tutto / Run all`** (per eseguire tutti i comandi dello script) oppure **`Esegui linea o selezione / Run line or selection`** (per eseguire solo i comandi precedentemente selezionati nello script con il mouse, o solo il comando sulla cui linea è posato il cursore. Se usate MacOS, usate **`Edit → Execute`**.
Si può anche cliccare con il tasto destro del mouse nello script per selezionare l'opzione desiderata.

L'output dei comandi selezionati è **stampato nella finestra Console**:



1.3.3 La finestra dei grafici

Come vedremo in dettaglio (Capitolo 3), R dedica ai grafici una finestra apposita, creata quando in console o nello script si chiede di produrre un particolare tipo di grafico. Attenzione, però: ogni grafico successivamente creato si sovrappone ai precedenti. Quindi, se si desidera conservare per usi futuri il grafico, questo deve essere salvato (come file immagine o, qualitativamente meglio, come .pdf) o copiato in un altro file (ad esempio di Word): basta posizionarsi nella finestra `Graph` per attivare, nel menu File, le necessarie opzioni.



Il grafico precedente (grafico a dispersione, usato per rappresentare visivamente la relazione tra due variabili continue) è stato creato con i comandi che vedete in console, riferiti ai dati che useremo nel paragrafo 3.

Infine, R produce, a richiesta, anche una finestra per la visualizzazione dei dati su cui si sta lavorando: si attiva con il comando di console `View(nomedeldataset)`, e la useremo nel prossimo capitolo.

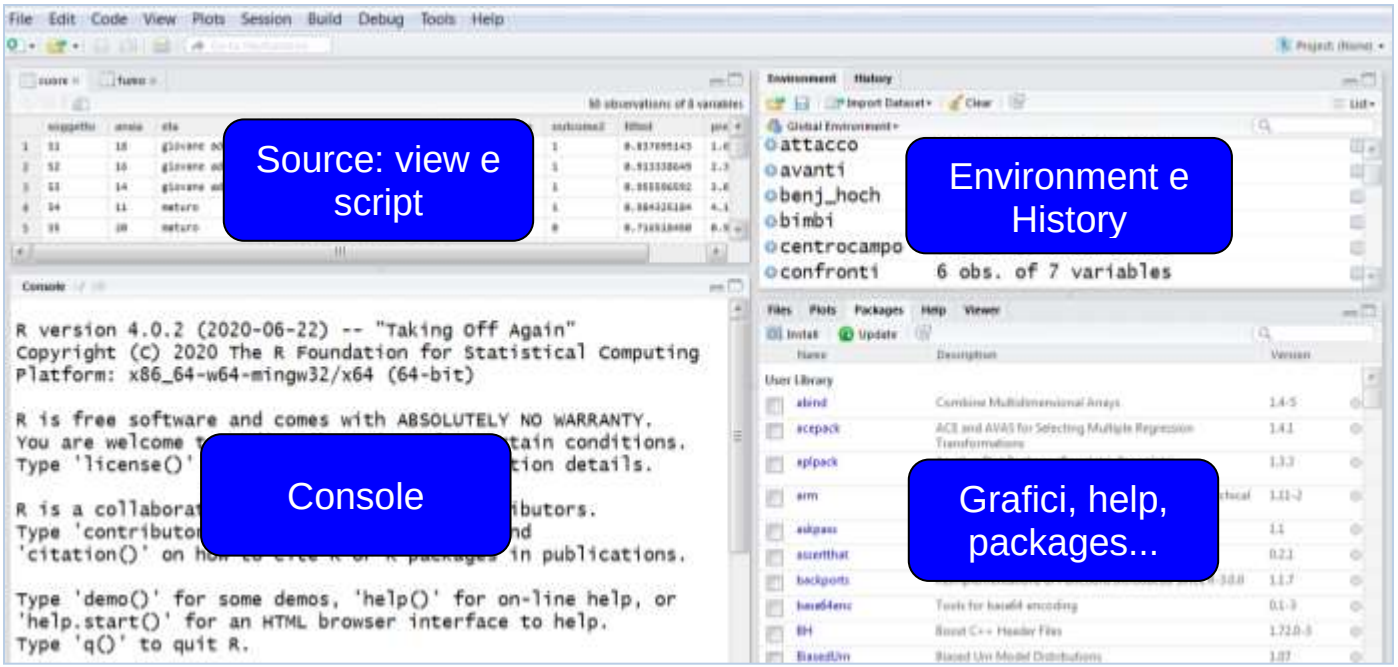
Vediamo adesso **due diverse interfacce grafiche** che possono sollevarci nel compito di interagire con R: la più importante, **RStudio**, deve essere scaricata a parte, ma la sua versione base è gratuita e perfettamente funzionale per i nostri scopi.
La seconda, `Rcommander`, è un **package** di R, e può quindi essere scaricata dopo aver installato R: è comoda soprattutto per fare analisi un po' più complesse delle analisi descrittive che vedremo in Tecniche di Analisi di Dati I, ma troveremo il modo di darle un'occhiata.

Non è assolutamente obbligatorio usare l'una o l'altra, o una e l'altra, o né l'una né l'altra: il software è sempre R, e quello che ci interessa è il prodotto, non il modo con cui interagiamo con lui. **Scegliete l'interfaccia che vi pare più coerente** con il vostro stile cognitivo e più produttiva per arrivare ai nostri fini, ovvero capire qualcosa da una massa di dati.
Ricordate che RStudio e `RCommander` NON sostituiscono R, che deve essere e restare installato.

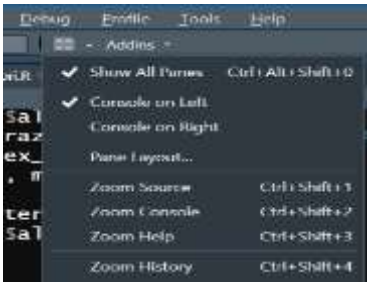
Nei prossimi capitoli, useremo indifferentemente R o RStudio per operare sui dai e fare le analisi, evidenziando le eventuali differenze di utilizzo delle due interfacce.

1.4 RStudio

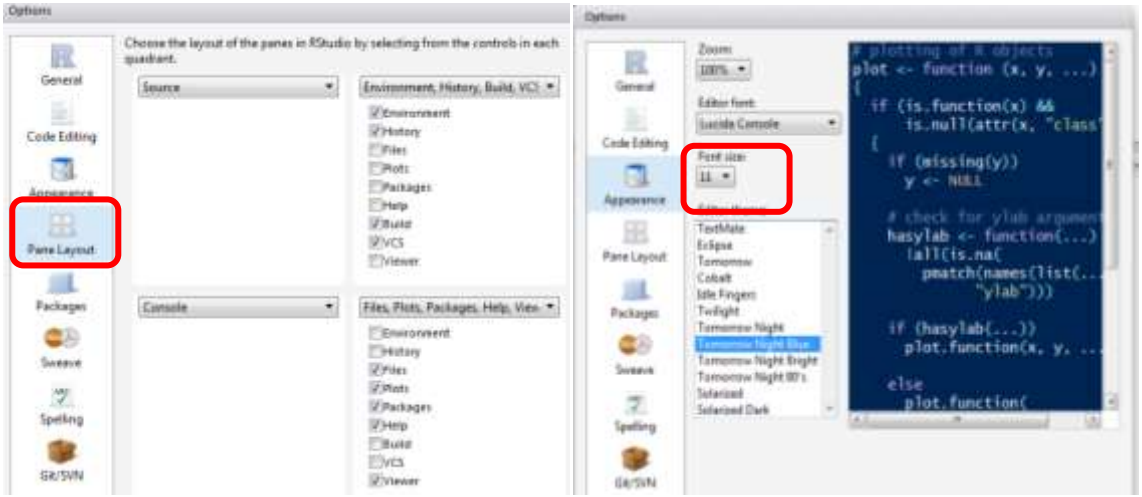
L'interfaccia RStudio non fa parte del mondo CRAN: si scarica da https://rstudio.com/products/rstudio/download/ . Esiste una versione "professionale", a pagamento, che fa un sacco di belle cose, ma per noi inutili; se volete usare RStudio, scaricate la versione *free* (**RStudio Desktop**, **non** RStudio Server) e installatela, **dopo** aver scaricato e installato R. RStudio riconosce automaticamente la versione di R che avrete installato sul vostro computer, e si adeguerà da solo ai suoi eventuali aggiornamenti (la versione più recente è stata aggiornata nella primavera del 2022).
Dopo il download, si esegue il file di installazione come per qualsiasi altra applicazione: potrete usare RStudio **senza aprire R**, perché condividerà tutte le libraries, le directories, i packages, i file che avete salvato nella directory di R. All'avvio, l'interfaccia si presenta **quadripartita:**



Nel Menu `Tools→ Global Options→ Panel LayoutErrore. Il segnalibro non è definito.` è possibile specificare opzioni di layout diverse, se desiderato, per le finestre Visualizza / script e Miscellanea, selezionando altri fogli o deselezionando quelli presenti di default. In `Tools→ Global Options→ Appearance` potete specificare zoom, font e varianti di colore-sfondo molto nerd. Potete anche usare **l'icona** sulla barra dei menu per impostare o ripristinare l'impostazione dei panel e il loro zoom:



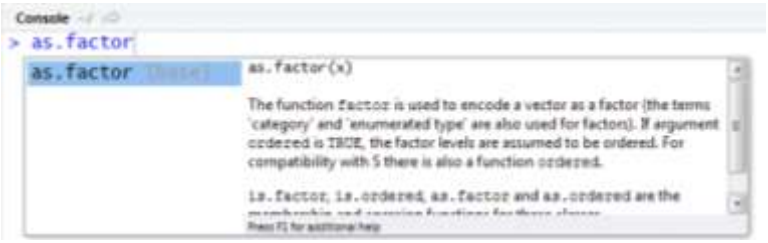
Le **ampiezze** orizzontali e verticali delle finestre possono essere aumentate o diminuite **posizionando il cursore sul telaio interno che separa le finestre e spostandolo**.



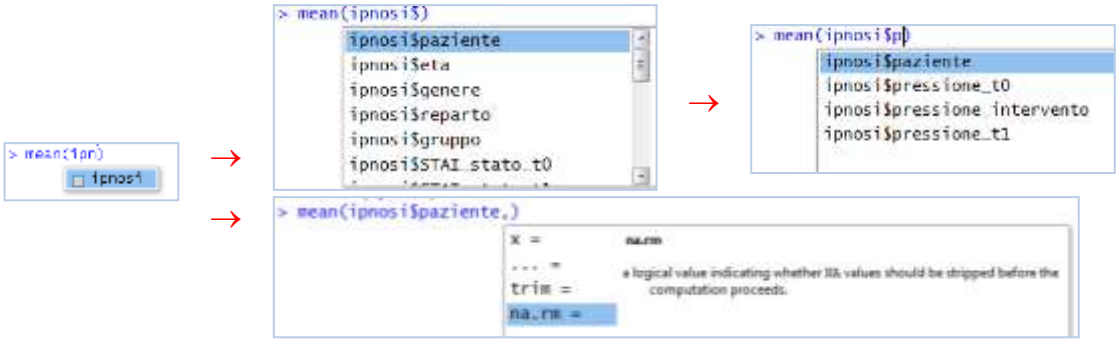
2.5.1 Console

All'apertura del programma, la finestra `Console` occupa tutta la parte sinistra del riquadro; la finestra `Visualizza` comparirà stabilmente, infatti, dopo aver caricato un dataframe. È la classica console di R (qui si digitano i comandi e compare l'output) e si gestisce anche con il menu **`Edit`** (`Clear console`, ad esempio, per fare pulizia). L'opzione di

completamento dei comandi è più intuitiva in RStudio che nell'interfaccia di R, ed è **sensibile al contesto** in cui si trova il cursore: premendo il tasto Tab →| durante la scrittura di una funzione compaiono in una finestra a discesa i candidati possibili, con una breve descrizione tratta dall'Help (ad esempio, `as.factor`); premendolo nuovamente si conferma l'inserimento (tasto Esc per annullare); se una sola opzione è possibile nel punto in cui termina la digitazione (ad esempio `as.fa`), il tasto Tab completa automaticamente la funzione.



Quando si digita **all'interno della funzione**, Tab **ricorda gli elementi da inserire**, a seconda della posizione del cursore nella funzione. Dopo il nome del `dataframe$,` propone tutte le variabili; digitata l'iniziale del nome della variabile, propone solo quelle coerenti; terminato un argomento della funzione, se la funzione ne prevede altri, opzionali od obbligatori, li elenca con una breve descrizione:



RStudio **fornisce automaticamente gli elementi che compaiono in coppia** (parentesi tonde e quadre, virgolette semplici e doppie) non appena digitato il primo, con un piacevole risparmio di errore.

Come in R, è possibile richiamare in Console un comando precedentemente digitato, con le frecce Up e Down; se la sessione di lavoro è lunga e la ricerca con le frecce lenta, c'è un trucco che integra il foglio `History` della finestra `Workspace`: con la combinazione di tastiera **`Crl+Up`** si apre una finestra pop up con i comandi precedentemente editati e selezionabili. Se usate la combinazione `Crl+Up` durante la digitazione di un comando, si apriranno solo i prompt congruenti con la digitazione:

2.5.2 Environment / History e Visualizza / Script

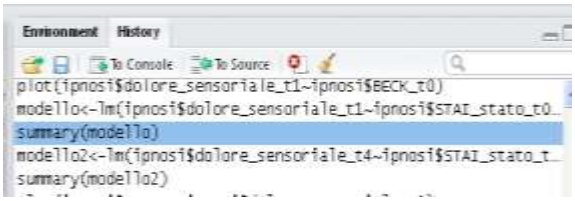
Nella finestra `Workspace` troviamo **`Environment`** e **`History`**, che descriveremo in dettaglio nel prossimo capitolo.

Anticipiamo che In `Environment` si salvano e si importano i dataset sui fare le analisi; nella sub-scheda **`Global Environment`** sono elencati i dataframe e gli altri oggetti che si creano nella sessione di lavoro. Facendo clic sul triangolino, si ottiene una sintetica descrizione degli oggetti in lista.

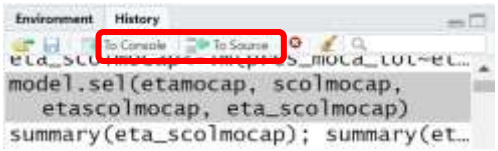
In **`History`**, invece, troviamo lo storico delle operazioni fatte nella sessione, nell'ordine con cui sono state eseguite.

Può essere comodo usare `History` per **creare uno script**, salvando la sequenza di funzioni utili per un'analisi.

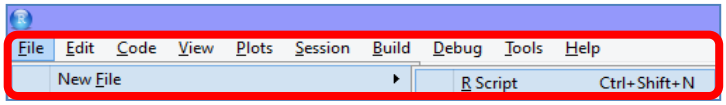
Selezionandone una o più dalla lista (clic e poi Shift+Up o Shift+Down) e cliccando due volte (o premendo Invio) il prompt è spedito alla Console (si può cliccare anche sull'icona `To Console`).

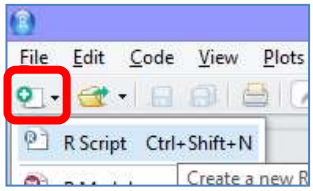


Invece, per creare uno script si può salvare l'intera lista di operazioni (`Save`) o selezionare solo i comandi desiderati e cliccare su `To Source`: comparirà a sinistra uno **`script`** `Untitled`, che sarà possibile salvare con nome a scelta ed estensione `.Rscript`, alternandone la visualizzazione con il dataframe in uso.

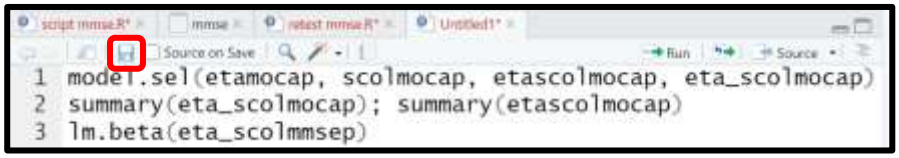


Naturalmente, come nell'interfaccia R è anche possibile creare nuovi script dal Menu `File` → `New` → `Script`, oppure dalla finestra, cliccando sull'icona con **+** → `RScript`:

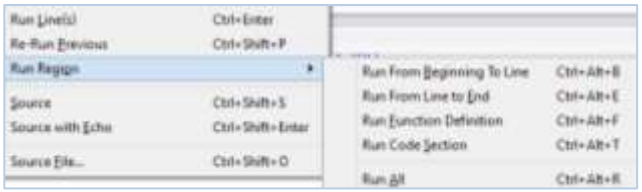




Dopo aver creato lo script, salvatelo!



Per **eseguirlo**, tutto o in parte, posizionate il cursore in un punto qualsiasi del comando / dei comandi e cliccate su `Run` (o fate `Crl+Invio`, in Windows): l'esito apparirà nella finestra Console.



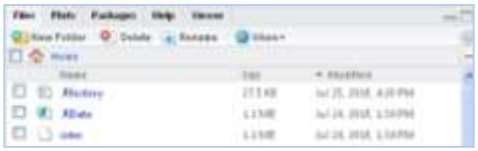
Il menu **`Code`** di `RStudio` offre una lunga serie di possibilità di esecuzione: eseguire la riga o le righe selezionate, ripetere il comando precedente, eseguire dall'inizio fino alla riga selezionata o dalla riga selezionata fino alla fine, ecc. ecc.

La finestra `Workspace` può essere ripristinata ex novo dal menu `Session`→ **`Clear Workspace`**.

2.5.3 Miscellanea

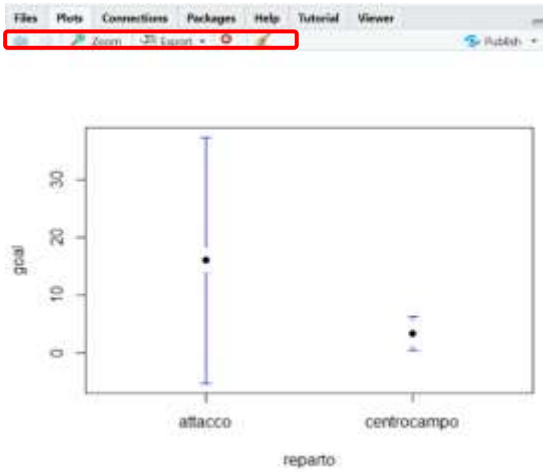
In questa finestra ci sono più fogli:

`Files`: elenco dei file e delle cartelle utilizzati e utilizzabili nella directory;



`Plots`: Qui compaiono i grafici richiesti nelle analisi; possono essere esportati (**`Export`**) e salvati come immagini (`Save as image`) o .pdf(`Save as PDF`) o copiati e incollati in altre applicazioni (`Copy to clipboard`: in questo caso, ricordate di selezionare l'opzione: `Copy as Metafile`). Per aprire una finestra in cui il grafico appare ingrandito, clic su `Zoom`. A differenza dell'interfaccia classica di R, i possono creare più grafici **spostandosi dall'uno altro con con le frecce**. Le ultime due icone sulla barra li cancellano uno per volta, o tutti quanti.

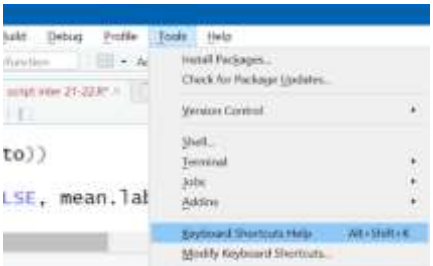
RStudio offre una library specifica per la costruzione interattiva di grafici: **`manipulate`**.

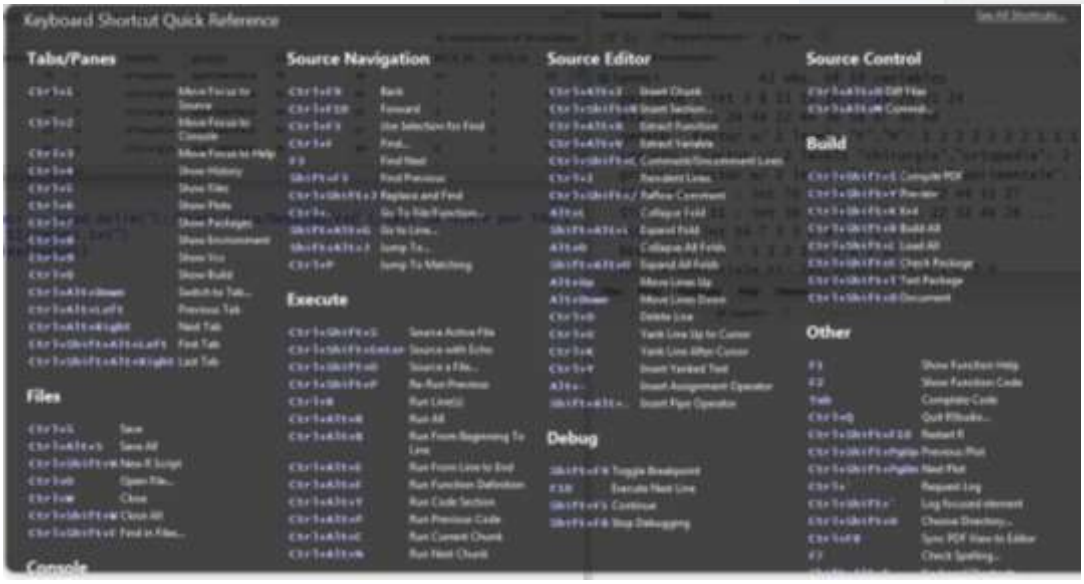


`Packages`: contiene l'elenco dei packages a disposizione: la descriveremo in dettaglio nel prossimo capitolo (§2.1.1).

`Help`: la finestra salvavita ☺ potete cercare informazioni su singole funzioni, package o l'intero sistema R, digitando la richiesta nello slot con la lente d'ingrandimento. L'aiuto è cercato nel materiale che è stato scaricato con i package, compresi quelli di base, per cui funziona anche se siete offline.

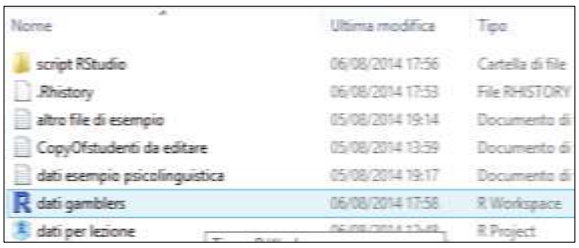
Rstudio offre una serie di comandi di tastiera, diversi a seconda del sistema operativo (la schermata che vedete si riferisce a Windows), che possono far risparmiare un po' di tempo: l'elenco è visibile con `Help→Keyboard shortcuts`:





Quando si chiude, il programma chiede se si vuole: "`Save the Workspace imagine`". Se si risponde affermativamente, RStudio salva tutto ciò che è contenuto nella finestra `Workspace`(dataframe e script) in un file **`.Rdata`**.

Nella successiva sessione di lavoro, viene caricato il workspace precedentemente salvato, che si può cambiare dalla finestra `Workspace`, cliccando sull'icona `Load Workspace` e aprendo il file `.Rdata` desiderata (anche dal `menu Session→ Load Workspace`). Il Workspace può essere salvato durante una sessione di lavoro cliccando sull'icona nella finestra o dal menu `Session – Save Workspace As`.



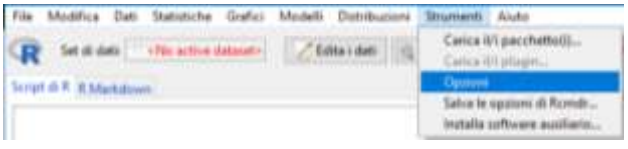
2.6 RCommander

L'interfaccia RCommander è contenuta nel package **`Rcmdr`**, scaricabile dal sito CRAN. Installatelo come spiegato: può volerci parecchio tempo, perché il package è collegato a tantissime dependencies (ricordate: se usate la finestra Packages di RStudio, selezionate  ; se usate `install.packages`, inserite l'argomento **`dependencies=TRUE`**). Quando sarà installato, caricatelo nella sessione di lavoro: si aprirà una nuova finestra di R, a prima vista un po' deludente:

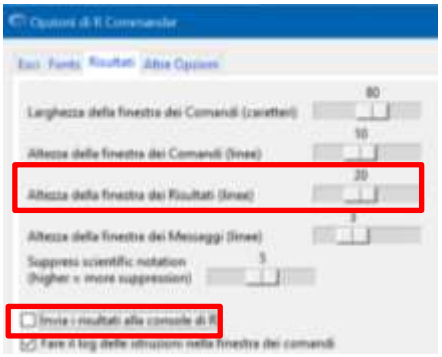
Come scritto nell'immagine, la finestra bianca in alto è uno **script** (notate che, infatti, non compare il prompt >), in cui i comandi devono essere digitati e poi eseguiti con **`Esegui`** (**`Run`**, se scaricate `Rcmdr` da un mirror anglofono). Sempre qui appaiono gli script che stanno "alle spalle" dei comandi richiesti con i **Menu**, nella parte superiore della finestra, che sono la vera "scorciatoia" di `RCommander`.

Può capitare che all'apertura **vediate solo lo script**: in questo caso, gli output delle analisi e i grafici compaiono nella relativa finestra di R o di RStudio, a seconda che abbiate aperto l'uno o l'altro, per cui bisogna fare avanti e indietro. Se volete vedere analisi e grafici solo in RCommander, dovrete fare una piccola operazione:

1. Andate nel menu **`Strumenti`** e selezionate **`Opzioni`**:

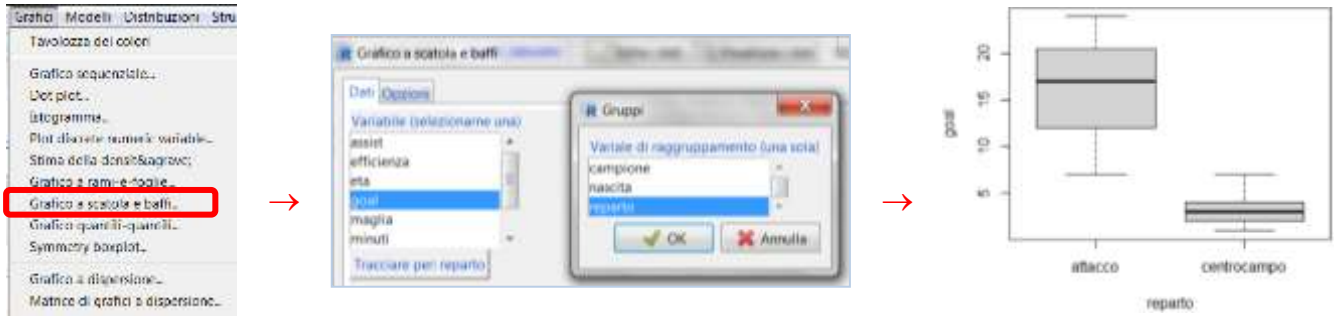


2. Nella scheda **`Risultati`**, **deselezionate** “**Invia i risultati alla console di R**”, e definite l’altezza della finestra dei risultati, che troverete impostata a zero.



3. Cliccate su `Restart RCommander` in fondo alla scheda e, al riavvio, troverete la bipartizione della finestra: i risultati appariranno in `RCommander`.

Una volta eseguito un comando con un menu, lo script relativo è modificabile dall’utente per affinare il risultato. Per esempio, creiamo con il menu **`Grafici`** un grafico a scatola (boxplot) per rappresentare la distribuzione della variabile `$goal` nei due reparti (media, distribuzione interquartilica, minimo e massimo: lo vedremo nel capitolo 3): è veramente *minimal* nell’aspetto, ma suggerisce che, per fortuna, gli attaccanti hanno segnato più dei centrocampisti:



Ora possiamo ravvivare il grafico: editiamo il comando, che è stato riportato nello script, aggiungendo l’informazione sul colore:

Con RCommander possiamo **eseguire molte analisi** (anche se non proprio tutte quelle nel nostro programma): nel seguito della Dispensa, vedremo per ciascuna di esse i relativi comandi, nonché i pro e i contro dell’uso di RCommander per eseguirle.

Ora cominciamo veramente a lavorare con R – o, meglio, a chiedere a R di lavorare per noi.

Capitolo 2

Usare R

AVVERTENZA PRELIMINARE IMPORTANTE

R è ***case sensitive***: considera una stessa lettera **maiuscola e minuscola come simboli diversi**. Attenzione, quindi, perché una delle situazioni più frustranti è trovarsi di fronte a una serie di messaggi di errore in cui, nonostante voi siate cocciutamente convinti di scrivere il comando corretto, R vi dice che non capisce cosa vogliate, solo perché avete scritto "`Sum`" (fai la somma di) invece di "`sum`", o "mean(`età`)" (fai la media dell'oggetto età) quando invece il nome dell'oggetto di cui volete conoscere la media è "`Età`".

2.1 Comandi, oggetti e funzioni

Come già detto, tutto quello che vogliamo R faccia deve essere digitato in console come **comando**. I comandi sono, in genere, strutturati in due parti: gli **oggetti** e le **funzioni**.

Un **oggetto** è **qualsiasi cosa sia creata in R**: una variabile, un modello statistico, un test, un grafico, una raccolta di variabili, eccetera. Può essere costituito da un singolo valore (ad esempio, la media di una serie di valori), o da una serie d'informazioni disparate (ad esempio, gli output delle analisi che impareremo a fare). Se ne distinguono di diverso tipo (`class`) e modo (`mode`), che vedremo man mano.

Le **funzioni** sono le **cose che si fanno in R per creare gli oggetti e per lavorare sugli oggetti**. Le funzioni sono, a loro volta, composte da **argomenti**, che sono le **istruzioni necessarie** a comporre le funzioni. Nei casi più semplici, l'unico argomento della funzione è semplicemente il nome dell'oggetto cui si riferisce. In (molti!) altri casi, le istruzioni da dare sono più complesse e quindi servono più argomenti, separati da virgole, ciascuno dei quali gestisce uno specifico aspetto della funzione; alcuni sono obbligatori perché R esegua il comando, altri sono opzionali e gestiscono le preferenze dell'utente. Li vedremo, naturalmente, funzione per funzione. Come regola generale, le funzioni si aspettano che gli argomenti vengano inseriti in un ordine prefissato: se omettete il nome dell'argomento nella funzione, allora dovete rispettare l'ordine previsto; se invece li inserite con il loro nome, potete metterli in qualsiasi ordine.

Molte delle funzioni che useremo sono contenute nelle statistiche di base scaricate al momento della prima installazione di R: dovremo quindi solo preoccuparci di scriverne correttamente il nome in console; altre sono contenute in *package* appositi, che scaricheremo secondo necessità.

Oggetti e funzioni **sono legati** dalla **funzione di assegnazione** `<-` (`segno di minore + trattino`), il cui significato per R è "**assegna all'oggetto le proprietà della funzione**" . Quindi, scrivendo:

```
oggetto <- funzione
```

diciamo a R: "all'oggetto Y vanno assegnate le proprietà della funzione X".

Se vi risulta più facile, potete immaginare `<-` all'incirca come la bacchetta magica con cui una funzione X **crea** l'oggetto:

```
sottoinsieme <- subset
```

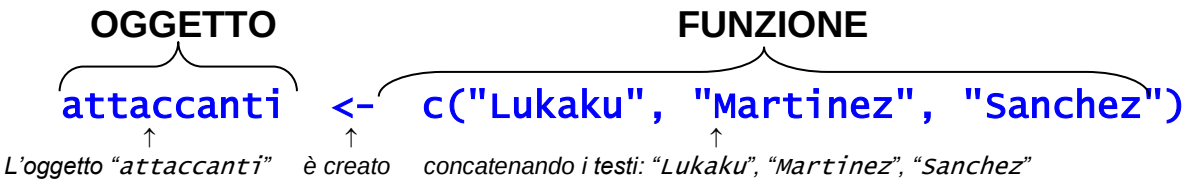
=

I **nomi** degli oggetti non possono contenere alcuni caratteri che R considera speciali ($, ~, /, ecc.), non possono iniziare con numeri né contenere spazi vuoti (che si possono sostituire con _ o .); è bene non denominare oggetti con nomi che identificano funzioni (non create un oggetto di nome "mean", che è la funzione per calcolare la media di una distribuzione). Usate un po' di razionale fantasia: se assegnate lo **stesso nome** a oggetti diversi, **l'oggetto più recente sovrascrive quello vecchio**, cancellandolo – e R non vi avvisa di quello che state per fare. Evitate gli accenti, più che altro perché possono essere letti male passando da un sistema operativo a un altro, o importando un dataframe creato con un altro programma.

Vediamo un primo esempio usando una delle funzioni più semplici, ma più onnipresenti e importanti: la funzione `c`.

`c` sta per **`combine`**, ovvero *unisci*, *combina*, *concatena*: la funzione serve quindi per unire più elementi in un solo oggetto. Può unire **più valori numerici**, oppure **più stringhe di testo**, ma **tutti** devono essere **dello stesso tipo**. La serie di elementi così uniti costituisce un **vettore**.

Nell'esempio, voglio creare un oggetto che rappresenta il **reparto di attacco della squadra Internazionale F.C** nell'anno dello scudetto 2020-2021: devo quindi unire **più elementi di testo**, ciascuno dei quali rappresenta un argomento della funzione. Quando gli elementi da unire sono stringhe di testo, devono essere racchiusi da "":



L'oggetto "attaccanti" è un **vettore** (una **variabile,** una **distribuzione**) costituito da più elementi testuali. La natura degli elementi che costituiscono una variabile, ovvero la loro scala di misura (ricordate? Nominale, metrica… le ripassiamo nel prossimo capitolo) definisce il **tipo (classe) di variabile**. R riconosce più tipi di variabili: quelli con cui ci troveremo ad avere a che fare più spesso sono di classe **`numeric`** (gli elementi sono numeri decimali $\mathbb{R}$) o **`integer`** (numeri interi $\mathbb{Z}$), **`character`** (gli elementi sono stringhe di testo), **`factor`** (ogni elemento identifica un diverso gruppo di casi: le variabili `factor` sono quindi **variabili di raggruppamento**). Possono essere create anche variabili di tipo `logic` ("TRUE", "FALSE"), `Date` (date), `complex` (numeri complessi $\mathbb{C}$, con parti reali $\mathbb{R}$ e parti immaginarie *i*). Naturalmente, la classe di una variabile delimita le operazioni e le analisi che possiamo sensatamente condurre su di essa. Difficilmente R sbaglia nel classificare la variabile creata, ma possiamo comunque verificare che la classe della variabile corrisponda alle nostre aspettative usando la funzione **`class`**, il cui argomento è il nome della variabile:

```
class(attaccanti)
[1] "character"
```

Vedremo successivamente come rimediare quando R fraintende la classe.

Per visualizzare l'oggetto creato, **se ne digita il nome**, seguito da **Invio**, oppure si usa la funzione **`print(nome dell'oggetto)`**, o ancora si racchiude la funzione che crea l'oggetto tra `()`:

```
attaccanti
[1] "Lukaku"   "Martinez" "Sanchez"
print(attaccanti)
[1] "Lukaku"   "Martinez" "Sanchez"
(attaccanti<- c("Lukaku","Martinez","Sanchez"))
[1] "Lukaku"   "Martinez" "Sanchez"
```

Potremmo voler sapere di quanti elementi è costituito il reparto attaccanti: facciamo finta di non saper contare gli argomenti inseriti e chiediamo **di quanti elementi è composta la variabile** `attaccanti` usando **`lenght`**, il cui argomento è il nome dell'oggetto:

```
length(attaccanti)
[1] 3
```

Se vogliamo sapere qual è l'elemento maggiore della variabile `attaccanti` (non nel senso del valore calcistico, ovviamente), useremo la funzione **`max(oggetto)`**: poiché la variabile è composta da stringhe di classe `character`, R restituisce l'ultimo elemento in ordine **alfabetico**; se fosse numerica, restituirebbe il valore più grande. Non stupirà sapere che per conoscere il valore più basso, numerico e alfabetico, si usa **`min(oggetto)`**:

```
max(attaccanti)
[1] "Sanchez"
min(attaccanti)
[1] "Lukaku"
```

Se, però, cerchiamo di applicare una funzione che non è compatibile con la classe della variabile, ad esempio se cerchiamo di calcolare la media di una variabile testuale con **`mean(distribuzione)`**, R si rifiuta e vi spiega perché:

```
mean(attaccanti)
[1] NA
Warning message:
In mean.default(attaccanti) : l'argomento non è numerico o logico: restituisco NA
```

NA (Not Available, non disponibile) è la sigla che R usa per indicare che il **dato**, in output o in una variabile, è **mancante**.

Si possono **richiamare i comandi** precedentemente digitati e usarli nuovamente, oppure modificarne una parte, usando i tasti **Freccia su** △↑ o **Freccia giù** ↓▽ della tastiera. Questo, di solito, garantisce un buon risparmio di tempo e di errori di digitazione

Si possono digitare più comandi[3] per volta in un'unica riga, separandoli con punto e virgola **`;`** Viene restituito un output in cui **ogni risposta a ogni comando è identificata dal numero [1]**:

```
max(attaccanti); min(attaccanti)
[1] "Sanchez"
[1] "Lukaku"
```

Cautela: fate un po' di pratica prima di azzardarvi in linee di comando complesse, perché è più facile trovare errori – se ve ne sono - in singole linee.

Le funzioni si possono applicare a oggetti per creare **altri** oggetti. Per esempio, posso salvare nell'oggetto `Sanchez` il valore più alto dell'oggetto `attacco`:

```
sanchez<-max(attaccanti)
sanchez
[1] "Sanchez"
```

Gli oggetti possono essere modificati senza doverli creare ex novo. Per esempio, nella stagione 2021-2022 Lukaku non era più tra gli attaccanti. Possiamo emendare `attaccanti` **eliminando l'elemento** "`Lukaku`", in questo modo:

```
attaccanti_meno_lukaku <- attaccanti[attaccanti != "Lukaku"]
```

il nuovo oggetto `attaccanti_meno_lukaku` — *è creato da* — *il vecchio oggetto `attaccanti`, in cui gli elementi di `attaccanti`* — *non sono uguali a* — *"`Lukaku`"*

[3] Unendo più comandi, potete creare e salvare le vostre funzioni: l'argomento esula dai fini del nostro programma, e ne faremo solo un cenno in Tecniche di Analisi di Dati II

La funzione si può tradurre come "usa l'oggetto `attaccanti` **senza** l'elemento che segue !=": **!=** sta, infatti per "**non uguale a**". Notate le **[ ]**, che si usano quando i comandi si riferiscono alla **struttura di oggetti** che contengono elementi atomici: variabili, vettori, dataframe, matrici, liste…; li vedremo dal paragrafo §3.2 e li useremo praticamente sempre. Il nostro nuovo oggetto si presenta come:

```
attaccanti_meno_lukaku
[1] "Martinez" "Sanchez"
```

Attenzione: abbiamo creato **un oggetto con nome** diverso, per evitare che assegnando lo **stesso nome R sovrascriva**, ovvero cancelli, il **precedente**! Come già detto, R non avvisa che quanto state per fare potrebbe essere un danno.

In attesa di notizie del mercato acquisti, potremmo riempire il posto di Lukaku con un attaccante ipotetico, concatenando all'oggetto un nuovo elemento con la funzione `c`:

```
attaccanti_meno_lukaku<-c(attaccanti_meno_lukaku, "Chilosà")
attaccanti_meno_lukaku
[1] "Martinez"  "Sanchez" "Chilosà"
```

Nella stagione 2022-2023 Lukaku è tornato! Torniamo allora al reparto precedente, unendo due comandi in un'unica riga: prima aggiungiamo "Lukaku" all'oggetto `attaccanti_meno_lukaku`, poi eliminiamo l'ipotetico `Chilosà` dalla squadra:

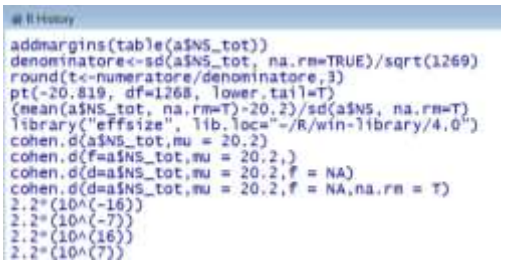
```
attaccanti<-c("Lukaku",attaccanti_meno_lukaku)
attaccanti<-attaccanti[attaccanti!="Chilosà"]
[1] "Lukaku"   "Martinez"  "Sanchez"
```

Ogni tipo di oggetto creato può essere salvato e restare disponibile in altre sessioni di lavoro, purché ci si ricordi di **salvare** il **`workspace`**, ovvero l'area di lavoro che contiene tutti gli oggetti e le funzioni creati nella sessione di lavoro. Si usa il menu **File→Salva area di lavoro / Save workspace** per salvare in una directory (cartella) a piacere tutto quello che è stato fatto nella sessione, racchiuso in un file (che si definisce `R Image`) con estensione `.RData`. In Rstudio, usate il menu **Session→ Save workspace as**. Se non si definisce alcun'altra directory, R salva per default nella cartella in cui è stato installato.

Nella classica interfaccia di R, potete vedere tutte le funzioni che avete usato nelle sessioni di lavoro precedenti con `history` - non mettete nulla tra le parentesi tonde:

```
history()
```

Si apre la finestra **`History`**, che è null'altro che un editor di testo da cui potete copiare e incollare le funzioni in console:



Se state usando Rstudio, abbiamo già visto che la History è sempre disponibile di fianco a Environment:

Quando si riapre R o RStudio per ricominciare a lavorare, viene caricato automaticamente il `workspace` con cui si era conclusa la precedente sessione. Se, invece, dovete usare un'altra area di lavoro precedentemente salvata, caricatela usando **File→ Carica area di lavoro / Load workspace** (in Rstudio usate il menu Session→ Load workspace)
Precauzionalmente, ogni volta che chiudete R il sistema vi chiede se volete salvare il lavoro fatto; ricordatevi di salvare il lavoro anche **durante le sessioni** (si può usare la combinazione di scelta rapida con i **tasti Crl+S** in Windows**, Cmd+S** per i Mac), se volete mantenere memoria di quanto state facendo, perché R, anche se è di solito molto affidabile, può andare incontro a improvvisi e irrimediabili crash.
Ricordiamoci che, come già illustrato nel §1.3.2, **una serie di comandi può essere salvata con l'editor di R in uno script**, e riutilizzata o modificata all'occasione. Durante il lavoro, potete aprire un nuovo script (**File→ Nuovo script / New script**), selezionare con il mouse il comando in console e copiarlo e incollarlo nello script (Crl+X, oppure Crl+C e poi Crl+V, oppure clic con il tasto destro del mouse); ricordate di togliere il prompt `>`, che non si usa negli script.
Per conoscere quali oggetti avete salvato nel workspace attivo, nel tempo, chiedetelo a R con **`objects()`** - non scrivete nulla tra le parentesi: in Console ne vedrete stampato l'elenco. Se tra essi ci sono oggetti ormai inutili, potete cancellarli usando la funzione **`rm(oggetto)`**, che sta per **remove**: attenzione, però, l'operazione di `rm` non è reversibile. Per esempio, dato che da qui in poi non useremo più il vettore `attaccanti_meno_lukaku`, possiamo rimuoverlo:

```
rm(attaccanti_meno_lukaku)
```

Riprenderemo il reparto di attacco nei paragrafi seguenti.

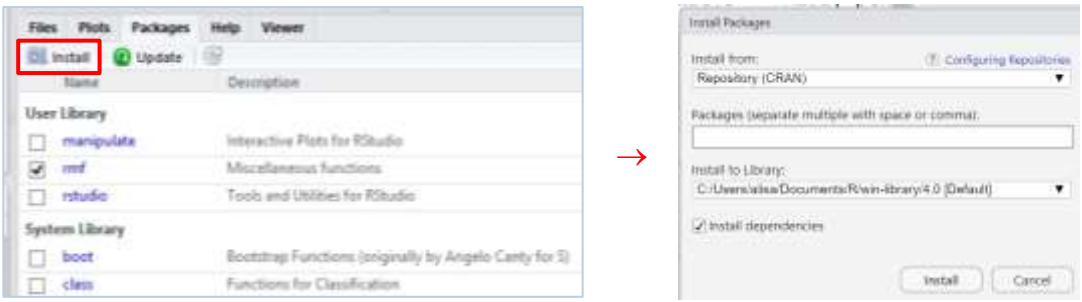
2.1.1 Installare e caricare packages

Quando installate R la prima volta, vi vengono forniti diversi **package** di base: i packages sono eterogenee raccolte di dataframe (usati per gli esempi delle funzioni contenute), funzioni e codici, che, al momento dell'installazione, sono salvati in directories (cartelle) chiamate **library**. I package di base (`stats`, `lattice`, `MASS`, `utils`...) contengono moltissime cose utili, sono salvati nella **System Library** del vostro *device* e, con qualche **eccezione**, **non devono essere** caricati nel workspace per usarne le funzioni, che sono subito disponibili. Useremo però moltissime funzioni che non sono previste nella dotazione di base: i packages che le contengono dovranno essere scaricati dallo sterminato archivio CRAN, utilizzando:

- in Windows: il **Menu `pacchetti` `Packages` → `Installa pacchetti / Install packages`**, che vi presenta prima la scelta tra uno dei mirror CRAN e poi il lunghissimo elenco delle disponibilità; potete scaricare più package contemporaneamente.
- in MacOS: selezionate **`Packages & Data` → `Pages Installer`**: cliccate su **`List`** e scegliete i package che vi servono, poi su **`Install selected`**.

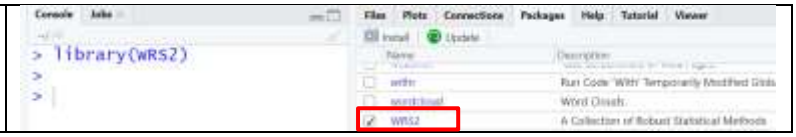
Indipendentemente dal sistema operativo, potete usare il comando: **`install.packages("nome del package")`**; ricordate di inserire il nome tra " ". Questi package scaricati dall'utente sono salvati in un'altra directory, la **User Library**, e devono essere **caricati nel workspace ogni volta** che se ne vuole usare una funzione. In caso contrario, R vi dirà mortificato che non trova la funzione che cercate, anche se il package è correttamente installato.
Per **caricare il package**, con Windows potete usare il menu **`Pacchetti / Packages→ Carica pacchetto / Load package`**, con MacOs **`Packages & Data→ Package Manager`**; in entrambi i casi, è comodo il comando **`library(package)`**, che si riferisce alla cartella (directory) in cui è contenuto il package sul vostro PC: per default, R andrà a cercare in tutte le library che trova (`lib.location= NULL`). Per semplificare le cose, questa volta NON dovete inserire il nome tra " " ☺

Con Rstudio, usiamo la scheda `Package`. Per **installare nuovi packages**, si clicca su `Install Packages`: Rstudio si prepara a installarli dal sito CRAN online (`Install from Repository`) , ma possono essere installati anche offline (`Package Archive File`), se li avevate già scaricati da CRAN e volete installarli dalla cartella in cui li avete salvati.



Per **caricare un package**, individuatelo nella lista e cliccate sul quadratino a fianco del nome: comparirà un segno di spunta e in Console apparirà il corrispondente comando:

```
library("effsize", lib.loc="~/R/win-
  library/4.0")
```



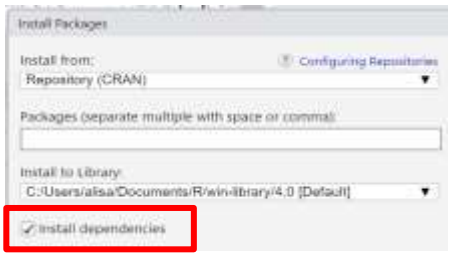
Quando si deseleziona il package dopo l'uso, in Console ci viene annunciato:

```
detach("package:effsize", unload=TRUE)
```

Moltissimi package comprendono diverse ***dependencies***: seguendo la logica *open source* di R, i loro sviluppatori hanno usato il codice di altri package per inserirlo nel proprio, così che quando scaricate il package A dovete scaricare dall'archivio CRAN anche gli altri package B, C, D..., i cui codici sono usati in A, affinché le funzioni di A si attivino correttamente[4]. Nove volte su dieci, R scarica senza problemi il package e tutte le *dependencies*, senza che dobbiate fare nulla più che inoltrare la richiesta di installazione. Per buona misura, **aggiungete un argomento** a `install.packages`:

```
install.packages (userfriendlyscience, dependencies= TRUE)
```

L'argomento `dependencies= TRUE`, separato con una virgola dal primo, è di tipo `logic`: se `TRUE`, R installa le dependencies; se `FALSE`, R non lo fa.



In Rstudio, controllate che sia spuntata l'opzione **`Install Dependencies`** (di default lo è) nella scheda di installazione:

Può comunque capitare, soprattutto con package "pesanti", che l'installazione vada solo apparentemente a buon fine: quando vi trovate a caricare il package nel workspace per lavorarci, può comparire un messaggio di errore, in rosso, che vi dice approssimativamente:

```
library(userfriendlyscience)
 Error in library(userfriendlyscience) : there is no package called 'XYZ'
```

Questo significa che nel momento in cui avete installato `userfriendlyscience`, che ha un sacco di dependencies da portare con sé, non è stato scaricato il package `XYZ`: nessun problema, dall'archivio CRAN s'installa `XYZ` mancante e si carica `userfriendlyscience`. Se `XYZ` era l'unico package mancante, ora `userfriendlyscience` sarà caricato correttamente; se altri mancano, però, si avrà di nuovo un messaggio di errore con il nuovo package che non si trova:

[4] La banalizzazione di questo discorso farà rabbrividire qualunque sviluppatore, ma basta che ci intendiamo...

```
library(userfriendlyscience)
 Error in library(userfriendlyscience) : there is no package called 'ABC'
```

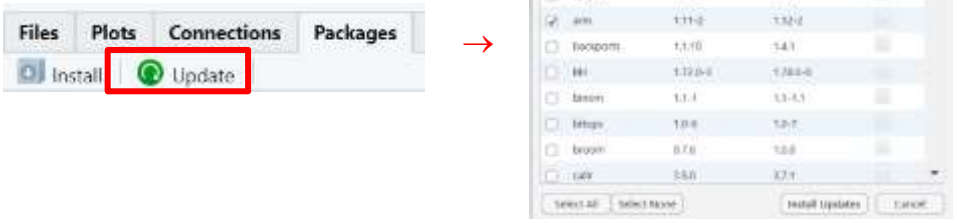
… e si torna daccapo a installare `ABC` da Cran.

Potete verificare quali package avete installato con la funzione `installed.packages()` – non scrivete nulla entro le parentesi – oppure scorrendo l'elenco del menu.

Periodicamente, potete **aggiornare i package** con la funzione **`update.packages()`** – non scrivete nulla entro le parentesi –, anche se, comunque, R avvisa, con un messaggio di ***warning***, quando caricate nell'area di lavoro un package ormai datato. Se non aggiungete nulla in `update.packages()`, R esplorerà tutti i packages nelle vostre libraries e vi chiederà conferma per ogni package da aggiornare; inserendo l'argomento logic **`ask= FALSE`**, invece, R farà da solo, aggiornando tutto quel che troverà datato.

Impariamo a distinguere un messaggio `Error` da un messaggio `Warning`: `Error` segnala l'impossibilità di eseguire qualcosa, `Warning` avvisa che la funzione richiesta è stata eseguita, ma vi chiede di prestare attenzione a quanto riportato nel resto del messaggio.

Con Rstudio, cliccate su **`Update`** (di fianco a `Install`): si aprirà una finestra in cui compaiono i package aggiornati rispetto al momento in cui li avete installati: selezionate quelli che intendete aggiornare e installate l'aggiornamento:



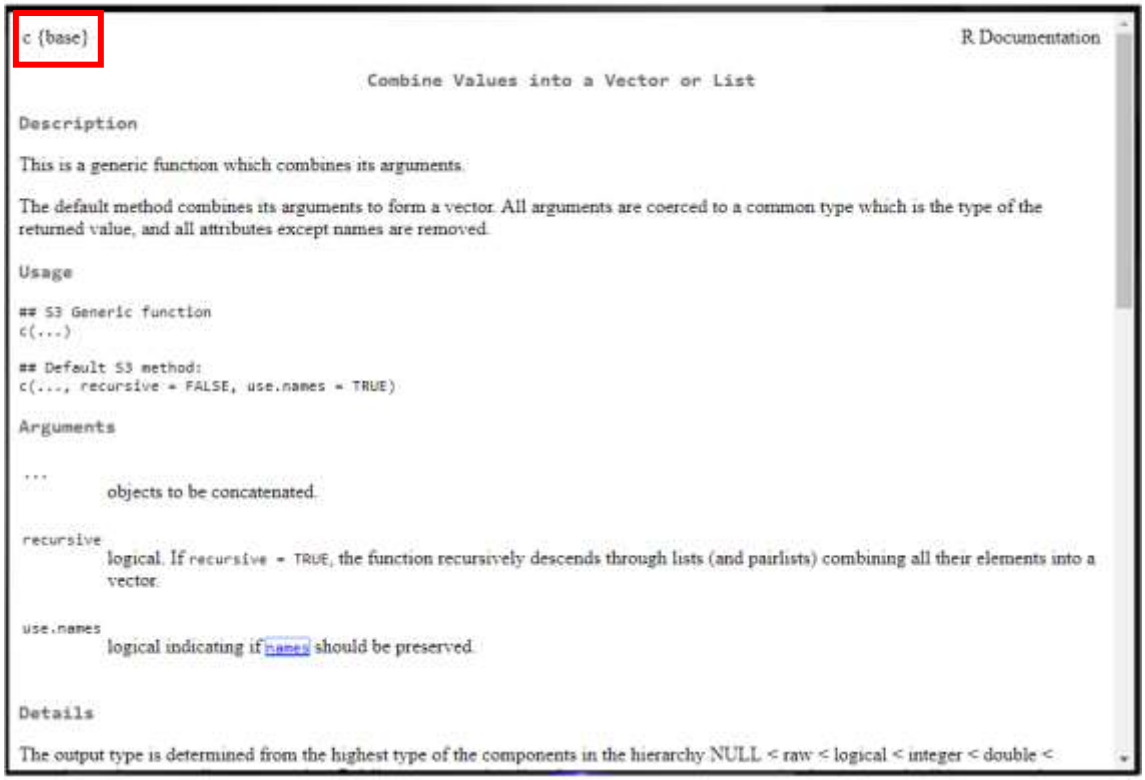
È raro, ma può capitare che due package diversi, entrambi installati e caricati, abbiano una funzione con lo stesso nome: in questo caso, R non sa quale scegliere, e **dobbiamo quindi disambiguare tra le funzioni**, facendo **precedere il nome della funzione dal nome del package che la contiene, separandoli con `::`**. Per esempio, useremo nel tempo sia il package `car` sia il package `Hmisc`, che contengono una stessa funzione, `recode`. Se sono contemporaneamente attivi nel workspace, per usare la funzione `recode` di `car` scriveremo `car::recode(oggetto)`, per usare la funzione `recode` di `Hmisc` scriveremo `Hmisc::recode(oggetto)`.

Rstudio vi facilita le cose: mentre scrivete in Console o in uno script il nome della funzione che desiderate, il suggeritore completa il nome e aggiunge, tra parentesi graffe, i package che contengono funzioni con lo stesso nome: basta selezionare quella che vi serve.

2.1.2 Aiuto

Un'incredibile quantità di materiale informativo sulle funzioni e su R in generale (tutorial, forum, articoli dedicati, blog, manuali), è a disposizione sul web e accessibile con una rapida ricerca per parole chiave sui motori di ricerca, perlopiù in inglese, ma non solo. Anche restando all'interno del mondo di R, però, c'è possibilità di avere aiuto: digitando

`help(funzione)` o `?(funzione)` si ottengono le informazioni sugli scopi della funzione e sui modi per applicarla, che sono **parte del materiale del package che contiene la funzione stessa**. Per esempio, digitando `help(c)` si apre il file:



c {base} R Documentation

Combine Values into a Vector or List

Description

This is a generic function which combines its arguments.

The default method combines its arguments to form a vector. All arguments are coerced to a common type which is the type of the returned value, and all attributes except names are removed.

Usage

```
## S3 Generic function
c(...)

## Default S3 method:
c(..., recursive = FALSE, use.names = TRUE)
```

Arguments

...	objects to be concatenated.
recursive	logical. If recursive = TRUE, the function recursively descends through lists (and pairlists) combining all their elements into a vector.
use.names	logical indicating if names should be preserved.

Details

The output type is determined from the highest type of the components in the hierarchy NULL < raw < logical < integer < double <

A fianco del nome della funzione, tra `{ }` è indicato il package che la contiene: nel caso di `c`, è il package di base – appunto – `base`.

2.2 Inserire dati in R

I dati su cui lavorare con R possono essere inseriti direttamente in R (un processo piuttosto lungo e doloroso, se i dati sono tanti e la struttura in cui sono inseriti è complessa) o importati in R da un fonte esterna. Cominciamo con il vedere il primo caso.

Abbiamo già creato una variabile testuale usando la funzione `c`, cioè l'oggetto **`attaccanti`**. Arricchiamo le nostre informazioni sul reparto di attacco dell'Inter dello scudetto creando un'altra variabile, questa volta **numerica**: il numero di goal segnati nel campionato dai tre attaccanti (fonte: sito ufficiale della Società). Usiamo ancora `c` e chiamiamo la variabile **`goal`**. **Attenzione**: **inseriamo gli argomenti di `c` nello stesso ordine** con cui abbiamo inserito i nomi nell'oggetto `attaccanti`: questo ci consentirà di combinare correttamente le due variabili `goal` e `attaccanti`, associando il nome del marcatore al numero dei suoi goal.

```
goal<-c(24,17,7)
class(goal)
[1] "numeric"
min(goal); max(goal); mean(goal)
[1] 7
[1] 24
[1] 16
```

Se volete conoscere il range di una distribuzione, invece di chiedere `min` e `max` potete usare `range(distribuzione)`:

```
range(goal)
[1] 7 24
```

R ha correttamente riconosciuto gli argomenti di `c` come numeri (notate che **non sono stati messi tra " "**), e quindi la variabile creata è di classe **`numeric`**. Il numero di minimo di goal segnati è 7, il massimo 24; in media, gli attaccanti hanno segnato 16 goal a testa.

Possiamo **combinare i due vettori `goal` e `attaccanti` in una struttura** chiamata **dataframe**: un dataframe è una sorta di tabella, assimilabile alla struttura di un foglio di calcolo (ad esempio, Excel), che può contenere **variabili di tipo diverso**: ogni riga contiene le osservazioni in tutte le variabili relative a uno specifico caso (**unità statistica**: soggetto, paziente, cane, gatto, ttaccante…); ogni colonna contiene i dati relativi a una sola variabile per tutti i soggetti. Questa tipologia di struttura si definisce **wide format** ed è la più usuale – ma non è l'unica. Useremo, quando affronteremo i disegni fattoriali a misure ripetute, una diversa struttura in cui ogni riga corrisponderà a una singola misurazione (**long format**).

Per creare il primo dataframe in wide format usiamo la funzione **`data.frame`**Errore. Il segnalibro non è definito., associandola a un nuovo oggetto: i suoi argomenti sono il **nome** scelto per identificare la variabile nel dataframe e il **vettore** che ne contiene i dati, **uniti da =,** ripetuti per tutte le variabili da inserire nella struttura.

```
data.frame(nome della prima variabile= vettore1, nome della seconda variabile =vettore2,...)
```

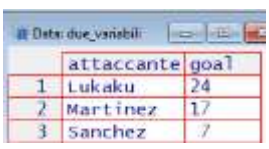
```
due_variabili<-data.frame(attaccante=attaccanti, goal=goal)
class(due_variabili)
[1] "data.frame
due_variabili
  attaccante goal
1    Lukaku    24
2  Martinez    17
3   Sanchez     7
```

Lukaku ha segnato 24 goal, Martinez 17, Sanchez 7.

Aggiungiamo un **argomento opzionale** che indica a R[5]: "quando trovi un vettore di classe `string`, nel dataframe **convertilo** in un oggetto di classe **`factor`**": `stringsAsFactors= TRUE`. È un argomento logico (gli argomenti `logic` possono assumere solo due valori: `TRUE` o `FALSE`), che quando è settato su `TRUE` esegue l'argomento ("stringhe come fattori"), e quando su `FALSE` non lo esegue. Di default, la funzione `stringsAsFactors` è `= FALSE`, quindi la variabile campione sarebbe importata come un `character`. Per ora, in realtà, poco ci importa, ma dovremo ragionare su `character` e `factor` nel momento in cui faremo analisi inferenziali, per cui abituiamoci a fare i conti con questo argomento.

```
due_variabili<-data.frame(attaccante=attaccanti, goal=goal, stringsAsFactors = TRUE)
```

Possiamo visualizzare la struttura dell'oggetto `data.frame` nella console, come sempre, ma se il dataframe è grosso questa visualizzazione è poco pratica. Possiamo invece usare la funzione **`View(dataframe)`** (attenti all'iniziale maiuscola), che in R apre una nuova finestra in cui il dataframe viene visualizzato come una tabella e in Rstudio attiva la visualizzazione nella finestra in alto a sinistra:



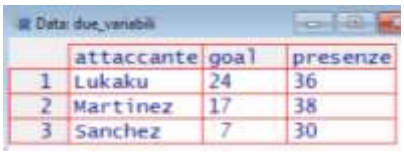
[5] Nelle versioni di R precedenti alla 4.0.0, di default abbiamo `stringAsFactors=TRUE`. Quindi, se state usando una versione vecchia di R, è inutile specificare l'argomento – ma dovreste aggiornare la vostra versione!

Una volta creato un dataframe, con `fix(data.frame)` si può modificarlo: viene aperta una **finestra di editing** in cui possono essere corretti i dati nelle singole celle (clic sulle celle), modificati i nomi e corretto il tipo delle variabili, aggiunte variabili. La presentazione è uguale in R e in Rstudio.

Se vogliamo arricchire il dataframe con la nuova variabile "presenze in campo", ne digitiamo il nome nella cella `var3`, specifichiamo il tipo (`numeric`) e inseriamo le presenze dei tre attaccanti nella cella della riga loro corrispondente:

Una volta chiusa la finestra di editing, la struttura del dataframe si conferma modificata:

```
due_variabili
  campione goal presenze
1   Lukaku   24       36
2 Martinez   17       38
3  Sanchez    7       30
```



Completiamo tutte le informazioni che ci serviranno in seguito, aggiungendo (sempre dati ufficiali) il numero di minuti di gioco, gli assist vincenti e il numero di maglia:

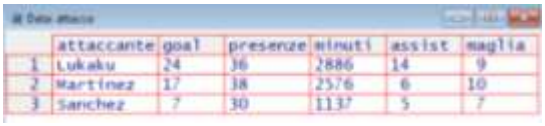
```
minuti<-c(2886,2576,1137)
assist<-c(14,6,5)
maglia<-c(9,10,7)
```

Ora dovremmo unire le tre nuove variabili create al dataframe `attacco`, ma **non possiamo usare `c`, che accetta solo oggetti dello stesso tipo**: noi abbiamo un oggetto `data.frame` e tre variabili / vettori che R classifica come `numeric`:

```
class(due_variabili)
 [1] "data.frame"
class(minuti);class(assist); class(maglia)
[1] "numeric"
[1] "numeric"
[1] "numeric"
```

Quindi, useremo ancora la funzione `data.frame`, che non si fa problemi a unire oggetti di classe diversa:

```
attacco<-data.frame(due_variabili,minuti,assist,maglia)
View(attacco)
```



Quando i vettori vengono uniti in una dataframe che li collega in un insieme, oltre al nome **acquistano una sorta di "nome di famiglia", un "cognome", rappresentato dal nome del dataframe** che li contiene: ora R li riconosce **solo se vengono correttamente indicati** per "**cognome e nome**", separati da **$**:

Nome del dataframe *Nome della variabile*

```
mean(attacco$presenze)
[1] 34.66667
```

Attenzione: naturalmente, i vettori che avete unito al dataframe **non sono stati eliminati**. Nello spazio di R convivono `minuti` e `attacco$minuti`, `assist` e `attacco$assist`, `maglia` e `attacco$maglia`... finché non li cancellerete intenzionalmente con `rm(oggetto)`. Altrettanto naturalmente, R li considera cose del tutto diverse, per cui un'eventuale correzione al numero di maglia di Lukaku fatta nel vettore `maglia` non si tradurrà in un analogo cambiamento nella variabile `attacco$maglia`.

Specularmente, **le variabili di un dataframe possono essere estratte dalla struttura e rese vettori indipendenti**.
Per esempio, possiamo tornare a:

```
presenze_bis<-attacco$presenze
presenze_bis
[1] 36 38 30
```

Una funzione alternativa per creare strutture organizzate di vettori può essere **`cbind`** – *columns bind, collega colonne* - che unisce vettori, dataframe o matrici (classe che vedremo subito) disponendo i loro elementi **per colonne** e creando oggetti di classe `data.frame`; anche `cbind` si applica a oggetti di classe uguale o diversa, purché siano **composti dallo stesso numero di osservazioni / righe**[6].

```
attacco<-cbind(due_variabili,minuti, assist, maglia)
class(attacco)
[1] "data.frame"
```

La funzione **`rbind`** – *rows bind, collega righe* - unisce oggetti disponendo i loro elementi **per riga,** secondo lo stesso principio.
Vediamo le diverse strutture che `rbind` e `cbind` formano quando sono applicati agli stessi oggetti. Creiamo 3 vettori, composti dallo stesso numero di numeri interi consecutivi: da 1 a 5, da 10 a 14, da 20 a 24. Potremmo usare `c`, ma è più veloce usare il segno **`:`** che indica a R "**usa tutti i valori compresi tra… e…"**:

```
A<-(1:5)
B<-(10:14)   →   cbind(A, B, C)
C<-(20:24)           A  B  C
                 [1,] 1 10 20
                 [2,] 2 11 21
                 [3,] 3 12 22
                 [4,] 4 13 23
                 [5,] 5 14 24
```

```
rbind(A, B, C)
  [,1] [,2] [,3] [,4] [,5]
A    1    2    3    4    5
B   10   11   12   13   14
C   20   21   22   23   24
```

2.2.1 Descrivere il dataframe

`View(data.frame)` ha un aspetto rassicurante – perché è familiare – per descrivere il contenuto di un dataframe, ma in effetti è poco informativo. Più informazioni, in un layout piuttosto compatto, si hanno con **`str(oggetto)`**: applicata a un dataframe, descrive la sua **struttura**, cioè il **numero** di osservazioni, **quante** variabili contiene, la loro **classe**, i **primi casi** di ogni variabile:

```
str(attacco)
'data.frame':   3 obs. of  6 variables:
 $ attaccante: Factor w/ 3 levels "Lukaku","Martinez",..: 1 2 3
 $ goal      : num  24 17 7
 $ presenze  : num  36 38 30
 $ minuti    : num  2886 2576 1137
 $ assist    : num  14 6 5
 $ maglia    : num  9 10 7
```

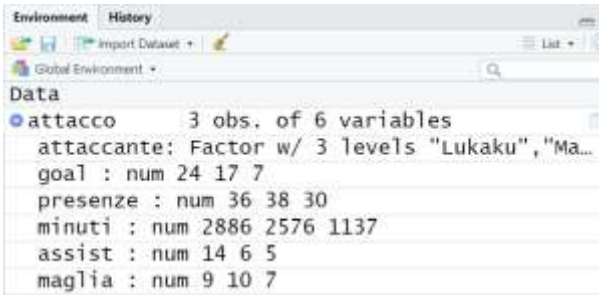
Può descrivere anche uno degli elementi del dataframe:

```
str(attacco$maglia)
num [1:3] 9 10 7
```

Se state usando **Rstudio**, la struttura del dataframe compare in `Global Environment`: basta cliccare sul triangolino a fianco del nome del dataframe.

[6]Se uno degli elementi combinati ha meno osservazioni di un altro, R lo segnala con un *warning*, e nella colonna deficitaria dell'oggetto creato vengono ciclicamente ripetute le osservazioni presenti fino a colmare il gap – il che è evidentemente insensato.

Questa visualizzazione "fotografa" lo stato del dataframe nel momento in cui si apre: per visualizzare qui i cambiamenti nel dataframe man mano che avvengono è necessario fare un *refresh* (ri-cliccando l'icona). L'elenco `Data` del foglio `Environment` non è direttamente editabile, con l'eccezione di alcuni oggetti di lunghezza unitaria (ad esempio `TRUE` o `FALSE`), che possono essere modificati.



Vediamo una curiosità, che ci preoccuperà nei disegni con variabili indipendenti. La variabile `$attaccante` è classificata come **fattore con 3 livelli** (`Factor w/3 levels`), che vengono indicati nel loro **ordine alfabetico** (**L**ukaku-**M**artinez-**S**anchez → 1-2-3)**.** R, infatti, **identifica i livelli secondo il loro ordine crescente**, **alfabetico** se espressi come testo, **numerico** se espressi come numeri.

L'informazione di `str` si legge quindi:

Livello 1- Lukaku *Livello 3- Sanchez*

```
$ attaccante: Factor w/ 3 levels "Lukaku","Martinez",..: 1  2  3
```

Livello 2- Martinez

Molte informazioni per tutte le variabili presenti si ottengono dalla funzione **`summary(oggetto)`**: l'useremo **estremamente spesso**. I suoi oggetti possono essere singoli vettori, strutture righe x colonne, modelli statistici, ecc., e gli output che restituisce variano a seconda della classe dell'oggetto cui si applica. Nei dataframe, `summary` dà la **frequenza** di ogni elemento `character` o `factor`, e alcune **statistiche** per le variabili `numeric`, che vedremo nel capitolo 3, ma probabilmente conoscete già: i valori minimo e massimo, il 1° e il 3° quartile, mediana e media.

Notate un incolpevole fraintendimento di R, di cui ci occuperemo nel §3.2.3: le statistiche ordinali (quartili, mediana) e intervallari (media) che il `summary` fornisce per la **variabile `$maglia`** non hanno senso, perché questa variabile è in **realtà nominale** (i numeri di maglia sono solo etichette qualitative, non operazionalizzano reali quantità né esprimono una gerarchia). Dovremo perciò costringere R a tornare sui suoi passi, correggendo la classe di `$maglia`.

```
summary(attacco)
     attaccante          goal          presenze         minuti          assist          maglia
 Lukaku  :1     Min.   : 7.0   Min.   :30.00   Min.   :1137   Min.   : 5.000   Min.   : 7.000
 Martinez:1     1st Qu.:12.0   1st Qu.:33.00   1st Qu.:1856   1st Qu.: 5.500   1st Qu.: 8.000
 Sanchez :1     Median :17.0   Median :36.00   Median :2576   Median : 6.000   Median : 9.000
                Mean   :16.0   Mean   :34.67   Mean   :2200   Mean   : 8.333   Mean   : 8.667
                3rd Qu.:20.5   3rd Qu.:37.00   3rd Qu.:2731   3rd Qu.:10.000   3rd Qu.: 9.500
                Max.   :24.0   Max.   :38.00   Max.   :2886   Max.   :14.000   Max.   :10.000
```

Naturalmente, oltre a queste visioni d'insieme della struttura, ci può essere la necessità di concentrare l'attenzione su sue parti più specifiche. Nella **struttura** complessa di un dataframe(o di una matrice), **ogni elemento** dello stesso è **identificato dalla cella che occupa**, ovvero dal **numero di riga** (*row*) **combinato** con il **numero della colonna** (*column*). Come anticipato, i comandi che riguardano la struttura di oggetti di questo tipo devono essere dati **usando le parentesi quadre**: all'interno delle **`[]`**, il **primo elemento indica la riga**, il **secondo indica la colonna**. Per esempio, se vogliamo sapere quanti goal (colonna 2) ha segnato Sanchez (riga 3), chiederemo:

nel dataframe `attacco`, *qual è il dato della riga 3* *incrociata con la colonna 2?*

```
attacco[ 3,      2]
[1] 7
```

La risposta è che nella cella corrispondente a riga-colonna indicata è presente "7", cioè sette goal.

Per sapere quali sono i dati corrispondenti a **righe consecutive** in corrispondenza di una sola colonna, per esempio quanti goal (colonna 2) hanno segnato Lukaku (riga 1) e Martinez (riga 2), si separano le righe con `:` , scrivendo:

nel dataframe `attacco`, ↓ *quali sono i dati delle righe tra 1 e 2 ↓* *incrociate con la colonna 2? ↓*

```
attacco[ 1:2,        2]
[1] 24 17
```

Quindi, 24 goal (colonna 2) per la riga 1 (Lukaku) e 17 goal per la riga 2 (Martinez).

Lo stesso procedimento si applica per **colonne consecutive**, per esempio associando i goal (colonna 2) e le presenze in campo (colonna 3) del solo Sanchez (riga 3):

nel dataframe `attacco`, ↓ *quali sono i dati della riga 3 ↓* *nelle colonne comprese tra 2 e 3? ↓*

```
attacco[ 3,          2:3]
  goal presenze
3    7       30
```

Se interessano informazioni in colonne (o in righe) **non consecutive**, si concatenano gli elementi dell'argomento colonna (o dell'argomento riga) usando la solita funzione `c`. Quindi, per conoscere anche gli assist (colonna 5), oltre che i goal, di Sanchez, chiediamo:

nel dataframe `attacco`, ↓ *quali sono i dati della riga 3 ↓* *nelle colonne 2, 3, e 5? ↓*

```
attacco[3,   c(2, 3, 5)]
  goal presenze assist
3    7       30      5
```

Eccetera eccetera, in tutte le possibili combinazioni:

```
attacco[c(1,3),c(2,5)]
  goal assist
1   24     14
3    7      5
```

Se vogliamo fare operazioni su **tutte le righe** e/o su **tutte le colonne**, basta **lasciare libero lo spazio prima della virgola** ("tutte le righe") **o dopo la virgola** ("tutte le colonne"): R interpreta questa istruzione come "applica quanto richiesto a tutte le righe / a tutte le colonne". Per esempio, per vedere i numeri di maglia (colonna 6) di tutti gli attaccanti, possiamo chiedere:

```
attacco[,6]
[1] 9 10 7
```

Invece, per vedere tutte le variabili riferite a Lukaku (riga 1), faremo:

```
attacco[1,]
  campione goal presenze minuti assist maglia
1   Lukaku   24       36   2886     14      9
```

Le informazioni sulla struttura possono essere richieste in modo ulteriormente creativo, per esempio chiedendo di visualizzare tutti gli elementi di una colonna **tranne** quelli corrispondenti a una o più righe, oppure gli elementi di una riga tranne quelli corrispondenti a una o più colonne, o gli elementi dell'intera struttura, **tranne** gli elementi corrispondenti a una o più righe e colonne. L'informazione **tranne** è trasmessa a R **anteponendo il segno meno –** al numero di riga o di colonna da omettere:

```
attacco[-3, 4:5]
  minuti assist
1   2886     14
2   2576      6
```

```
attacco[c(-2,-3), 3:4]
  presenze minuti
1       36   2886
```

Ricordiamo che questa operazione non modifica il dataframe, ma seleziona solo le informazioni da visualizzare.

2.2.2 Liste, matrici e tibble

Altri oggetti che in R contengono e strutturano altri oggetti sono **liste**, create dalla funzione `list(oggetto)` e che possono contenere oggetti di classi diversa, e **matrici**, create dalla funzione `matrix`. Le liste sono elenchi non strutturati in righe e colonne di oggetti; troveremo un esempio di lista quando affronteremo le funzioni usate per i test di significatività (§6.5, §7.2, ecc.): queste funzioni possono essere salvate come oggetti (di classe `htest`, `lm`, `glm`), i cui ciascuno dei quali è una lista degli elementi che compongono le parti dell'output del test. Anticipiamone un esempio. Creiamo l'oggetto `goal_minuti` in cui sono contenute le informazioni di una correlazione lineare tra i minuti giocati e i goal segnati (la funzione per fare la correlazione è `cor.test(variabile1, variabile2)`; cap. 9). L'oggetto è di classe `htest` e contiene la lista delle informazioni del modello lineare: il coefficiente di correlazione, il $p - value$ per la significatività, la statistica del test di correlazione di Pearson, eccetera, come vedremo in Tecniche di Analisi di Dati II.

```
goal_minuti<-cor.test(attacco$goal, attacco$minuti)
class(goal_minuti)
[1] "htest"
```

```
goal_minuti$
  statistic
  parameter
  p.value
  estimate
  null.value
  alternative
```

Per i curiosi: la relazione esiste, è positiva e fortissima; più si gioca, più si segna (e viceversa).

```
goal_minuti$estimate
      cor
0.9676114
```

Useremo un po' più spesso le matrici, perché alcune funzioni per specifiche analisi lavorano solo su oggetti di classe `matrix` (per esempio, `friedman.test`, `rcorr` del package `Hmisc`, i grafici e i test per l'analisi della normalità multivariata, eccetera; d'altro canto, il package per grafici `ggplot2` lavora solo su dataframe).

Rispetto ai dataframe, la particolarità delle **matrici** è che devono **essere composte da variabili o solo numeriche o solo testuali**; la loro costruzione obbliga a definire, oltre a **quali sono le variabili** che compongono, anche di **quante righe sono composte** e **quante colonne** costituiranno la matrice. Richiedono però meno memoria e sono più efficienti per calcoli complessi. Quindi, mentre lo scopo dei dataframe è essenzialmente immagazzinare dati in maniera organizzata e facile da comprendere, quello delle matrici è lavorarci su.

Per esempio, creiamo un oggetto in cui siano contenute solo le misure di effettivo **rendimento** degli attaccanti, ovvero i goal e gli assist vincenti, entrambe numeriche. Chiediamo a R di comporre la matrice **`rendimento`**:

```
rendimento  <-  matrix(data=c(goal, assist), nrow=3, ncol=2)
```

L'oggetto `rendimento` | è creato | da una matrice | che concatena i vettori `goal` e `assist`; | fatta da 3 righe | e da 2 colonne

L'aspetto della matrice è analogo a quello di un dataframe; dovremmo però **aggiungere i nomi delle due variabili**, per identificarle senza ambiguità: possiamo usare `fix(rendimento)`, come per il dataframe precedente, oppure imparare funzioni che ritroveremo spesso: `rownames(matrice)` e `colnames(matrice)`.

Data: rendim...

	V1	V2
1	24	14
2	17	6
3	7	5

Usando la freccia di assegnazione, si assegnano i **nomi che identificano le righe** (`rownames`) o le **colonne** (`colnames`), disposti nella sequenza desiderata con la funzione `c`. Nel nostro caso (attenti alle " "):

```
colnames(rendimento)  <-  c("goal", "assist")
```

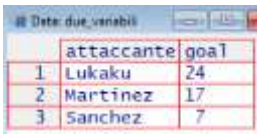
Come nomi delle colonne dell'oggetto `rendimento` | assegna | la combinazione delle stringhe "goal" e "assist"

Data: rendime...

	goal	assist
1	24	14
2	17	6
3	7	5

Se invece volete cambiare i nomi delle variabili di un oggetto di classe `data.frame`, usate `names(oggetto)`, una funzione ancora più onnipresente. Usata **da sola**, restituisce in output i nomi degli elementi che compongono l'oggetto; **insieme a** ←, serve per assegnare i nomi agli elementi dell'oggetto.

Per esempio, riprendiamo il vecchio dataframe `due_variabili`:



	attaccante	goal
1	Lukaku	24
2	Martinez	17
3	Sanchez	7

e decidiamo di chiamare la prima colonna "uno" e la seconda "due". Però, prima, per non sovrascrivere `due_variabili`, che potrebbe servirci ancora, **creiamo un nuovo dataframe**, che chiamiamo **due**, con le stesse caratteristiche di `due_variabili`:

```
due<-due_variabili
```

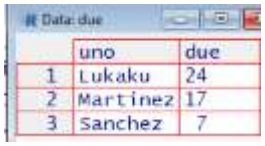
Per sapere **quali sono i nomi delle variabili in due**, usiamo:

```
names(due)
[1] "campione" "goal"
```

Invece, **per cambiare i nomi delle variabili**, usiamo:

```
names(due)   <-   c("uno", "due")
```

Come nomi degli elementi dell'oggetto due — *è assegnata* — *la combinazione delle stringhe "uno" e "due"*



	uno	due
1	Lukaku	24
2	Martinez	17
3	Sanchez	7

Alla matrice `rendimento` ora possiamo applicare la statistica di Friedman, un test non parametrico che useremo per sapere se esistono differenze non casuali tra più misure prese su uno stesso soggetto – nel nostro esempio, per sapere se il numero di goal fatti dal nostro reparto supera in maniera non casuale il numero di assist vincenti.

ATTENZIONE, SPOILER: forse qualcuno lo saprà già interpretare, ma tutti lo vedremo a suo tempo: il numero di goal fatti è solo casualmente diverso dal numero di assist degli attaccanti… ma con qualche incertezza:

```
friedman.test(rendimento)
        Friedman rank sum test
data:  rendimento
Friedman chi-squared = 3, df = 1, p-value = 0.08326
```

Se, incautamente, inseriamo in una matrice una variabile `character` oltre a quelle numeriche (`numeric` o `integer`), R, zitto zitto, cambia **tutte le variabili `numeric` in `character`**:

```
A<-1:5
D<-c("uno","due","tre","quattro","cinque")
class(A);class(D)
[1] "integer"
[1] "character"

mista<-matrix(c(A,D),5,2)
mista
     [,1] [,2]
[1,] "1"  "uno"
[2,] "2"  "due"
[3,] "3"  "tre"
[4,] "4"  "quattro"
[5,] "5"  "cinque"
```

Le virgolette ci dicono che la prima colonna della matrice `mista` (A) non è più `numeric`, ma `character`. Possiamo chiedere conferma di quale sia la classe del primo elemento dell'oggetto `mista`, usando `[1]`:

```
class(mista[1])
[1] "character
```

Le differenze tra dataframe e matrici sono in realtà più profonde, e altre considerazioni possono essere fatte su cosa distingue i due oggetti e ne rende più o meno conveniente l'uso (si devono usare matrici per fare operazioni di algebra

lineare, come in MANOVA; le matrici hanno un uso più efficiente della memoria per dataframe estremamente ampi; i dataframe sono più flessibili e chiari), ma si rimandano i curiosi ai diversi materiali online: per il livello richiesto nell'esame, ci fermiamo qui,

Per concludere l'elenco delle strutture di dati in R, citiamo i **`tibble`** (ne vedremo un esempio in Tecniche di analisi di dati II, §12.1.2), che sono un'aggiunta più recente e non fanno parte delle statistiche di base, ma sono usati in diversi package aggiuntivi: `tibble`, per esempio. Un `tibble` è un dataframe che perde alcune caratteristiche (per esempio, non consente di cambiare il nome e la classe delle sue variabili) e ne aggiunge altre che possono renderlo più "appetibile" per calcoli complessi. Ne vediamo solo un esempio, per completezza: recuperiamo le distribuzioni `A` e `D` della matrice e aggiungiamo una variabile `X` di classe `Date`. Per creare un `tibble`, le raggruppiamo in un dataframe che trasformiamo con la funzione `as_tibble(dataframe)` del package `tibble`.

```
ADX<-as_tibble(data.frame(A, D, X))
ADX
# A tibble: 5 x 3
      A D       X
  <int> <chr>   <date>
1     1 uno     2021-01-01
2     2 due     2021-02-02
3     3 tre     2021-03-03
4     4 quattro 2021-04-04
5     5 cinque  2021-05-05
```

I `tibble` mostrano al massimo le prime dieci righe della struttura, e solo le colonne che si adattano alla dimensione della finestra; come in `str(dataframe)`, viene sempre mostrata la loro classe (che, ricordiamo, non può essere cambiata).

Solo per chi si sta appassionando: i `tibble` nascono nel mondo `tidyverse` (https://www.tidyverse.org/), che è un collettore di ottimi package (ne citeremo un altro, `ggplot2`, che si occupa di grafici). Se volete approfondire, cliccate sul link.

2.2.3 Rinominare le variabili

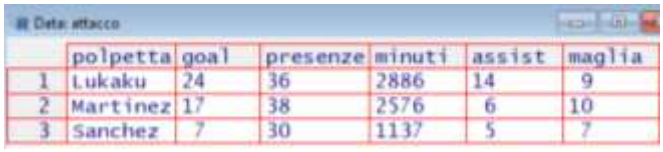
Ci sono un paio di cose da sistemare nel dataframe `attacco`. La prima è opzionale: ripensandoci, il **nome** della variabile `$attaccante` è ridondante rispetto al nome del dataframe (sono tutti attaccanti) e può fare confusione con il nome del dataframe stesso. Scegliamo, allora, di **rinominare** questa variabile in "`campione`": consolidiamo l'uso di **`names`**:

```
names(attacco)<-c("campione","goal","presenze","minuti","assist","maglia")
names(attacco)
[1] "campione" "goal"     "presenze" "minuti"   "assist"   "maglia"
```

Dato che si cambia un solo nome, sarebbe comodo **anche usare la struttura del dataframe** per indicare l'unica sostituzione:

```
names(attacco)[1]    <-      "polpetta"
```

Al nome del primo elemento della struttura `attacco` è assegnato il testo "`polpetta`"


Data: attacco

	polpetta	goal	presenze	minuti	assist	maglia
1	Lukaku	24	36	2886	14	9
2	Martinez	17	38	2576	6	10
3	Sanchez	7	30	1137	5	7

… ma torniamo a definizioni più consone:

```
names(attacco)<-c("campione","goal","presenze","minuti","assist","maglia")
```

2.2.4 Riattribuire le classi alle variabili

Sistemiamo ora la seconda cosa che non va nel dataset, più seria: riguarda le variabili `attacco$campione` e `attacco$maglia`. Rivediamo la classe attribuita a queste variabili:

```
str(attacco)
'data.frame':   3 obs. of  6 variables:
 $ campione: Factor w/ 3 levels "Lukaku","Martinez",..: 1 2 3
 …
 …
 $ maglia  : num  9 10 7
```

La classe `numeric` delle altre variabili è corretta: i loro dati sono veri numeri, su scala metrica; non è così invece per i dati della variabile **maglia di gioco**, i cui numeri sono solo etichette alfanumeriche, su scala nominale, prive di una reale qualità aritmetica[7]. La corretta classe di questa variabile dovrebbe essere **`character`**, non `numeric`. Allo stesso modo, i nomi degli attaccanti **non corrispondono a diversi gruppi**[8], dato che non ci sono più giocatori che appartengono allo stesso gruppo chiamato "Lukaku" (anche se talvolta si può aver avuto questa impressione in campo): pure in questo caso, la classe più adeguata è **`character`**, non `factor`.

Abbiamo visto che si può impostare **qualsiasi variabile** composta da stringhe di testo come `factor` o come `character` sin dalla creazione del dataframe, predisponendo nella funzione `data.frame` l'argomento di classe `logic` **`stringsAsFactors=TRUE`** ("i vettori composti da stringhe di testo sono fattori") o **`=FALSE`** (tutte le stringhe sono testuali, di default[9]). Naturalmente, però, è realistico che nello stesso dataframe alcune variabili stringa debbano essere correttamente considerate `factor` e altre come `character`. Correggere gli errori di classe dopo **aver creato il dataframe** è semplice: abbiamo usato `fix(dataframe)`, ma possiamo sfruttare anche **`as.*`**. L'asterisco è **sostituibile da una qualsiasi classe** riconosciuta in R: avremo quindi `as.numeric` per cambiare in valori numerici, `as.character` (quello che ci serve) per cambiare in stringhe di testo, `as.factor` per cambiare in livelli di una variabile di raggruppamento, `as.Date`**`Errore. Il segnalibro non è definito.`** per creare variabili che contengono date, ecc.

La corretta variabile è quindi creata dalla vecchia variabile trasformata come (`as`) numero, carattere, fattore, ecc. Scriveremo:

```
attacco$maglia <- as.character(attacco$maglia)
```

la variabile `$maglia` è creata trasformando in classe `character` la variabile `$maglia`

e anche:

```
attacco$campione<-as.character(attacco$campione)
```

Adesso:

```
str(attacco)
'data.frame':   3 obs. of  6 variables:
 $ campione: chr  "Lukaku" "Martinez" "Sanchez"
 $ goal    : num  24 17 7
 $ presenze: num  36 38 30
 $ minuti  : num  2886 2576 1137
 $ assist  : num  14 6 5
 $ maglia  : chr  "9" "10" "7"
```

Per **correggere valori di cella** errati, invece, basta assegnare il valore di cella corretto indicando il valore da sostituire con le coordinate di riga – colonna.

Per esempio, se venisse finalmente riconosciuto a Lukaku quel goal che **non** era in fuorigioco, correggeremmo i suoi 24 goal scrivendo:

```
attacco[1,2]<-25
```

	campione	goal
1	Lukaku	25
2	Martinez	17
3	Sanchez	7

[7] Ripasseremo le scale di misura nel capitolo 3

[8] In realtà, ogni soggetto può essere considerato un gruppo di ampiezza N= 1, e quindi la variabile che li contiene può essere davvero un `factor`; nei dataframe in long format delle misure ripetute, in cui ogni riga corrisponde a una misura presa dallo stesso soggetto, questa variabile dovrà davvero essere di classe `factor`. Per ora, nei disegni trasversali che stiamo affrontando, convertiamo da `factor` a `charachter`, dato che non useremo questa variabile per fare analisi su gruppo.

[9] `= TRUE` può scriversi anche come `=1`, `= FALSE` anche come `= NULL`

2.2.5 La classe Date

Una classe di oggetti citata un po' frettolosamente è `Date`, che li identifica come date. Se si vogliono fare operazioni che coinvolgono date, per esempio calcolare l'età in anni al momento della ricerca conoscendo l'anno di nascita, o il tempo intercorso tra due date, è essenziale che la variabile sia correttamente codificata: in caso contrario, R la interpreterà come testo.

Il formato di **ogni elemento** di un vettore `Date` deve essere del tipo **aaaa/mm/gg (**anno/mese/giorno); per creare un vettore `Date`, si concatenano gli elementi combinando le funzioni `as.Date` e `c`, così:

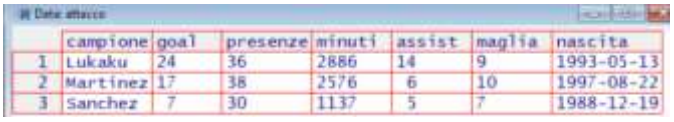
```
nascita <- as.Date (c("1993/05/13", "1997/08/22", "1988/12/19"))
```

Il vettore `nascita` — è creato — trasformando in classe `Date` — la concatenazione di questi elementi

```
class(nascita)
[1] "Date"
```

Visto che l'abbiamo creata, **aggiungiamola** al dataframe:

```
attacco$nascita<-nascita
str(attacco)
'data.frame':   3 obs. of  7 variables:
 $ campione: chr  "Lukaku" "Martinez" "Sanchez"
 $ goal    : num  24 17 7
 $ presenze: num  36 38 30
 $ minuti  : num  2886 2576 1137
 $ assist  : num  14 6 5
 $ maglia  : chr  "9" "10" "7"
 $ nascita : Date, format: "1993-05-13" …
```



	campione	goal	presenze	minuti	assist	maglia	nascita
1	Lukaku	24	36	2886	14	9	1993-05-13
2	Martinez	17	38	2576	6	10	1997-08-22
3	Sanchez	7	30	1137	5	7	1988-12-19

2.2.6 I valori mancanti NA

Come anticipato a pag. 10, quando una cella di una matrice o un dataframe non contiene un dato valido (e, auspicabilmente, il dato manca per motivi del tutto casuali), R lo sostituisce con il codice `NA` (**Not Available**). Per esempio, se conoscessimo il numero di assist di tutti gli attaccanti tranne Sanchez, dovremmo scrivere:

```
assist_mancanti<-c(14,6,NA)
```

Alcune funzioni gestiscono la presenza di `NA` senza che l'essere umano debba specificare come farlo, altre invece necessitano di istruzioni ad hoc. Per esempio, se chiediamo la media del vettore `assist_mancanti`, R non restituisce un output:

```
mean(assist_mancanti)
[1] NA
```

Dobbiamo aggiungere un argomento che dice a R di **togliere i dati mancanti prima di eseguire il calcolo** della media: **`na.rm=TRUE`**, ovvero ***NA remove***; l'operatore logico `TRUE` dice a R di eseguire la rimozione:

```
mean(assist_mancanti, na.rm=TRUE)
[1] 10
```

Vedremo man mano come le diverse funzioni gestiscono i dati mancanti, ove necessario.

All'apertura di un dataframe (§3.4), uno degli argomenti opzionali della funzione `read.delim` o `read.csv` è **`na.strings= "NA"`,** che definisce come mancanti le **celle vuote**. È però plausibile che in alcune occasioni altri tipi di dati debbano essere considerati mancanti: per esempio, in un compito di tempi di reazione semplice un dato superiore a 2000 msec deve essere considerato omissione, e un dato inferiore a 100 msec è un anticipo di risposta. In questi casi, il valore da considerare mancante va inserito in `na.strings= …` . Comunque, è possibile riconvertire a posteriori un dato come mancante con **`dataframe[dataframe==valore] ← NA`**. Per esempio, riteniamo che 5 assist vincenti

siano un risultato impossibile per un giocatore di serie A in un'intera stagione, e che quindi questo dato debba essere erroneo, magari digitato malamente al momento della creazione del dataframe: per prudenza, lo consideriamo *missing*. Possiamo verificare quanti "5" e quanti "NA" siano presenti in una variabile usando una funzione che impareremo a memoria: **`table(oggetto)`**, che restituisce la frequenza assoluta di tutti gli elementi presenti nella variabile; aggiungiamo l'argomento opzionale **`exclude=NULL`**, ovvero istruiamo R a non omettere alcun dato – in caso contrario, i NA non sarebbero mostrati nell'output:

```
table(attacco$assist, exclude=NULL)
5  6 14
1  1  1
```

Nessuna cella vuota, ma un giocatore con solo 5 assist c'è. Possiamo usare la funzione `which(criterio)`, cioè "qual è il caso che soddisfa la condizione... ?" per sapere qual è il caso / la riga con assist = 5:

```
which(attacco$assist==0)
[1] 3
```

È il giocatore i cui dati sono contenuti nella riga 3. Assegniamo a questo opinabile 5 nella terza riga - quinta colonna ($assist) lo status di NA, usando la struttura del dataset:

```
attacco[3,5]<-NA
```

Verifichiamo, enfatizzando la differenza a seconda dell'argomento `exclude`:

```
table(attacco$assist, exclude=NULL)
 6   14 <NA>
 1    1    1
```

```
table(attacco$assist)
 6 14
 1  1
```

Giusto: nessun 5 e un dato mancante.

Per sapere quali siano i casi con dati missing, associamo **`which`** alla funzione **`is.na`**, che **identifica i valori mancanti**: **`which(is.na(dataframe$variabile))`**:

```
which(is.na(attacco$assist))
```

← *qual è che il caso che soddisfa la condizione "è un dato mancante" nella variabile assist?"*

```
[1] 3
```

Un solo caso, corrispondente alla riga 3.

Torniamo ad assegnare i 5 assist a Sanchez:

```
attacco[3,5]<-5
```

Prima di proseguire con la teoria, un esercizio.

Create il dataframe "centrocampo" *usando le seguenti informazioni:*

I centrocampisti sono: Hakimi, Perisic, Vidal, Barella e Brozovic; i loro numeri di maglia sono: 2, 14, 22, 23, 77. Hanno fatto rispettivamente 7, 4, 1, 3 e 2 goal. Hanno registrato 37, 32, 23, 36 e 33 presenze in campo. Hanno giocato rispettivamente per 2672, 1800, 1142, 2900 e 2584 minuti, facendo 8, 4, 1, 7, e 6 assist vincenti. Sono nati nei seguenti giorni: 4/11/1998, 2/2/1998, 22/5/1987, 7/2/1997, 16/11/1992.

Il dataframe non dovrà avere gli errori che abbiamo corretto in `attaccanti`: la maglia di gioco e il nome del giocatore devono essere stringhe di testo.

Nel crearlo, considerate che nel ***prossimo paragrafo uniremo i due dataframe attacco e centrocampo*** *in un unico dataframe: questa informazione guiderà in qualche maniera i criteri che seguirete?*

Nel paragrafo successivo illustreremo come fare questa unione tra dataframe, ma in realtà avete già tutti gli strumenti per sapere come si fa, senza andare a leggere: potete provare da soli?

*Lo script per eseguire tutto quanto richiesto è in fondo alla dispensa, ma è **inutile andarlo a vedere senza almeno provarci** (e riprovarci, e riprovarci ☺)*

2.3 Fare operazioni con le variabili

Nell'esercizio precedente avete creato il nuovo dataframe `centrocampo`. Da qui in poi, lavoreremo su un nuovo dataframe, che chiameremo **`avanti`**, che comprende attaccanti e centrocampisti. Crearlo è velocissimo: uniamo per riga (`rbind`) i due dataframe. Ricordate? Abbiamo usato la funzione su tre vettori:

```
A<-(1:5)
B<-(10:14)
C<-(20:24)
```

```
rbind(A, B, C)
  [,1] [,2] [,3] [,4] [,5]
A    1    2    3    4    5
B   10   11   12   13   14
C   20   21   22   23   24
```

Ma è perfettamente funzionale anche su dataframe:

```
avanti<-rbind(attacco,centrocampo)
```

	campione	goal	presenze	minuti	assist	maglia	nascita
1	Lukaku	24	36	2886	14	9	1993-05-13
2	Martinez	17	38	2576	6	10	1997-08-22
3	Sanchez	7	30	1137	NA	7	1988-12-19
4	Hakimi	7	37	2672	8	2	1998-11-04
5	Perisic	4	32	1800	4	14	1998-02-02
6	Vidal	1	23	1142	1	22	1987-05-22
7	Barella	3	36	2900	7	23	1997-02-07
8	Brozovic	2	33	2584	6	77	1992-11-16

```
str(avanti)
'data.frame':   8 obs. of  7 variables:
 $ campione: chr  "Lukaku" "Martinez" "Sanchez" "Hakimi" ...
 $ goal    : num  24 17 7 7 4 1 3 2
 $ presenze: num  36 38 30 37 32 23 36 33
 $ minuti  : num  2886 2576 1137 2672 1800 ...
 $ assist  : num  14 6 NA 8 4 1 7 6
 $ maglia  : chr  "9" "10" "7" "2" ...
 $ nascita : Date, format: "1993-05-13" "1997-08-22" "1988-12-19" "1998-11-04" ...
```

2.3.1 Creare variabili factor

Bene, ma ora abbiamo un problema. Nel dataframe `avanti`, **attaccanti e centrocampisti sono confusi**, senza nessuna caratteristica che li distingua. Abbiamo bisogno di assegnarli al corretto **gruppo di appartenenza**, creando una **nuova variabile di raggruppamento di classe `factor`**, con due livelli corrispondenti ai due reparti di gioco. Per R, le variabili `factor` sono **numeri**, cui è possibile assegnare **etichette** per ricordare meglio il livello: ogni soggetto appartenente a un gruppo/livello avrà lo stesso numero di ogni altro appartenente al medesimo gruppo, che sarà diverso per gli appartenenti a qualsiasi altro gruppo. Nel nostro esempio, i primi tre soggetti del dataframe appartengono al livello – diciamo – 1, dal quarto in poi appartengono al livello numero 2.

Ci sono vari modi per creare una variabile `factor`. Il più intuitivo consiste nel creare la sequenza, correttamente ordinata, delle etichette dei livelli del fattore: questa variabile, di classe `character`, viene poi mutata in classe `factor` con la funzione `as.factor(oggetto)`.

Cominciamo, allora, creando una variabile **`reparto`** composta da tre "attacco" e cinque "centrocampo". Usando la solita funzione `c`, possiamo scegliere tra una modalità nota, ma lenta e inelegante:

```
reparto <- c("attacco", "attacco", "attacco", "centrocampo", "centrocampo", "centrocampo",
 "centrocampo", "centrocampo")
```

oppure una frizzante novità: **`rep(oggetto da replicare, numero di repliche)`**, cioè ***replicate***. Gli argomenti della funzione sono la "cosa" da replicare (per noi, "attacco" e "centrocampo") e il numero di repliche da fare. Il nostro `reparto` è composto unendo tre repliche del livello "attacco" e cinque repliche del livello centrocampo", quindi:

```
reparto   <-  c(  rep("attacco", 3),  rep("centrocampo", 5))
```
La variabile reparto — *è creata* — *concatenando* — *3 repliche di "attacco"* — *e 5 repliche di "centrocampo"*

```
 class(reparto)
[1] "character"
```

Ora la rendiamo fattore:

```
reparto<-as.factor(reparto)
```

Se non temete di fare confusione, **potete anche combinare `as.factor` e `c` in un solo comando**:

```
reparto <- as.factor(c(rep("attacco", 3), rep("centrocampo", 5)))
str(reparto)
 Factor w/ 2 levels "attacco","centrocampo": 1 1 1 2 2 2 2 2
```

Nella struttura sono indicati i livelli (in effetti, numeri) con le etichette rispettivamente assegnate secondo l'ordine **alfanumerico**: **a**ttacco= 1, **c**entrocampo= 2

Usiamo un'altra modalità imparando una nuova funzione. Prima creiamo la variabile **`reparto2`** combinando tre 1 e cinque 2: la variabile risultante è di classe `numeric`:

```
reparto2<-c(rep(1,3), rep(2,5)
```

Ed ecco il nostro oggetto:

```
reparto2
 [1] 1 1 1 2 2 2 2 2
```

Ora lo trasformiamo in fattore usando **`factor`**, che per la prima volta ci impegna un po' sugli argomenti: dobbiamo dire a R **qual è la variabile** da trasformare, **quali sono i suoi livelli**, quali **etichette** intendiamo assegnarle:

```
reparto <- factor(x= reparto2, levels=c(1:2), labels= c("attacco", "centrocampo"))
```
La variabile Reparto — *è creata* — *rendendo fattore* — *il vettore reparto,* — *che avrà due livelli* — *cui assegniamo queste etichette*

```
reparto
 [1] attacco    attacco    attacco    centrocampo centrocampo centrocampo centrocampo centrocampo
Levels: attacco centrocampo
str(reparto)
 Factor w/ 2 levels "attacco","centrocampo": 1 1 1 2 2 2 2 2
```

Concludiamo inserendo il vettore `reparto` nel dataframe `avanti`, assegnando le sue proprietà alla variabile `$reparto`:

```
avanti$reparto<-reparto
```

Naturalmente, avremmo potuto creare l'oggetto come variabile del dataframe direttamente – e così faremo d'ora in poi:

```
avanti$reparto <- as.factor(c(rep("attacco", 3), rep("centrocampo", 5)))
str(avanti)
'data.frame':   8 obs. of  8 variables:
 $ campione: chr  "Lukaku" "Martinez" "Sanchez" "Hakimi" ...
 $ goal    : num  24 17 7 7 4 1 3 2
 $ presenze: num  36 38 30 37 32 23 36 33
 $ minuti  : num  2886 2576 1137 2672 1800 ...
 $ assist  : num  14 6 NA 8 4 1 7 6
 $ maglia  : chr  "9" "10" "7" "2" ...
 $ nascita : Date, format: "1993-05-13" "1997-08-22" "1988-12-19" "1998-11-04" ...
 $ reparto : Factor w/ 2 levels "attacco","centrocampo": 1 1 1 2 2 2 2 2
```

Vedremo molto, molto più avanti (parlando dell'analisi della varianza) come ulteriormente gestire i livelli delle variabili `factor`.

2.3.2 Creare o modificare variabili da variabili esistenti

Ogni vettore / variabile può essere trasformato associandolo a un altro vettore / variabile, o a un unico valore scalare, tramite i soliti operatori matematici (+, -, *, /, ecc.) e/o gli operatori logici, che sono elencati nella Tabella 1.

Tabella 1. I principali operatori di R

Operatore	Cosa fa
+	Somma
-	Sottrae
*	Moltiplica *(per moltiplicare matrici: %*%)*
/	Divide
^ (o **)	Esponente: eleva a potenza (x^2 significa x^2, x**3 significa x^3, ecc.)
<	Minore di
<=	Minore o uguale a
>	Maggiore di
>=	Maggiore o uguale a
==	Esattamente uguale a (**attenzione, questo si sbaglia facilmente!**)
!=	Non uguale a
!x	Non *x*
x | y	**O**, nel senso di *vel*: l'operazione richiesta fornisce i valori che soddisfano la caratteristica X o la caratteristica Y, ma anche entrambe
x & y	**E**: l'operazione richiesta fornisce i valori che soddisfano sia la caratteristica X sia la caratteristica Y, ma non quelli che hanno una sola delle due:
isTRUE(x)	Verifica se x è TRUE

Per esempio, vogliamo creare la **variabile** `$rendimento` (attenti a non confonderla con la **matrice** `rendimento` che abbiamo creato prima), data dalla somma delle azioni finalizzate con successo (**goal + assist vincenti**). La nuova variabile è quindi la **somma** delle altre due:

```
avanti$rendimento <- avanti$goal + avanti$assist
avanti$rendimento
[1] 38 23 12 15  8  2 10  8
```

Se dovete sommare parecchie variabili ed è scomodo scriverle tutte, usate `rowSums(dataframe[,colonne da sommare])`:

```
avanti$rendimento <- rowSums(avanti[,c(2,5)])
[1] 38 23 12 15  8  2 10  8
```

Attenti a non confondere `rowSums` con la funzione di base `rowsum(x=matrice, group= fattore)`, che serve ad altro: restituisce in output la somma di più colonne (riunite in una matrice) per ciascun livello di una variabile di raggruppamento (`group=`). Ad esempio, se vogliamo sapere quanti goal e quanti assist hanno fatto gli attaccanti e i centrocampisti, prima creiamo la matrice che contiene queste due variabili, poi inseriamo in `rowsum` il vettore che li separa nei due gruppi (tre attaccanti, cinque centrocampisti):

```
rendimento2<-matrix(data=c(avanti$goal, avanti$assist), 8, 2)
colnames(rendimento2)<-c("goal", "assist")
rowsum(x = rendimento2, group = c(rep("attaccanti",3), rep("centrocampisti",5)))
               goal assist
attaccanti       48     25
centrocampisti   17     26
```

Per vedere se i conti di `RowSums` sono giusti, usiamo due nuove funzioni di visualizzazione: **`head(oggetto)`** e **`tail(oggetto)`**. La prima mostra solo le prime righe del dataframe (o di un vettore, matrice, tabella, funzione), la seconda le ultime: di default, il numero di righe visualizzate è **6**: per visualizzarne di più o di meno, se ne aggiunge il numero con **`n=…`** come secondo argomento. Scegliamo di vedere i primi 3 e gli ultimi 3 soggetti:

```
head(avanti$goal, n= 3); head(avanti$assist,3);head(avanti$rendimento,3)
[1] 24  17  7       ← goal
[1] 14   6  5       ← assist
```

```
[1] 38  23 12        ← efficienza
tail(avanti$goal, n=3); tail(avanti$assist,3);tail(avanti$rendimento,3)
[1] 1  3  2
[1] 1  7  6
[1] 2 10  8
```

Invece dei minuti di gioco, **cambiamo l'unità di misura** e otteniamo le **ore di gioco**, dividendo la variabile `$minuti` per 60: facciamo quindi un'operazione tra un vettore e uno scalare:

```
avanti$ore_gioco<-avanti$minuti / 60
head(avanti$minuti,3); head(avanti$ore_gioco,3)
[1] 2886 2576 1137                ← minuti
[1] 48.10000 42.93333 18.95000    ← ore gioco
```

Usiamo la variabile creata `$ore_gioco` per calcolare un altro indicatore di qualità di gioco: rapportiamo il rendimento alle ore di gioco, creando la variabile **`$efficienza`** (lo stesso rendimento in chi gioca di meno, rappresentato da un più alto valore in questa variabile, merita uno stipendio maggiore)

```
avanti$efficienza<-avanti$rendimento / avanti$ore_gioco
```

Impariamo un'altra funzione nel visualizzare questo output: poiché troppi decimali affaticano la leggibilità dell'interpretazione senza arricchirla, **arrotondiamo la visualizzazione** dei primi tre casi, cioè `head(avanti$efficienza,3)`, usando solo due decimali; si usa la funzione **`round(oggetto di cui arrotondare l'output, numero di decimali)`**, facendo attenzione alle parentesi:

```
round(head(avanti$rendimento,3),2);round(head(avanti$ore_gioco,3),2);round(head(avanti$efficienz
a,3),2)
[1] 38  23 12             ← rendimento
[1] 48.10 42.93 18.95      ← ore di gioco
[1] 0.79 0.54 0.63         ← efficienza
```

Alleniamoci un po': vogliamo sapere **chi sono i giocatori con la maggiore e la minore efficienza**. Vediamo il massimo e il minimo di questa variabile:

```
max(avanti$efficienza); min(avanti$efficienza)
[1] 0.7900208
[1] 0.1050788
```

Quali righe occupano i calciatori che corrispondono a questo massimo e a questo minimo?

```
which(avanti$efficienza>.79); which(avanti$efficienza<.11)
[1] 1
[1] 6
```

E chi sono i soggetti nella riga 1 e nella riga 6?

```
avanti[c(1,6),1]
[1] "Lukaku" "Vidal"
```

Infine, lavoriamo con la variabile `$nascita` (di classe `Date`) per ottenere **l'età dei giocatori**. L'età è calcolabile come la quantità di giorni trascorsi da quello di nascita (la nostra variabile) alla convenzionale data in cui si è conclusa la stagione 2020-2021; creiamo prima questa variabile, cioè un oggetto di classe **`Date`**:

```
data_fine_stagione<-as.Date("2021/05/30")
class(data_di_oggi)
[1] "Date"
```

Ora creiamo la variabile `$eta` come **differenza tra la variabile `$nascita` e l'oggetto `data_di_oggi`**; poiché è usuale interpretare l'età degli adulti in anni, e non in giorni, **dividiamo per 365** questa differenza espressa come numero di giorni, così da trasformarla in **anni**:

```
avanti$eta<-(data_fine_stagione-avanti$nascita)/365
```

Per visualizzare gli anni come interi, senza decimali, usiamo ancora `round`, **senza specificare il secondo argomento, cioè il numero di decimali**: di default, saranno restituiti 0 decimali:

```
round(avanti$eta)
Time differences in days
[1] 28 24 32 23 23 34 24 29
```

La classe della variabile è nuova per noi: **`difftime`**, ovvero una differenza tra epoche. A noi serve di tipo `numeric`, però: trasformiamola.

```
avanti$eta<-as.numeric(avanti$eta)
class(avanti$eta)
[1] "numeric"
```

Abbiamo usato `summary` per un oggetto di classe `data.frame`; vediamo per la prima volta `summary` applicato a un oggetto di classe `numeric`: lo useremo tantissimo.

```
summary(avanti$eta)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  22.58   23.67   26.19   27.15   29.53   34.05
```

L'età media del reparto avanzato (`Mean`) è pari a 27.2 anni, mentre la mediana (`Median`) è pari a 26.2 anni; il 50% dei calciatori (range interquartilico) ha un'età compresa tra 23.7 (`1st Qu` – primo quartile) e 29.5 anni (`3rd Qu` – terzo quartile). Il calciatore più giovane ha 22.6 anni (`Min`), il più anziano 34.1 anni (`Max`).

Medie, mediane e quartili dovrebbero esservi noti dalla triennale: li rivedremo, comunque, nel capitolo 3.

2.3.3 Esportare un dataframe

Con tutte le operazioni precedenti, abbiamo decisamente ingrandito il nostro dataframe:

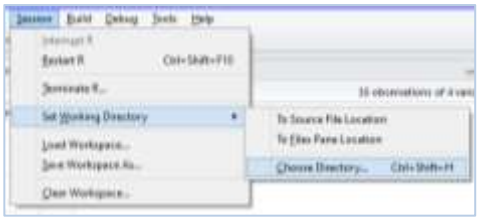
```
str(avanti)
'data.frame':8 obs. of  12 variables:
 $ campione  : chr  "Lukaku" "Martinez" "Sanchez" "Hakimi" ...
 $ goal      : num  24 17 7 7 4 1 3 2
 $ presenze  : num  36 38 30 37 32 23 36 33
 $ minuti    : num  2886 2576 1137 2672 1800 ...
 $ assist    : num  14 6 5 8 4 1 7 6
 $ maglia    : chr  "9" "10" "7" "2" ...
 $ nascita   : Date, format: "1993-05-13" "1997-08-22" ...
 $ reparto   : Factor w/ 2 levels "attacco","centrocampo": 1 1 1 2 2 2 2 2
 $ rendimento: num  38 23 12 15 8 2 10 8
 $ ore_gioco : num  48.1 42.9 18.9 44.5 30 ...
 $ efficienza: num  0.79 0.536 0.633 0.337 0.267 ...
 $ eta       : num  28.1 23.8 32.5 22.6 23.3 ...
```

Vale la pena **salvarlo come file in una delle directory** (cartelle) del vostro computer: se non specificate quale, sarà utilizzata la **cartella predefinita**, che potete chiedere con `getwd` - *get working directory*:

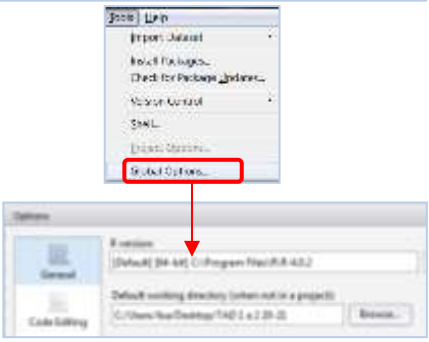
```
getwd()
[1] "C:/Users/lisa/Documents"
```

Se volete specificarne una diversa, potete usare **`setwd`** – *set working directory* - specificando tra le parentesi il nuovo percorso, o **`setwd(choose.dir())`** senza indicare nulla tra le parentesi di `choose.dir`: si aprirà la finestra di navigazione tra le cartelle del vostro sistema (lo stesso risultato si ha con il menu **File → Cambia directory**):

Lavorando con **Rstudio**, per **selezionare la working directory** si può usare il Menu `Session` → `Set Working Directory` → `Choose directory...` e poi via a individuare la cartella con i **comodi** comandi a finestra.



Come detto, a ogni avvio di una sessione di lavoro R fa riferimento alla directory di default, per cui ogni volta bisogna reindirizzarsi alla directory in cui sono contenuti i file che interessano: è possibile **cambiare l'opzione di default**, se i file sono contenuti in una specifica cartella, per evitare di usare `setw` ogni volta. Dal menu `Tools` si seleziona `Global Options` e si indica la nuova directory di default.



Si possono esportare i file creati in R in tanti, tanti formati diversi; a noi potranno servirne di due tipi: un formato in cui i valori sono delimitati da **tabulazioni** (tasto Tab →| ; *tab-delimited file*) o un formato in cui i valori sono **separati da virgole** (**CSV**: *comma-separated values*). Entrambi sono facilmente leggibili da Excel, da editor di testo come Notepad e da un sacco di altri programmi: le informazioni sono immagazzinate come testo, senza formattazioni che possono confondere alcuni software. Quando il file viene caricato in un altro programma – diciamo, per esempio, Excel - questo separerà in colonne diverse i dati separati da tabulazioni o da virgole, a seconda del formato.

Per esportare un dataframe come *tab-delimited*, si crea un oggetto associandovi la funzione **`write.table(nome del dataframe, "nome del file.txt", sep="\t", row.names=FALSE)`**:

```
avanti_tabulato <- write.table(avanti, "avanti_tabulato.txt", sep="\t", row.names=FALSE)
```

↑	↑	↑	↑	↑	↑	↑
il dataframe da salvare	*è creato*	*scrivendo come tabella*	*questo dataframe*	*assegnandogli questo nome*	*il separatore di colonna è TAB*	*Non creare una colonna con i nomi di riga, non servono*

Per salvare un file in formato .csv: **`write.csv(dataframe, "nome del file.csv")`**

```
avanti_virgole <- write.csv(avanti, "avanti con virgole.csv", row.names= FALSE)
```

↑	↑	↑	↑	↑	↑
il dataframe da salvare	*è creato*	*scrivendo come .csv*	*questo dataframe*	*assegnandogli questo nome*	*Non creare una colonna con i nomi di riga, non servono*

Fate attenzione: R, come sempre, sovrascrive oggetti con lo stesso nome, anche quando sono file esportati. Se volete usare una sola cartella per salvare tutti i vostri file (vedremo oltre come cambiare directory di lavoro), accertatevi di non usare un nome già assegnato ad altri dataframe. Potete visualizzare i file esistenti nella directory predefinita con `dir()`, senza specificare nulla tra parentesi, oppure con `list.files()`, tra le cui parentesi potete specificare una diversa cartella (o, come suggerito, scrivere `choose.dir`).

```
list.files()
 [1] "10 - MMPI2 Modelli psicometrici.pptx"
 [2] "2010 - APA 6° edizione.pdf"
 [3] "atlante storico.docx"
 [4] "avanti con virgole.csv"
 [5] "avanti_tabulato.txt"
...
[61] "Video"
[62] "video_out"
[63] "zanetti.docx"
```

Se avete cartelle piene di cose, come quella sopra, potete visualizzare un sottoinsieme di file con l'argomento `pattern=`, in cui dovete indicare un criterio di selezione. Per esempio, se volete sapere solo quali sono i file .txt:

```
list.files(, pattern= ".txt")
[1] "a.txt"                "avanti_completo.txt" "avanti_tabulato.txt" "boh.txt"
[5] "google.txt"           "nota libri.txt"
```

Potete omettere la virgola e indicare solo l'argomento `pattern`:

```
list.files(pattern= ".txt")
```

Oppure, solo quelli che hanno il testo "My" nel nome:

```
list.files(, pattern= "My")
[1] "My Books"             "My Digital Editions" "My Games"            "My Weblog Posts"
```

2.3.4 Selezionare parti di un dataframe

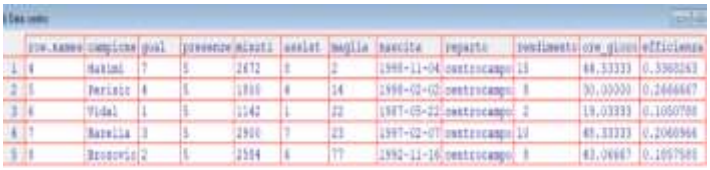
Soprattutto con dataset ampi, con molti soggetti e/o molte variabili (come quelli con cui lavoreremo), può essere utile o necessario **estrarre parti di un dataframe** per lavorare solo su alcuni casi o su alcune variabili. Ci sono più modalità alternative per farlo.

La prima modalità utilizza la struttura `[riga,colonna]` del dataframe, creando un nuovo oggetto di classe `dataframe` che contiene solo i casi e/o solo le colonne necessarie: `nuovo_dataframe←vecchio_dataframe[righe, colonne]`.

Per esempio, selezioniamo solo i centrocampisti del dataset `avanti`, corrispondenti alle righe da 4 a 8, e teniamo tutte le variabili:

```
centro <-  avanti [4:8 ,     ]
```

il dataframe centro è creato *dal dataframe avanti* *di cui seleziona le righe da 4 a 8* *e tieni tutte le colonne*



Abbiamo già visto che se non viene specificato nulla nel posto in cui R si aspetta istruzioni su righe o colonne, R opera su **tutte** le righe o le colonne: in questo caso, trasferisce tutte le colonne nel nuovo oggetto.

Ora selezioniamo solo gli attaccanti usando l'etichetta "attacco" della variabile `factor $reparto` e mantenendo tutte le colonne:

```
att<-avanti[avanti$reparto=="attacco",]
att
  campione goal presenze minuti assist maglia    nascita reparto rendimento ore_gioco efficienza      eta
1   Lukaku   24       36   2886     14      9 1993-05-13 attacco         38  48.10000  0.7900208 28.06575
2 Martinez   17       38   2576      6     10 1997-08-22 attacco         23  42.93333  0.5357143 23.78630
3  Sanchez    7       30   1137      5      7 1988-12-19 attacco         12  18.95000  0.6332454 32.46575
```

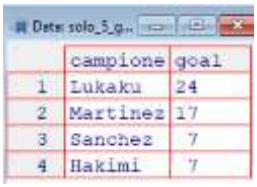
Infine, per usare **goal e presenze in campo dei soli centrocampisti**:

```
goal_presenze_centrocampo<-avanti[4:8,2:3]
goal_presenze_centrocampo
  goal presenze
4    7       37
5    4       32
6    1       23
7    3       36
8    2       33
```

Invece della struttura, si può usare la funzione `subset`: `nuovo dataframe← subset(x= vecchio dataframe, subset= casi da mantenere, select=c(colonne da mantenere))`. `subset` estrae le righe corrispondenti ai soggetti desiderati, che vengono specificati nel suo argomento `subset=casi`; l'argomento `select`, opzionale, serve per conservare solo alcune colonne nel nuovo dataframe: se non viene specificato, tutte le colonne sono importate nel subset. I casi da esportare nel subset si identificano usando – talvolta in maniera creativa – gli operatori logici sulle caratteristiche d'interesse (Tabella 1).

Selezioniamo solo i marcatori con **almeno 5 goal all'attivo,** e visualizziamone solo nome e goal:

```
solo_5_goal<-subset(x= avanti, subset= goal>=5, select=c(campione, goal))
```

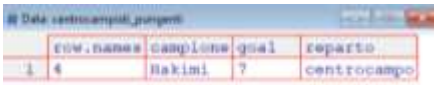


	campione	goal
1	Lukaku	24
2	Martinez	17
3	Sanchez	7
4	Hakimi	7

Notate che negli argomenti `subset=` e `select=` **potete omettere di far precedere il nome della variabile da quello del dataframe** cui appartengono, perché l'avete **già indicato nell'argomento `x= dataframe`**. Troveremo questa modalità in molti altri casi.

Selezioniamo solo i centrocampisti con più attitudine al goal; attenzione al doppio `==`, e alle virgolette in cui racchiudiamo la stringa di testo "`centrocampo`":

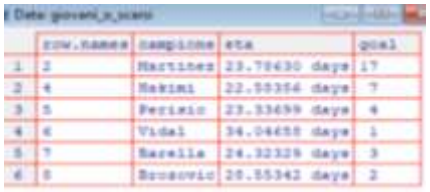
```
centrocampisti_pungenti<-subset(x= avanti, subset= reparto
   == "centrocampo" & goal >= 5, select=c(campione, goal,
   reparto)
```



	row.names	campione	goal	reparto
1	4	Hakimi	7	centrocampo

Selezioniamo i **più giovani** (meno di 25 anni) **o** quelli che comunque hanno **segnato poco** (4 goal o meno):

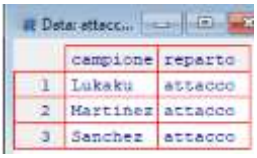
```
giovani_o_scarsi<-subset(x= avanti, subset= eta<25|goal<=4,
   select=c(campione, goal, eta))
```



	row.names	campione	eta	goal
1	2	Martinez	23.78630 days	17
2	4	Hakimi	22.99354 days	7
3	5	Perisic	23.53659 days	6
4	6	Vidal	34.04658 days	1
5	7	Barella	24.32329 days	3
6	8	Brozovic	28.55342 days	2

Infine, selezioniamo in maniera estrosa solo gli attaccanti:

```
attaccanti_in_negativo<-subset(x=avanti,    subset=reparto!="centrocampo",
select=c(campione, reparto))
```



	campione	reparto
1	Lukaku	attacco
2	Martinez	attacco
3	Sanchez	attacco

Usiamo `subset` per ripetere una verità universale nelle funzioni con più argomenti: **se omettete il nome dell'argomento** (`x=`, `subset=`, ecc.), R si aspetta che gli **argomenti si succedano nell'ordine previsto dalla funzione** (nel caso di `subset`, prima il dataframe, poi il criterio di selezione delle righe, infine il criterio di selezione delle colonne). Se gli argomenti non rispettano quest'ordine, R non trova quello che si aspetta e restituisce un errore o un output insensato. Invece, **se indicate il nome dell'argomento, potete inserirli nella funzione in qualsiasi ordine**. Per esempio:

`attaccanti_in_negativo<-subset(select= c(campione,reparto), x= avanti, subset=reparto!="centrocampo")`	Invertire l'ordine previsto degli argomenti denominandoli **funziona**.
`attaccanti_in_negativo<-subset(avanti, reparto!="centrocampo", c(campione, reparto))`	Rispettare l'ordine degli argomenti senza denominarli **funziona**.
`attaccanti_in_negativo<-subset(c(campione,reparto), avanti, reparto!="centrocampo")` `Error in subset(c(campione, reparto), avanti, reparto != "centrocampo") :  oggetto "campione" non trovato`	Invertire l'ordine previsto degli argomenti non denominandoli **non funziona**.

2.4 Importare dati in R

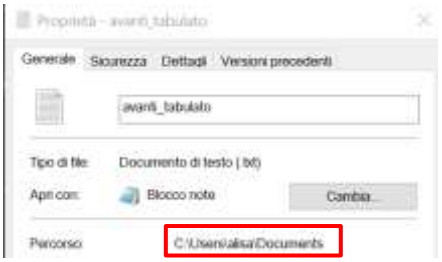
Nella nostra prossima realtà, lavoreremo molto più spesso con dati provenienti da un archivio esterno e importato in R come dataframe. Tipicamente, l'archivio consiste in un foglio di calcolo organizzato in wide format (una riga → un'unità di osservazione; una colonna → una misura), in cui i dati sono stati precedentemente inseriti a mano o esportati da un altro programma (ad esempio, i tempi di reazione in un esperimento al computer). R importa dati da file di molti formati diversi, ma, per evitare troppa eterogeneità, noi lavoreremo con esercizi ed esami importando dati da file in formato .txt o formato .csv.

Se non diversamente specificato, R si indirizza automaticamente, nella ricerca dei file da aprire (o da salvare) alla cartella in cui il programma è installato. Per caricare il dataframe "`avanti_tabulato.txt`" dovremmo digitare l'intero percorso in cui il file è stato salvato:

```
avanti_tabulato <- read.delim("C:/Users/lisa/Documents/avanti_tabulato.txt", header=TRUE)
```

- *il dataframe `avanti_tabulato`*
- *è creato*
- *leggendo il file in formato delimitato da tabulazioni*
- *che è così identificato nel computer*
- *e in cui la prima riga costituisce l'intestazione delle colonne*

Se avete dubbi sul nome del percorso, cliccate sull'icona del file e selezionate Proprietà: lì è immagazzinato il percorso. Attenti agli slash, però: R usa **/** per separare cartelle e sottocartelle, mentre il vostro sistema usa ****.



`read.delim` è una "sottofunzione" della più generica **`read.table`**, che istruisce R a leggere file in formato tabulare e crearne il corrispondente dataframe; uno degli argomenti di `read.table` è `sep=" "`, in cui dovremmo specificare `"\t"` per file delimitati da tabulazioni o `","` per file con valori separati da virgola. Più rapidamente, invece, **`read.delim`** delinea, senza bisogno di ulteriori specificazioni, il tipo di file che R deve aprire; analogamente, per aprire un file .CSV, useremo la funzione **`read.csv`**.

Un trucco per velocizzare l'operazione è ricordarsi di fissare la working directory in cui si trovano i dati che saranno analizzati in tutta la sessione di lavoro usando **`setwd`**`(percorso)`:

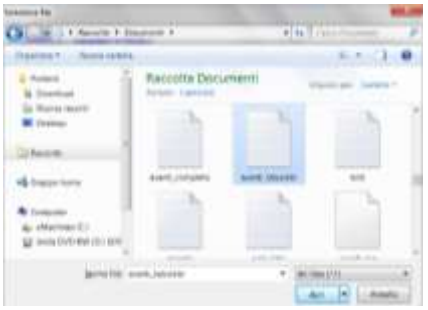
```
setwd("C:/Users/lisa/Desktop/TAD 1 e 2")
```

In questo modo, in qualsiasi momento della sessione di lavoro potremo importare un file specificandone solo nome e intestazione:

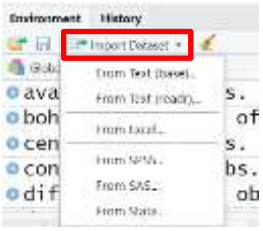
```
avanti_tabulato<-read.delim("avanti_tabulato.txt", header=TRUE)
```

Ancora più facile, però, è inserire l'argomento **`file.choose`**: analogo a `dir.choose` che abbiamo già visto per selezionare una directory: una volta dato Invio, apre la classica finestra di dialogo in cui si può scegliere il file nella cartella che lo contiene:

```
avanti_tabulato<-read.delim(file.choose(), header=TRUE)
```

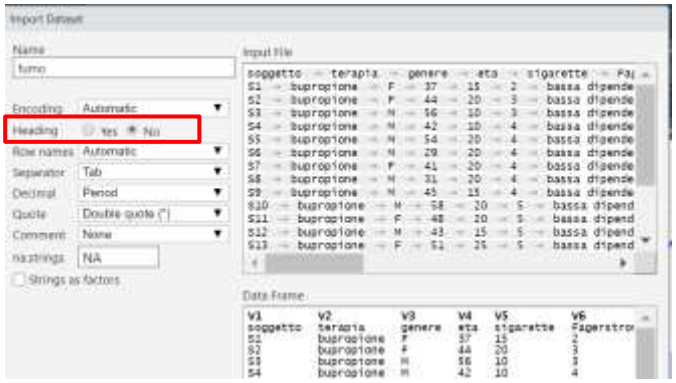


Se lavorate con **Rstudio**, per **importare dati** scegliete dalla finestra `Data: Import Dataset` → `From Text File` e poi usate le finestre per individuare la posizione del file da importare.

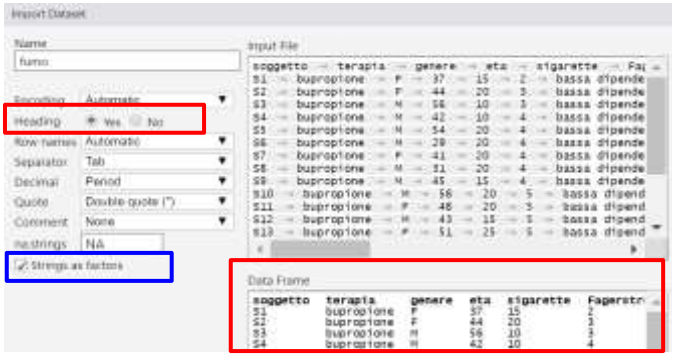


Attenzione: prima dell'apertura, un'utile finestra di **Preview** vi consente di individuare eventuali errori nell'importazione, prima che sia troppo tardi.

L'errore di importazione più comune – ma relativamente raro - è **l'intestazione delle colonne**, che R interpreta erroneamente come prima riga dei dati: la conseguenza è che tutte le variabili sono considerate di classe `character`, ed è assegnato un nome di default alle colonne (V1, V2, V3…).



Per evitarlo, assicuratevi sempre che sia spuntata l'opzione YES nella riga **`Heading`**. Per comodità, accertatevi che sia spuntato **`Strings as factors`**, perché nelle analisi che faremo le stringhe di testo sono perlopiù usate come `factor`. Dopo la conferma con Importa, il corrispondente comando viene stampato in Console, nella Finestra Data compare il nome del dataframe e si apre la finestra Visualizza.

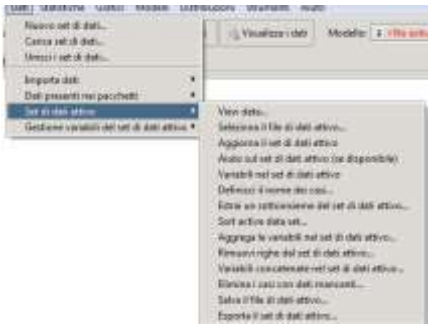


Infine, se volete usare `Rcommander`, usate il menu **`Dati → Nuovo set di dati`**:

Si assegna un nome al dataset, si compilano le righe e le colonne della struttura che si apre e, cliccando su OK, si salva l'oggetto.

Per modificare **dataframe esistenti**, nel menu **`Dati`** si sceglie **`Importa dati`**, specificandone il formato (.txt, .xls, ecc.) tra le opzioni presentate e poi cercandolo tra le cartelle. Se è un dataframe su cui si è già lavorato, si sceglie con l'icona **`Set di dati`**, decidendo poi di visualizzarlo (**`Visualizza i dati`**) o di modificarlo (**`Edita i dati`**):

Una volta caricato un dataframe, il menu **`Dati → Set di dati attivo`** offre una gran quantità di possibilità di azioni sulla sua struttura:



Per esempio, per estrarre solo alcuni casi abbiamo usato la funzione **subset**; con RCommander, per estrarre i giovani con meno di 30 anni sceglieremmo **Estrai un sottoinsieme del set di dati attivo**. Quindi, nella finestra di dialogo decidiamo se portare con noi tutte le variabili o solo alcune, qual è il criterio di selezione (avanti_tabulato$eta<30) e diamo un nome al subset.

Dopo aver cliccato su **OK**, il subset è creato e nello script è stampata la funzione utilizzata:

```
Script di R  R Markdown

giovani <- subset(giovani, subset=avanti_tabulato$eta<30, select=c(campione,goal))
```

Il menu **Dati → Gestione variabili del set di dati attivo** offre altrettante possibilità di agire sulle singole variabili che lo compongono: possiamo calcolarne, rimuoverle, standardizzarle, cambiare la loro classe, rinominarle, ecc.

Capitolo 3
La statistica e i modelli statistici univariati

"Le statistiche sostengono che più tempo si passa in automobile sulle strade e più aumenta la probabilità di incidenti; la prudenza consiglia quindi di possedere un'auto molto veloce e di correre a tavoletta"

[Lucie Olbrechts-Tyteca]

"Ci sono tre professori di Statistica che vanno a caccia di lepri. Ad un tratto ne vedono una. Il primo spara ... un metro a destra. Il secondo spara ... un metro a sinistra. Il terzo esclama: 'L'abbiamo presa!'"

"I numeri sono come le persone: torturali abbastanza ed essi ti diranno qualsiasi cosa"

Dovremmo aver intuito che in questo esame ci occuperemo di **Statistica**. Una delle possibili definizioni (Wilcoxon, nel 1935, ne aveva contate 115 diverse) la descrive come: "[...] la disciplina che elabora i principi e le metodologie che presiedono al processo di rilevazione e **raccolta dei dati**, alla **rappresentazione sintetica** e alla **interpretazione** dei dati stessi e, **laddove ve ne siano le condizioni**, alla **generalizzazione** delle evidenze osservate".

Per questo esame, i dati su cui lavoreremo **saranno già stati raccolti da altri**: useremo i dataframe a disposizione su Elly, composti da **unità statistiche** (caso individuale, **soggetto**), che compongono il **collettivo statistico** (**campione**) su cui sono stati rilevati **caratteri** (aspetto elementare oggetto di rilevazione; **variabile**), ciascuno dei quali presente in diverse **modalità**:

Unità statistiche
Si chiama unità statistica il caso individuale che compone un collettivo statistico (campione, insieme di riferimento…). Nella ricerca psico-sociale viene definito partecipante o, fino a qualche anno fa, soggetto.

Caratteri
Si chiama carattere ogni aspetto elementare oggetto di rilevazione nelle unità statistiche del campione. Spesso vi si riferisce con il termine variabile, in senso generale.

	sogg	genere	eta	stato_civile	istruzione	cresciuto_animali_domestici
1	S1	F	23	single	diploma superiore	si
2	S2	M	21	single	laurea	si
3	S3	F	68	coniugato	laurea	no
4	S4	M	18	single	diploma superiore	si
5	S5	F	23	single	diploma superiore	si
6	S6	M	68	coniugato	laurea	no
7	S7	F	55	coniugato	laurea	no

Ci concentreremo, quindi, sul resto del lavoro su dati empirici: la loro **rappresentazione sintetica** con grafici e numeri (cioè, la **costruzione di modelli**), la loro **interpretazione** e la **generalizzazione** dei dati dal campione su cui sono stati raccolti alla **popolazione** da cui il campione è stato estratto (**inferenza**).

Secondo la definizione, la Statistica non è matematica in senso stretto, ma usa la matematica, ovvero usa numeri e formule per descrivere la realtà. Il passaggio dalla realtà ai numeri segue **regole rigorose**: i numeri non devono stravolgere la realtà, ovvero dobbiamo usare il sistema numerico come se avesse le stesse caratteristiche del sistema empirico che deve rappresentare→ **omomorfismo**. Nel passaggio dal reale al matematico, possiamo usare i numeri per descrivere **qualità** o **quantità**: le regole di trasposizione dal sistema empirico al sistema numerico definiscono la **scala di misura** del dato. Conoscete già le scale di misura dal vostro percorso di studi precedente, ma, poiché possono essere fonte di catastrofici fraintendimenti, ripassiamole rapidamente.

3.1 La misurazione e le scale di misura

Misura ciò che è misurabile, e rendi misurabile ciò che non lo è.
Galileo Galilei (1564-1642)

L'importanza attribuita alla misurazione nella tradizione scientifica può essere riassunta da questa affermazione di Kelvin (1891, pag. 80): "Ho spesso sostenuto che quando puoi misurare quello di cui stai parlando, ed esprimerlo in numeri, allora puoi dire di conoscere qualcosa su di lui; ma quando non puoi misurarlo, quando non puoi esprimerlo in numeri, la tua conoscenza è scarsa e insoddisfacente: può forse essere l'inizio della conoscenza, ma sei consapevole di essere avanzato ben poco sul piano della scienza, qualunque possa essere la materia".
D'altronde, però, Yule (1921, pag. 106-107), ammonisce: "misurare non necessariamente significa progresso. Se fosse impossibile misurare quello che si desidera, l'avidità di misurare potrebbe, per esempio, semplicemente sfociare nel misurare qualcos'altro – e forse nel dimenticare la differenza -, o nell'ignorare qualcosa solo perché non può essere misurato".
Identificare *tout court* la scienza con la misurazione è un errore: la scienza riguarda osservazioni sistematiche e controllate, e il tentativo di verificare o falsificare queste osservazioni. Se le prescrizioni della scienza richiedessero che tutte le osservazioni debbano essere quantificabili, allora le scienze naturali e sociali sarebbero gravemente impedite. I dubbi sull'assoluta utilità della descrizione quantitativa, espressi un secolo fa, potrebbero essere ben ponderati da chi pratica oggi la psicologia sperimentale. Comunque, il dato di fatto è che, sia pure con rilevanti eccezioni, la psicologia ha fin dai suoi primi passi ricercato e accettato la necessità di quantificare i fenomeni oggetto del suo studio. Anche se la critica di Yule è sempre valida, la combinazione di disegno sperimentale e metodo statistico introdotta da Fisher ha reso spesso (ma non sempre) possibile la combinazione di controllo sperimentale e controllo statistico.

La **misurazione consiste nell'applicazione della matematica a eventi**. Usiamo numeri per designare oggetti ed eventi e la relazione tra loro. Talvolta, gli oggetti della misura sono reali e le relazioni immediatamente comprensibili: per esempio, i tavoli da pranzo e le loro dimensioni, il loro peso, la loro superficie, ecc. In altre occasioni, si può avere a che fare con fenomeni non tangibili, come l'intelligenza, la personalità, l'autostima: in questi casi, le misure sono descrizioni del compito, che, secondo la nostra opinione, riflettono il costrutto sottostante (**operazionalizzazioni**). Il punto fondamentale, in entrambi i casi, è che **la misura ci possa fornire precise ed economiche descrizioni degli eventi, in un modo che sia facilmente comunicabile** agli altri. Qualunque sia l'opinione che si possa nutrire per la matematica, per la sua complessità e difficoltà, è generalmente considerata una disciplina chiara, ordinata e razionale: il ricercatore tenta di aggiungere chiarezza, ordine e razionalità al mondo usando la misurazione.
Peraltro, tutte le misurazioni implicano difficoltà pratiche, che si riuniscono a formare la **sorgente dell'errore di misura**. Dal punto di vista statistico, l'errore di misura **aumenta la variabilità nei dati**, diminuendo la precisione delle analisi di sintesi e delle inferenze. Ne deriva che gli scienziati puntano all'accuratezza, affinando continuamente tecniche e strumenti di misurazione, pur nella consapevolezza che ottenere una misura assolutamente precisa (***reliable***) è impossibile. Anche se Quetelet e Galton possono essere criticati per aver diffuso la nozione di distribuzione degli errori come legge di natura (Capitolo 6), il loro lavoro ha riconosciuto l'irriducibile varietà della materia vivente: questo esprime l'essenza dell'approccio statistico, ovvero che **non può esistere un'assoluta accuratezza nella misurazione**, ma solo un **giudizio di accuratezza**, nei termini dell'intrinseca **variazione entro e tra gli individui**. Le statistiche sono gli strumenti per **valutare le proprietà di queste fluttuazioni casuali**.

Il problema di porre in relazione la scala di misura con la natura dei dati raccolti è stato posto per la prima volta da Stanley Smith Stevens (1946): "*Possiamo dire che la misurazione, nel senso più ampio, consiste nell'attribuzione di numeri a oggetti o eventi seguendo determinate regole. Il fatto che si possano* ***assegnare dei numeri seguendo regole differenti*** *porta a* ***differenti tipi di scala*** ***e livelli di misurazione***".

Secondo Stevens, non si possono effettuare operazioni *matematiche* su scale nominali e ordinali, il che distinguerebbe una statistica **parametrica** (i modelli applicabili a dati misurati su scala a intervalli o rapporti equivalenti) da una statistica **non parametrica** (i modelli applicabili a dati su scala nominale e ordinale, che non usano tutte le proprietà dei numeri che li rappresentano). Non tutti, però, sono d'accordo su questa impostazione piuttosto rigida, sia da un punto di vista teorico (***"since the numbers don't remember where they came from, they always behave just the same way, regardless"***)[10], sia da un punto di un punto di vista pragmatico.

Ogni scala o livello di misura mantiene le proprietà del livello precedente, aggiungendovi caratteristiche peculiari. Quando, per rappresentare un sistema empirico, è utilizzata solo la proprietà di **simbolo** del sistema numerico, con lo scopo di "categorizzare" gli oggetti o i soggetti che interessano, il dato è misurato su **scala nominale**: questa scala consente soltanto di classificare le unità del collettivo statistico in tanti gruppi distinti quante sono le modalità del carattere**.** Il simbolo – numero può essere sostituito da qualsiasi altro simbolo grafico senza perdere informazioni sul dato empirico, purché la sostituzione rispetti le proprietà della scala nominale: equivalenza entro le categorie (tutti i membri di ogni categoria sono rappresentati dallo stesso numero), non equivalenza tra categorie (tutti i membri di diverse categorie sono rappresentati da numeri diversi). In questa scala vige il principio di equivalenza, simmetria (equivalenza simmetrica: $A = B$, $B = A$, e, naturalmente, **non** equivalenza simmetrica: $A \neq B$, $B \neq A$) e transitiva (se $A = B$ e $B = C$, allora $A = C$).

L'unico attributo considerato in questo insieme di eventi è la ***qualità*** *"tipo di fiore": quindi, l'unico livello di misurazione consentito è* ***qualitativo****, in cui le regole di corrispondenza per assegnare un numero ad ogni evento sono qualitative: ad ogni evento diverso deve essere assegnato un numero diverso, e ad ogni evento uguale deve essere assegnato un numero uguale*

Nel livello di misura nominale, i tre mazzi di fiori sono tutti validi esempi della regola di corrispondenza, ***poiché rispettano l'omomorfismo tra evento e numero assegnato.***

La **Scala ordinale** è la prima scala "quantitativa", in cui viene introdotto il concetto di ordine tra le ripartizioni che vengono effettuate. Questa rappresentazione numerica consente, come la scala nominale, di classificare le unità statistiche in gruppi omogenei, e in più, permette di "graduare" i gruppi in base all'ordine che le modalità presentano. Se i numeri sono assegnati secondo un ordine crescente, i soggetti cui è assegnato il numero 1 presentano una quantità inferiore della caratteristica oggetto di misura rispetto ai soggetti ai quali è assegnato il numero 2 e così via. Tutti i soggetti con lo stesso numero presentano la medesima (o equivalente) quantità della caratteristica in esame. Oltre al principio di equivalenza della scala nominale, in questa scala troviamo il **principio d'ordine**, che produce relazioni d'ordine simmetriche (se $A < B \rightarrow B > A$) e transitive (se $A < B \; e \; B < C \rightarrow A < C$)

[10]Lord, *On the statical treatment of the football numbers*, 1953. L'apologo è breve, spiritoso, illuminante e disponibile su Elly: dedicategli dieci minuti....

In questo stesso insieme di eventi, ora l'attributo considerato è il ***grado*** *"quanto ti piace…": aggiungiamo una regola di corrispondenza in più, ferma restando la precedente: ad ogni evento diverso deve essere assegnato un numero diverso e ad ogni evento uguale deve essere assegnato un numero uguale;* ***la relazione di grandezza tra i numeri assegnati deve corrispondere alla relazione di preferenza*** *tra gli eventi.*

Nel livello di misura ***ordinale****, solo due esempi sono validi, poiché rispettano l'omomorfismo tra evento e numero assegnato.*

La **scala a intervalli equivalenti** utilizza una **unità di misura** costante, dotata, però, di uno 0 arbitrario (l'esempio tipico è la scala di temperatura Celsius, che può assumere anche valori negativi); quando lo 0 non è arbitrario, ma assoluto (pertanto non sono possibili valori negativi), la scala si definisce a **rapporti equivalenti**. Oltre alle proprietà di equivalenza e ordine, nella scala a intervalli equivalenti troviamo la **costanza del rapporto tra intervalli**, per cui le differenze tra i valori sono equivalenti (tra 5 e 10 c'è la stessa differenza che tra 15 e 20), mentre nella scala a rapporti vige **anche** la proprietà della **costanza del rapporto tra i valori** (20 è il doppio di 10, come 6 è il doppio di 3).

Consideriamo ora l'attributo ***grandezza della corolla****: usando la scala ordinale, la differenza tra il fiore blu e il fiore bianco è = |1|, e NON è equivalente alla differenza |2| tra il fiore rosa ed il fiore bianco; se questi fossero veri numeri, significherebbe che il primo ed il secondo fiore (blu e bianco) sarebbero tra loro più simili per diametro di quanto lo siano il secondo ed il quarto fiore (bianco e rosa) –* ***cosa empiricamente non vera****. Conosciamo quindi solo un "pezzo" di informazione rispetto al fenomeno.*

Usando ***un'unità di misura****, aggiungiamo un'altra regola:* ***l'equivalenza degli intervalli****. In questo secondo esempio, in cui l'attributo "grandezza" è misurato su* ***scala a rapporti equivalenti****, vediamo, usando un* ***centimetro****, che la differenza tra il fiore blu e il fiore bianco è =|2|, ed è* ***equivalente*** *alla differenza |2| tra il fiore rosa e il fiore bianco.*

All'interno dei dati misurati su scala a intervalli o a rapporti, si possono distinguere variabili **discrete** (dati due valori della variabile che esaminiamo, non è sempre possibile trovare un valore **teorico** compreso tra essi) e variabili **continue**: dati due valori della variabile che esaminiamo, è sempre possibile trovare un valore **teorico** tra essi, anche se, pragmaticamente, non sempre è possibile trovare strumenti di misura così sensibili da rilevare valori intermedi molto piccoli. Questa differenza è comunque solo teorica, perché la misurazione di un carattere continuo comporta necessariamente un'approssimazione dovuta al troncamento dei numeri, ovvero al livello di precisione. In realtà, quindi, la misurazione di un carattere continuo avviene come se fosse discreto.

Un diverso modo di categorizzare i livelli di misura è la distinzione tra misure continue e misure discrete. Una misura è continua quando vi sono infinite possibili misurazioni tra una qualsiasi misurazione e un'altra (ad esempio, il peso, l'altezza, l'età…). Il dato empirico che stiamo misurando deve essere sempre considerato solo un'approssimazione al valore reale, che dipende dalla sensibilità dello strumento di misura. Invece, una misura è discreta quando un qualsiasi valore della distribuzione è del tutto separato da qualsiasi altro livello: tra essi non è possibile individuare valori intermedi, indipendentemente dalla sensibilità dello strumento di misura. Le variabili misurate a livello nominale (**qualitative)** sono sempre discrete, mentre quelle **quantitative** sono discrete solo se non è possibile individuare una misura intermedia tra altre due (per esempio le frequenze, che affronteremo tra poco).

3.2 I modelli

"The hallmark of good science is that it uses models and "theory", but never believes them"
Wilks, cit. in Tukey, 1961

Genericamente parlando, un **modello** è una **riproduzione,** di un qualsivoglia fenomeno, che consente di replicare il fenomeno stesso in **scala**, risparmiando tempo ed energia, e di valutare ipotesi e **previsioni** sul fenomeno in maniera **realistica** e **affidabile**. Noi useremo modelli, fatti di numeri e grafici, per rappresentare sinteticamente dati che dovrebbero operazionalizzare costrutti e relazioni tra dimensioni psicologiche.
Innumerevoli altre discipline usano modelli: per esempio, un architetto potrebbe costruire diversi modelli per riprodurre in scala il London Bridge:

I quattro modelli di ponti sono più o meno **affidabili** nella loro capacità di riflettere le caratteristiche del ponte reale: se volessimo **stimare** la capacità di resistenza al vento del London Bridge usando simulazioni con i ponti C e D, e apportassimo modifiche ai tiranti del ponte usando tali stime, provocheremmo una strage di massa al primo temporale. I modelli C e D, infatti, **non si adattano al fenomeno reale**, ovvero (come diremo da ora in poi) **non hanno un buon fit**: questo vuol dire che le **previsioni** e le **generalizzazioni** sul fenomeno – London Bridge fatte sulla base di questi due modelli **non sono affidabili**.
Noi – purtroppo – non useremo modellini con i Lego per riprodurre in scala la realtà, ma **modelli statistici**, e comunque secondo la logica appena esposta: se i modelli statistici si adattano al fenomeno reale, ovvero se hanno un buon fit, allora le nostre previsioni sul fenomeno basate su tali modelli saranno affidabili; se i modelli statistici non si adattano al fenomeno reale, ovvero se non hanno un buon fit, allora le nostre previsioni sul fenomeno basate su tali modelli non saranno affidabili.

La **goodness of fit** di un modello è quindi la sua capacità di riprodurre il più semplicemente e fedelmente possibile un dato reale, di solito complesso. È una capacità che si può **quantificare**.

Scendendo su un piano più specifico, un modello statistico è una **funzione delle variabili esplicative X** (variabili indipendenti, predittori) il cui fine è quello di **spiegare il meglio possibile la variazione nella variabile dipendente Y** (risposta): l'obiettivo sarà quello di **individuare i valori dei parametri della funzione** – modello che portano alla **migliore goodness of fit** del modello ai dati.

Il **miglior modello** è quello che lascia la **minima quantità di variabilità di Y non spiegata dalle variabili esplicative**.

Vedremo due diversi metodi per stimare i parametri del miglior modello: il **metodo dei minimi quadrati** (nel modello lineare generale) e il **metodo della massima verosimiglianza**, così come vedremo molteplici modi di quantificare la goodness of fit, diversi a seconda del tipo di modello costruito. Riprenderemo altre caratteristiche della costruzione di modelli (**modeling),** come la **parsimonia e l'adeguatezza** quando parleremo della costruzione di modelli con più X (regressione multipla, analisi della varianza fattoriale).

Iniziamo per ora con i modelli statistici più semplici, che rappresentano sinteticamente **caratteristiche di una sola distribuzione di valori** (variabile) per volta; li differenzieremo per scala di misura del dato che vogliono rappresentare e per scopo (il tipo di caratteristica che intendono esprimere). Vedremo le **distribuzioni di frequenza**, gli **indici di tendenza centrale**, che permettono di rappresentare l'ordine di grandezza di un fenomeno, gli **indici di dispersione**, che forniscono una misura della variabilità del fenomeno, gli **indici di posizione**, che identificano la posizione di un dato all'interno di una distribuzione ordinata, e gli **indici di forma** dell'intera distribuzione. Vedremo gli indici più adeguati a seconda del tipo di scala di misura adottata.
Poi, procederemo con l'ambizioso tentativo di generalizzare alla popolazione le caratteristiche che i modelli hanno più o meno efficacemente rappresentato nel campione, e passeremo dalla descrizione del dato alla stima della sua probabilità. Successivamente, passeremo a modelli più interessanti, che rappresentano sinteticamente caratteristiche di **più** distribuzioni di valori congiuntamente considerate.
In **tutti** questi passaggi useremo R e le possibilità che ci offre.

*Iniziamo usando il dataframe **gatti**; scaricatelo da Elly, insieme alla descrizione delle variabili che lo compongono: leggete la descrizione e aprite il dataframe in R, prima di procedere oltre.*

3.2.1 Modelli per distribuzioni univariate nominali

	sogg	genere	eta
1	S1	F	23
2	S2	M	21
3	S3	F	68
4	S4	M	18
5	S5	F	23

Si chiama **distribuzione statistica disaggregata** secondo il carattere X l'insieme delle osservazioni (rappresentate da numeri o espressioni verbali) relative alle N unità del campione.
In simboli, la distribuzione statistica disaggregata sarà rappresentata come: $x_1, x_2, \dots x_N$.

Per le variabili a livello nominale, i descrittori utilizzabili riguardano la **distribuzione di frequenza**, cioè lo schema con cui si associa a ciascuna modalità della variabile la **frequenza** pertinente, ossia il numero di volte in cui si verifica un determinato "evento" in un gruppo di altri eventi. Queste sono dette anche **frequenze assolute** e sono indicate con N_1, dove $i = 1, 2, \dots, k$ (k corrisponde al numero delle modalità del carattere).

Si possono definire anche le **frequenze relative** o **proporzioni**, date dal rapporto tra le frequenze assolute e il totale delle unità del campione.

$$f_i = \frac{n_i}{N}$$

Le proporzioni consentono di interpretare rapidamente l'importanza, della singola modalità nell'ambito della distribuzione; possono essere trasformate in **percentuali**, moltiplicandole per 100.
Infine, possiamo calcolare le **frequenze cumulate assolute**, ovvero la somma di tutte le frequenze assolute che si susseguono dalla prima all'ultima modalità; dividendole per il totale delle osservazioni si avranno le frequenze cumulate **relative**, e, moltiplicando queste ultime per 100, le frequenze cumulate **percentuali**.
Le distribuzioni di frequenza sono sintetizzate da R con la funzione `table(variabile)`, che mostra le **frequenze assolute** per ciascuna delle modalità della variabile in una **tabella di contingenza**; `table` è utile con variabili **discrete**.
Per esempio, vogliamo vedere quanti maschi e quante femmine hanno partecipato alla ricerca sui gatti: usiamo quindi la variabile `$genere`[11] per chiedere:

```
table(gatti$genere)
F  M
48 31
```

C'è una prevalenza del genere femminile; vediamo quanti vivono con un gatto:

```
table(gatti$vive_con_gatto)
no si
38 41
```

I campione è ben bilanciato per questa variabile. Infine, lo stato civile:

```
table(gatti$stato_civile)
 coniugato divorziato     single
        16          2         61
```

Questa distribuzione dà un po' fastidio: due soli divorziati su 79 costituiscono una classe troppo piccola per essere utilizzabile a fini di confronto: uniamo i divorziati alla categoria "single", **dicotomizzando** la variabile. `$stato_civile` è una variabile di classe `factor` (verificatelo...): i livelli di un `factor` possono essere riferiti a categorie le cui etichette manifestano solo qualità diverse, senza esprimere un qualche tipo di ordinamento tra livelli, oppure indicare un **ordine** tra i livelli (ne vedremo un esempio tra poco, quando categorizzeremo la variabile `$eta`): in quest'ultimo caso avremo

[11] Per amor di chiarezza, quando negli esempi useremo una variabile di dataframe si adotterà sempre la modalità dataframe`$variabile` per indicarla nelle funzioni. È anche possibile usare la funzione `attach(dataframe)` all'inizio di ogni sessione di lavoro e usare solo il nome delle variabili, senza anteporre il dataframe, nelle funzioni successive – anche se questo rende più fastidioso il completamento automatico e i suggerimenti: scegliete liberamente la modalità che vi è più comoda.

un **ordered factor**. Tra i livelli del fattore `$stato_civile` non è possibile riconoscere un qualche tipo di ordine, quindi la ri-categorizzazione delle categorie ci pone un problema in meno di quello che affronteremo con i fattori ordinati: basta ri-assegnare l'etichetta della categoria "single" alla categoria "divorziato":

```
gatti$stato_civile[gatti$stato_civile=="divorziato"]<-"single"
table(gatti$stato_civile)
 coniugato divorziato     single
        16          0         63
```

Adesso possiamo **eliminare il livello "divorziato"**, che **ha frequenza = 0**. Per creare la nuova variabile a due livelli `$stato_civile2` usiamo **droplevels(factor)**, che cancella quei livelli di un fattore che registrano N=0:

```
gatti$stato_civile2<-droplevels(gatti$stato_civile)
table(gatti$stato_civile2)
coniugato    single
       16        63
```

Possiamo usare **addmargins(table)** per visualizzare il **marginale**, ovvero il totale; nelle tabelle di contingenza per una sola distribuzione (tabelle di contingenza **a una via**) avremo un solo marginale, che corrisponde al totale delle osservazioni. Nelle tabelle di contingenza per due distribuzioni (**tabelle di contingenza a due vie**, §7.1) avremo tre marginali: il totale delle osservazioni rappresentate nelle righe (prima distribuzione: **marginale di riga**), il totale delle osservazioni rappresentate nelle colonne (seconda distribuzione; **marginale di colonna**) e il **totale** delle osservazioni. Per ora:

```
addmargins(table(gatti$genere))
  F   M Sum
 48  31  79
```

La **modalità che compare con la maggiore frequenza** nella variabile è la **moda**, cioè **l'indice di tendenza centrale per le variabili a livello nominale**. Nei nostri esempi, la moda per il genere è "femmina", per l'istruzione è la laurea, per la convivenza con un gatto è la risposta "sì". Tutte queste variabili presentano una sola modalità con frequenza maggiore delle altre: sono quindi distribuzioni **unimodali**; possono naturalmente verificarsi variabili in cui più modalità hanno una stessa frequenza più alta, determinando così distribuzioni bimodali, trimodali, ecc.

Quando nella distribuzione sono presenti valori mancanti (che, ricordiamo, R codifica con `NA`), in `table` specifichiamo l'argomento **exclude=FALSE** (§2.2.6) che istruisce R a **non omettere** nella tabella in output i dati mancanti; se questo argomento viene omesso, R non presenterà valori NA nell'output. Vediamo per un esempio il dataframe **attaccamento**, che useremo tra un po', in cui sono riportati i dati di una ricerca su caregiver di persone anziane con demenza; venti dei pazienti sono ricoverati in una Residenza Sanitaria Assistenziale, altri venti sono a casa con il caregiver: solo per questi ultimi, si è chiesto al caregiver di quale aiuto potessero usufruire. La variabile presenta quindi dati mancanti, corrispondenti alle risposte dei venti caregiver il cui assistito è ricoverato in struttura. Vediamo come si presentano le risposte:

```
table(attaccamento$con_aiuto_di)
assistenza domiciliare   badante     nessuno
                     1        10         9
table(attaccamento$con_aiuto_di, exclude=NULL)
assistenza domiciliare   badante     nessuno        <NA>
                     1        10         9          20
```

Per calcolare le **frequenze relative**, ovvero le **proporzioni**, si usa **prop.table(table(variabile))**: l'oggetto della funzione `prop.table` è quindi un oggetto `table`:

```
prop.table(table(gatti$cresciuto_animali_domestici))
       no        si
0.4177215 0.5822785
```

Lo 0.58 del campione vive con un gatto, lo 0.42 no. Possiamo abbinare a `prop.table` la funzione `round(oggetto, decimali)`, che già conosciamo, per eliminare un po' degli inutili decimali della tabella:

```
round(prop.table(table(gatti$cresciuto_animali_domestici)),2)
  no   si
0.42 0.58
```

Fate attenzione ai valori **mancanti** quando calcolate le proporzioni: se in `table` **non** indicate `exclude=NULL`, il totale su cui saranno calcolate sarà riferito ai **soli dati non mancanti**; se invece specificate `exclude=NULL`, le proporzioni saranno calcolate sul totale di **tutte le osservazioni**, dati mancanti compresi. Usando l'esempio dei caregiver, se calcoliamo le proporzioni sui soli venti caregiver con pazienti in casa avremo:

```
round(prop.table(table(attaccamento$con_aiuto_di)),2)
assistenza domiciliare                badante                nessuno
                  0.05                   0.50                   0.45
```

Le proporzioni sono calcolate come rapporto tra frequenza di ogni modalità e totale dei dati **non mancanti** (20).

Se invece scriviamo:

```
round(prop.table(table(attaccamento$con_aiuto_di, exclude=FALSE)),2)
assistenza domiciliare                badante                nessuno                  <NA>
                  0.02                   0.25                   0.22                  0.50
```

, le proporzioni sono calcolate come rapporto tra frequenza di ogni modalità e totale di **tutte** le osservazioni (40).

Se vi trovate più a vostro agio con le **percentuali** che con le proporzioni, basta **moltiplicare queste ultime** per 100:

```
round(prop.table(table(gatti$cresciuto_animali_domestici)),3)*100
  no   si
41.8 58.2
```

La funzione **`Freq`** del package `DescTools` fornisce in un unico output frequenze assolute, percentuali e cumulate – ma non è flessibile come `table`, che troveremo in molte altre situazioni. Comunque, nessuno vieta di usarla per descrivere rapidamente una semplice distribuzione univariata nominale:

```
Freq(gatti$cresciuto_animali_domestici)
   level  freq   perc  cumfreq  cumperc
1     no    33  41.8%       33    41.8%
2     si    46  58.2%       79   100.0%
```

Se ci sono dati mancanti, l'argomento per gestirli è `useNA=`: la sua opzione di default è `"no"`, che si può cambiare impostando `useNA="always"`.

```
Freq(attaccamento$con_aiuto_di)
                    level  freq   perc  cumfreq  cumperc
1  assistenza domiciliare     1   5.0%        1     5.0%
2                badante    10  50.0%       11    55.0%
3                nessuno     9  45.0%       20   100.0%
Freq(attaccamento$con_aiuto_di, useNA = "always")
                    level  freq   perc  cumfreq  cumperc
1  assistenza domiciliare     1   2.5%        1     2.5%
2                badante    10  25.0%       11    27.5%
3                nessuno     9  22.5%       20    50.0%
4                   <NA>    20  50.0%       40   100.0%
```

Quando la distribuzione **non è di tipo numerico**, la presentazione delle categorie segue **l'ordine dei livelli**: si può cambiare usando l'argomento `ord=`, per esempio mostrando le categorie in ordine crescente (`ord="asc"`) o decrescente (`ord= "desc"`):

```
Freq(gatti$istruzione, ord="desc")
                            level  freq   perc  cumfreq  cumperc
1                          laurea    37  46.8%       37    46.8%
2              diploma superiore    35  44.3%       72    91.1%
3  specializzazione post lauream     7   8.9%       79   100.0%
```

Quando le osservazioni riguardano caratteri continui, oppure caratteri discreti che presentano un numero elevato di modalità, si ricorre al **raggruppamento dei dati in classi**, anche se questa operazione comporta una perdita di informazioni. Le classi possono essere di uguale o diversa **ampiezza**: l'ampiezza è data dalla differenza tra il **limite inferiore** e il **limite superiore** della classe (in effetti, quando la distribuzione è continua ogni modalità può essere considerata una classe di ampiezza = 1). Quando uno dei limiti non è specificato (ad esempio, "da 70 anni in su"; "dal punteggio più piccolo fino a 20"), l'intervallo della classe è definito **aperto** (superiormente o inferiormente).
Vediamo, per esempio, come si presenta la distribuzione di frequenza di `gatti$eta`:

```
table(gatti$eta)
18 19 20 21 22 23 24 25 26 27 28 29 30 31 33 37 40 41 46 49 51 52 53 54 55 56 58 61 65 66 67 68
 1  2  1  1  4 13 10  6  2  3  1  2  8  1  2  1  1  1  2  1  1  1  1  1  2  1  2  1  1  2  1  2
```

La moda è 23 (anni, $f = 13$), e l'ampiezza delle classi è $a = 1$: ogni classe corrisponde a una modalità della variabile.
Vediamo le frequenze cumulate assolute di questa distribuzione continua, usando **`cumsum(table)`**:

```
cumsum(table(gatti$eta))
18 19 20 21 22 23 24 25 26 27 28 29 30 31 33 37 40 41 46 49 51 52 53 54 55 56 58 61 65 66 67 68
 1  3  4  5  9 22 32 38 40 43 44 46 54 55 57 58 59 60 62 63 64 65 66 67 69 70 72 73 74 76 77 79
```

Da questa variabile continua creiamo una variabile discreta, dividendola in quattro classi di diversa ampiezza: ragazzi da 18 a 25 anni (c_1), giovani adulti da 26 a 45 anni (c_1), maturi da 46 a 60 anni (c_3) e anziani da 61 a 68 anni (c_4). La variabile che creeremo sarà quindi di classe **`factor`**, dato che le classi raggruppano i soggetti in base alla loro età e le variabili `factor` sono variabili di raggruppamento. Non solo: tra le classi è possibile riconoscere un ordine crescente, per cui la variabile finale dovrà essere un **`ordered factor`**.
Ci sono **più modi per raggiungere lo scopo**: vediamo prima il più arzigogolato per imparare una nuova funzione, e poi la modalità rapida, che useremo in diverse altre occasioni.
Abbiamo già visto `rep(cosa da replicare, numero di repliche)`: usiamola per creare la variabile `factor` **`eta_cat`**. Secondo la tabella di contingenza, abbiamo 38 ragazzi fino a 25 anni, 22 giovani da 26 a 45, 12 maturi da 46 a 60 e 7 anziani con oltre 61 anni: scriviamo allora:

```
eta_cat<-c(rep(1,38),rep(2,22),rep(3,12), rep(4,7))
eta_cat
[1]  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4
[77] 4 4 4
```

Ora dovremmo trasformare la distribuzione `eta_cat` in fattore, aggiungendola al dataframe `gatti`, ma abbiamo un problema da risolvere, prima: il dataframe **non è ordinato in base all'età**, ma in base al codice identificativo assegnato ai soggetti (variabile `$sogg`). Questo comporta che l'ordine delle età nelle righe del dataframe **non è lo stesso ordine crescente dei livelli** appena creati: dovremo quindi prima **ordinare il dataframe secondo l'età**, da 18 in su.

	sogg	genere	eta	*Cat.*
1	S1	F	23	← **1**
2	S2	M	21	← **1**
3	S3	F	68	← **4**
4	S4	M	18	← **1**
5	S5	F	23	← **1**
6	S6	M	68	← **4**
7	S7	F	55	← **3**
8	S8	F	53	← **3**

Ecco la nuova funzione **`order(variabile)`**: prudenzialmente, creiamo un nuovo dataframe, **`gatti2`**, che contiene tutte le righe e tutte le colonne del precedente, ma ordinate secondo i livelli dell'età in **senso crescente**. L'ordinamento è gestito dall'argomento opzionale `decreasing= TRUE /FALSE`, che di default è `FALSE`: se non si specifica nulla, si ottiene un ordinamento crescente.

```
gatti2 <- gatti [order(gatti$eta),  ]
```

il dataframe gatti2 è creato *dal dataframe gatti* *ordinato in base a* *i valori delle righe di $eta, in senso crescente,* *conservando tutte le colonne*

Possiamo verificare (la prima colonna di `gatti2` contiene i numeri di riga corrispondenti ai soggetti di `gatti`):

Dataframe gatti				Dataframe gatti2			
head(gatti[3],5)		tail(gatti[3],5)		head(gatti2[3],5)		tail(gatti2[3],5)	
	eta		eta		eta		eta
1	23	75	27	4	18	33	66
2	21	76	25	12	19	34	66
3	68	77	24	38	19	32	67
4	18	78	65	18	20	3	68
5	23	79	25	2	21	6	68

Adesso che l'ordine delle righe segue l'ordine crescente per età, possiamo trasformare la distribuzione `eta_cat` in variabile di classe `factor`: **`gatti2$eta_cat`**.

```
gatti2$eta_cat <- factor(eta_cat, levels = c(1:4), labels = c("ragazzi", "giovani", "maturi",
  "anziani"))
class(gatti2$eta_cat)
[1] "factor"
table(gatti2$eta_cat)
ragazzi giovani  maturi anziani
38       22      12       7
```

In questa distribuzione in classi, la moda – o meglio la **classe modale** – è "ragazzi".

Vediamo adesso due modi decisamente più funzionali per creare una variabile `factor` , che chiameremo `$eta_cat2`.

Una prima soluzione per raggruppare dati continui in classi è usare le `[]`, assegnando a tutti i soggetti con età fino a 25 il livello "ragazzi", a quelli tra 26 e 45 il livello "giovani", tra 46 e 60 il livello "maturi", sopra i 60 il livello "anziani":

```
gatti$eta_cat2[gatti$eta<=25]<-"ragazzi"
gatti$eta_cat2[gatti$eta>25 & gatti$eta<=45]<-"giovani"
gatti$eta_cat2[gatti$eta>45 & gatti$eta<=60]<-"maturi"
gatti$eta_cat2[gatti$eta>60]<-"anziani"

table(gatti$eta_cat2)
anziani giovani  maturi ragazzi
7       22      12      38
class(gatti$eta_cat2)
[1] "character"
```

C'è un dettaglio da sistemare: la variabile creata è di classe `character`, mentre noi desideriamo un `ordered factor` i cui i livelli **seguano un ordine corrispondente all'aumentare dell'età**, da giovani ad anziani. Ai fini delle analisi che usano variabili `factor`, la differenza fra `factor` e `ordered factor` è limitata quasi esclusivamente ai contrasti da utilizzare nell'analisi della varianza a misure ripetute[12], ma già che ci siamo impariamo come si fa. Trasformiamo la variabile da `character` a `factor` con **`as.factor(variabile)`**:

```
gatti$eta_cat2<-as.factor(gatti$eta_cat2)
class(gatti$eta_cat2)
[1] "factor"
```

Vediamone i livelli con la funzione **`levels(factor)`**:

```
levels(gatti$eta_cat2)
[1] "anziani" "giovani" "maturi"  "ragazzi"
```

[12] Capitolo 12

R, in assenza di ulteriori istruzioni, ha **assegnato l'ordine dei livelli usando l'ordine alfabetico** delle loro etichette, che non corrisponde all'ordine delle età che rappresentano. Ordiniamoli con `ordered(factor, livelli=)`, in cui scriveremo l'ordine corretto dei livelli:

```
gatti$eta_cat2 <- ordered(gatti$eta_cat, levels=c("ragazzi", "giovani", "maturi", "anziani"))
```

Verifichiamo:

```
class(gatti$eta_cat2)
[1] "ordered" "factor"

table(gatti$eta_cat2)
ragazzi giovani  maturi anziani
     38      22      12       7
```

Finito.

Vediamo l'ultima funzione, che useremo spesso per ri-categorizzare variabili: `ifelse(test= se trovi questa condizione, yes= fai questo, no= altrimenti fai quest'altro)`. Il primo argomento `test=` definisce la condizione da soddisfare ("se l'età è compresa tra i 18 e i 25 anni), il secondo `yes=` definisce l'azione da eseguire se la condizione è soddisfatta ("etichetta i soggetti come ragazzi"), il terzo `no=` l'azione da eseguire se la condizione di `test=` non è soddisfatta.

Il modo più semplice di usare `ifelse` è per creare variabili dicotomiche (solo due livelli), ma la funzione si può **nidificare** per creare variabili politomiche. Cominciamo con l'esempio facile, dividendo i soggetti in giovani (fino a 25 anni) e adulti (sopra i 25 anni):

```
gatti$eta_due<-ifelse(test= gatti$eta<=25, yes= "ragazzi", no= "adulti")
table(gatti$eta_due)
 adulti ragazzi
     41      38
```

Ora creiamo le tre categorie "ragazzi", "giovani" e "adulti": se la condizione per l'etichetta ragazzi ("inferiore a 25 anni") non è soddisfatta, R procede a verificare la **seconda condizione** ("età compresa tra 25 e 45 anni): se questa è soddisfatta, assegna "giovani", altrimenti "maturi":

```
gatti$eta_tre<-ifelse(test= gatti$eta<=25, yes= "ragazzi", no= ifelse(test= gatti$eta>25 &
 gatti$eta<=45, yes= "giovani", no="maturi"))
table(gatti$eta_tre)
giovani  maturi ragazzi
     22      19      38
```

Finiamo con le quattro categorie, da "ragazzi" ad "anziani": se la seconda condizione non è soddisfatta, R procede a verificarne una terza ("età compresa tra 46 e 60 anni"): in caso positivo, assegna al caso "maturo", altrimenti "anziano")

```
gatti$eta_quattro<-ifelse(test= gatti$eta<= 25, yes= "ragazzi", no= ifelse(test= gatti$eta> 25 &
 gatti$eta<= 45, yes= "giovani", no= ifelse(test= gatti$eta >45 & gatti$eta <=60, yes= "maturi",
 no= "anziani")))
table(gatti$eta_quattro)
anziani giovani  maturi ragazzi
      7      22      12      38
```

Questa suddivisione in classi di età può essere evolutivamente sensata, ma ha creato classi di ampiezza differente: il range d'età dei ragazzi (18-25) e degli anziani (61-68) comprende meno anni di quello di giovani (26-45) e maturi (46-60). Potremmo calcolare l'ampiezza di classe sottraendo il limite inferiore a quello superiore:

```
25-18     45-26     60-46     68-61
[1] 7     [1] 19    [1] 14    [1] 7
```

, ma nella pratica si usano i **limiti reali** della classe. Quando il dato prevede valori decimali, il suo inserimento in una categoria o nell'altra tiene conto dell'approssimazione: per esempio, 25.8 anni → 26 e 45.2 anni → 45 cadono nella

categoria 26-44. Quindi, in realtà i confini di questa categoria sono 25.5 (limite reale inferiore) e 45.5 (limite reale superiore), ottenuti sottraendo 0.5 al limite inferiore e sommando 0.5 al limite superiore[13]. Le ampiezze delle classi sono:

```
25.5-17.5   45.5-25.5   60.5-45.5   68.5-60.5
[1] 8       [1] 20      [1] 15      [1] 8
```

Il **rapporto tra la frequenza di ogni classe e la sua ampiezza** è la **densità di frequenza**, interpretabile come il **numero medio di osservazioni per unità di ampiezza della classe**.

$$h_i = \frac{n_i}{c_i - c_{i-1}}$$

Nel nostro caso:

```
(numeratore<-table(gatti2$eta_discreta))
ragazzi giovani  maturi anziani
     38      22      12       7
(denominatore<-c(25.5-17.5, 45.5-25.5, 60.5-45.5, 68.5-60.5))
[1]  8 20 15  8
densita_frequenza<-numeratore/denominatore
ragazzi giovani  maturi anziani
  4.750   1.100   0.800   0.875
```

La numerosità dei ragazzi in rapporto all'ampiezza della loro classe di appartenenza (la loro densità di frequenza) è di **oltre quattro volte maggiore** di quella dei giovani; la loro frequenza assoluta, invece, è meno di due volte maggiore della frequenza assoluta dei giovani. Confrontando ragazzi e anziani, che hanno una medesima ampiezza di classe, la densità di frequenza dei ragazzi è di oltre cinque volte maggiore: naturalmente, in questo caso il rapporto tra le frequenze assolute delle due classi corrisponde al rapporto tra le loro densità di frequenza (dato che dividiamo N per il medesimo denominatore).

Prima di procedere, altre due parole sulla funzione **order**, che potremmo ritrovare in futuro.
Come anticipato, il senso dell'ordinamento può essere specificato dall'argomento logico `decreasing` (di default `=FALSE`), oppure **scrivendo il nome della variabile facendolo precedere dal segno –** per ordinare in senso decrescente; senza altre specificazioni, l'ordinamento è in senso crescente. Per esempio, se volessimo ordinare il dataframe in base alla **decrescente empatia** per gli animali (variabile `$AES_empatia_animali`), potremmo scrivere:

```
cattivi<-gatti[order(-gatti$AES_empatia_animali),]
```

Nel dataframe `cattivi`, ordinato in senso decrescente, nelle prime righe sono rappresentanti i soggetti più empatici e nelle ultime righe quelli meno empatici:

```
head(cattivi$AES_empatia_animali); tail(cattivi$AES_empatia_animali)
[1] 184 182 174 173 173 171
[1] 122 118 117 106  97  94
```

Una funzione simile, che useremo nel prossimo paragrafo, è **sort(variabile)**: a differenza di `order`, serve per ordinare **una sola distribuzione per volta**.
Infine, un'occhiata a `Freq(distribuzione)`:

```
Freq(gatti$eta)
      level  freq   perc  cumfreq  cumperc
1   [15,20]     4   5.1%        4     5.1%
2   (20,25]    34  43.0%       38    48.1%
3   (25,30]    16  20.3%       54    68.4%
4   (30,35]     3   3.8%       57    72.2%
[omissis]
11  (65,70]     5   6.3%       79   100.0%
```

[13] Per la precisione: il limite inferiore è dato dalla somma del limite inferiore della classe e del limite superiore della classe precedente, divisa per due $(26 + 25)/2 = 25.5$; il limite superiore è dato dalla somma del limite superiore della classe e del limite inferiore della classe successiva, divisa per due: $(45 + 46)/2 = 45.5$.

Quando la variabile è di tipo **numerico**, la funzione decide in autonomia una propria suddivisione in classi di ampiezza costante, di cui mostra frequenze e frequenze cumulate. Il numero di intervalli può essere definito con l'argomento `breaks=` ; ad esempio, per suddividere la distribuzione in quattro parti, scriveremo:

```
Freq(gatti$eta,breaks = 4)
          level  freq   perc  cumfreq  cumperc
1  [17.9,30.5]    54  68.4%       54    68.4%
2    (30.5,43]     6   7.6%       60    75.9%
3    (43,55.5]     9  11.4%       69    87.3%
4    (55.5,68]    10  12.7%       79   100.0%
```

Prima di proseguire:

1. *Calcolate le frequenze assolute, le proporzioni e le percentuali delle tre variabili che descrivono la capacità dei soggetti di discriminare l'intenzione comunicativa dei gatti nei tre contesti: quali commenti potremmo fare sul risultato?*
2. *Anche la distribuzione del livello di istruzione non è ottimale: unite i soggetti con specializzazione post lauream ai laureati, creando la variabile `$istruzione2`; che tipo di variabile avete creato?*
3. *Selezionate solo i soggetti che vivono con un gatto e fate le stesse operazioni del punto 1: l'interpretazione del dato cambia?*
4. *Considerate per tutto il campione la variabile `$empatia_gatti`, che esprime l'autovalutazione sull'empatia specifica per i gatti:*
   a. *descrivete la distribuzione di frequenza della variabile;*
   b. *considerate i punteggi fino a 8 come indicatori di bassa empatia, da 9 a 18 come indicatori di media empatia e da 19 fino al più grande come indicatori di travolgente empatia: dividete la distribuzione della variabile in base a queste tra classi e calcolatene la densità di frequenza.*

Lo script per eseguire tutto quanto richiesto è in fondo alla dispensa, ma è inutile andarlo a vedere senza almeno provarci (e riprovarci, e riprovarci ☺)

3.2.3 Modelli per distribuzioni univariate ordinali

Nelle distribuzioni su scala ordinale, i numeri sono utilizzati per rappresentare la posizione dell'osservazione all'interno della distribuzione ordinata, in senso crescente o decrescente: primo, secondo, terzo.... penultimo, ultimo. Abbiamo quindi una distribuzione di **ranghi**.

Per i prossimi esempi, selezioniamo una parte precisa dei nostri soggetti, che vogliamo siano davvero "esperti" nel campo felino: vogliamo che vivano con un gatto, che siano cresciuti con animali domestici e che abbiano più di 25 anni. Abbondiamo, perciò, con i criteri di selezione:

```
una_vita_con_gatto <- subset(gatti, cresciuto_animali_domestici== "si" & vive_con_gatto== "si" &
  eta >25)
```

In quale altro modo avremmo potuto selezionare i soggetti più maturi?

```
length(una_vita_con_gatto$sogg)
[1] 11
```

Abbiamo 11 soggetti "esperti". Creiamo un mini-dataframe, **`esperti`**, che comprende solo il codice del soggetto esperto e il suo punteggio di empatia verso i gatti: abbiamo già visto come fare.

```
esperti <- data.frame(soggetto= una_vita_con_gatto$sogg, empatia_gatti=
   una_vita_con_gatto$empatia_gatti)
```

```
esperti
   soggetto empatia_gatti
1       S16             6
2       S17             8
3       S19            25
4       S20            22
5       S21             8
6       S28            27
7       S33            19
8       S34            22
9       S48            16
10      S73            22
11      S74            11
```

In quali altri modi avremmo potuto esportare solo il codice identificativo e il punteggio di empatia dei soggetti più maturi ?

Vediamo la **distribuzione di frequenza** dei valori di empatia verso i gatti:

```
table(esperti$empatia_gatti)
 6  8 11 16 19 22 25 27
 1  2  1  1  1  3  1  1
```

La moda è 22, un punteggio piuttosto alto. **Trasformiamo i valori di empatia in ranghi**, ovvero in posizioni all'interno della distribuzione di questi 11 valori, ordinata in senso crescente: il punteggio 6 dovrebbe vedersi assegnato il rango "1", dato che è il più basso, mentre il punteggio 27, il più alto, dovrebbe avere il rango più alto, cioè 11. Con R la trasformazione di punteggi in ranghi è molto semplice: usiamo **`rank(variabile)`**. Notiamo che ogni valore è rappresentato una volta, **tranne** il valore 8, ottenuto da due soggetti, e il valore 22, ottenuto da tre soggetti: vediamo come R gestisce questa situazione:

```
esperti$ranghi<-rank(esperti$empatia_gatti)
esperti
   soggetto empatia_gatti ranghi
1       S16             6    1.0
2       S17             8    2.5
3       S19            25   10.0
4       S20            22    8.0
5       S21             8    2.5
6       S28            27   11.0
7       S33            19    6.0
8       S34            22    8.0
9       S48            16    5.0
10      S73            22    8.0
11      S74            11    4.0
```

Per chiarire meglio cosa è successo, usiamo `order` per ordinare il dataframe secondo una crescente empatia:

```
esperti<- esperti[order(esperti$empatia_gatti),]
esperti
   soggetto empatia_gatti ranghi
1       S16             6    1.0
2       S17             8    2.5
5       S21             8    2.5
11      S74            11    4.0
9       S48            16    5.0
7       S33            19    6.0
4       S20            22    8.0
8       S34            22    8.0
10      S73            22    8.0
3       S19            25   10.0
6       S28            27   11.0
```

Così è lampante: al punteggio 6 è assegnato il primo rango (R_1). La seconda e la terza posizione della distribuzione sono occupate da due soggetti (S_{17} e S_{21}), a pari merito con il punteggio 8: dato che non si può stabilire se l'empatia di

S_{17} viene prima di quella di S_{21} o viceversa, R attribuisce ai due il loro **rango medio**, ovvero la media dei ranghi che avrebbero occupato: $R_{17,21} = (2 + 3)/2 = 2.5$. Si prosegue poi con il punteggio 11, al quarto posto (R_4), con 16 al quinto, 19 al sesto e poi con tre soggetti (S_{20}, S_{34} e S_{73}) di nuovo pari merito: il loro rango medio è $R_{20,34,73} = (7 + 8 + 9)/3 = 8$. Concludiamo con il decimo e l'undicesimo posto per i punteggi 25 (R_{10}) e 27 (R_{11}). I valori pari merito nella distribuzione si definiscono **ties**: ricordate questa terminologia, che ritroveremo nei test non parametrici.

La funzione `rank` ha due argomenti opzionali: **`ties.method=`** specifica come trattare i valori pari merito appena spiegati; di default, come abbiamo visto, è `ties.method= "average"`, ovvero sostituisce i ties con il loro rango medio. È possibile usare anche altri metodi: per esempio, `"min"` e `"max"` assegnano a tutti i valori pari merito rispettivamente il rango più piccolo e il rango più alto che toccherebbe loro (nelle gare di atletica si usa questo ordinamento):

```
esperti$ranghi<-rank(esperti$empatia_gatti, ties.method = "min")
esperti$ranghi
 [1]  1  2  2  4  5  6  7  7  7 10 11

esperti$ranghi<-rank(esperti$empatia_gatti, ties.method = "max")
esperti$ranghi
 [1]  1  3  3  4  5  6  9  9  9 10 11
```

Potete curiosare con `help(rank)` per vedere altre tipologie di sostituzione, ma noi, noiosamente, useremo sempre il rango medio.

Infine, l'altro argomento opzionale è l'argomento logico **`na.last=`**, che gestisce i ranghi nel caso di dati mancanti: di default è `na.last=TRUE`, il che comporta che i casi con dato mancante sono posti in fondo alla distribuzione dei ranghi. Se si indica `=FALSE`, invece, questi casi sono messi nei primi posti; se `na.last=NA`, i dati mancanti sono omessi dai ranghi, se `na.last="keep"`, sono mantenuti nella distribuzione con rango NA.

All'interno della distribuzione ordinata troviamo l'indice di tendenza centrale per il livello ordinale, ovvero la **mediana**Errore. Il segnalibro non è definito.: è la **modalità dell'osservazione che divide la distribuzione ordinata in due parti uguali**, ovvero quel valore della distribuzione ordinata al di sopra o al di sotto del quale cade un ugual numero di osservazioni. Quando la numerosità della distribuzione (N) è dispari, la mediana è il valore che occupa il posto centrale, cioè il posto $(N + 1)/2$ della distribuzione ordinata; quando N è pari, si assume come mediana la media aritmetica dei termini che occupano le due posizioni centrali della graduatoria, ossia le posizioni $N/2$ e $(N/2) + 1$.

Nel nostro esempio, il punteggio sopra e sotto il quale cade la stessa numerosità di osservazioni è **19**:

```
table(esperti$empatia_gatti)
 6  8 11 16 19 22 25 27
 1  2  1  1  1  3  1  1
       5              5
```

Molto più rapidamente, useremo **`median(distribuzione)`**:

```
median(esperti$empatia_gatti)
[1] 19
```

Oltre a essere un indice di tendenza centrale, la mediana è un particolare indice di posizione: gli **indici di posizione** indicano – appunto – i valori corrispondenti a specifiche posizioni all'interno della distribuzione ordinata.

I **quartili** sono i tre valori che dividono la distribuzione ordinata in quattro parti uguali: si indicano con q_1, q_2 e q_3. Il primo quartile è il valore al di sotto del quale cade il 25% della distribuzione ordinata (la sua posizione si ricava con $\frac{1}{4} \times (N + 1)$); il secondo quartile è il valore al di sotto del quale cade il 50% della distribuzione ordinata, e quindi corrisponde alla mediana (posizione: $\frac{2}{4} \times (N + 1)$); il terzo quartile è il valore al di sotto del quale cade il 75% della distribuzione ordinata (posizione: $\frac{3}{4} \times (N + 1)$). I valori che cadono tra q_1 e q_3 costituiscono il **range interquartilico** (***IR*** o ***IQR***) o **differenza**

interquartilica, cioè il 50% dei valori che occupano le posizioni centrali della distribuzione ordinata. Altri indici di posizione che determinano diverse partizioni della distribuzione sono i **decili** (nove quantità che la suddividono in dieci parti di pari numerosità), e i **centili** (o **percentili**: novantanove quantità che la dividono in cento parti di uguale numerosità). Quartili e percentili (i decili sono meno diffusi) sono molto usati nella clinica per individuare i valori corrispondenti ai casi eccezionali, verso l'alto o verso il basso.
Tutti questi indici di posizione sono **quantiliErrore. Il segnalibro non è definito.**: **valori che dividono la distribuzione ordinata di una variabile aleatoria (capitolo 6) in intervalli uguali**.

In R, il 1°, il 2° (mediana) e il 3° quartile sono indicati nella funzione **`summary(distribuzione)`**; avevamo già visto questa funzione applicata alla descrizione di un dataframe, ma ora la usiamo per descrivere una sola variabile:

```
summary(esperti$empatia_gatti)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.00    9.50   19.00   16.91   22.00   27.00
           ↑                       ↑
      1° quartile             3° quartile
```

Per conoscere il valore corrispondente a una qualsiasi posizione all'interno di una distribuzione ordinata, si può usare `quantile(x= distribuzione campionaria, probs= posizione)`. L'argomento `probs=(valore da 0 a 1)` serve a indicare la posizione del **percentile** desiderato: di default, la serie dei quantili mostrati è: `probs=c(0, 0.25, 0.5, 0.75 1)`, cioè gli stessi indici di posizione di `summary` espressi però come percentili.
Per farne qualche esempio, usiamo la più ampia distribuzione `$empatia_gatti` per tutti i 79 soggetti del dataframe `gatti`. Considerando esperti e non esperti nel complesso, la **mediana** dell'empatia è:

```
median(gatti$empatia_gatti)
[1] 15
```

Ampliamo le informazioni:

```
quantile(gatti$empatia_gatti)
  0%  25%  50%  75% 100%
 3.0  8.0 15.0 18.5 27.0
```

Possiamo chiedere qualsiasi altra serie di quantili:

```
quantile(gatti$empatia_gatti, probs= c(.05, .33, .98))
  5%   33%   98%
 3.00 10.00 23.88
```

Oltre a `summary`:

```
summary(gatti$empatia_gatti)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
3.00    8.00   15.00   13.62   18.50   27.00
```

, potremmo usare **`fivenum(distribuzione)`**, che ritroveremo parlando dei boxplot (capitolo 4) e che riporta i cinque descrittori consigliati da **Tukey**[14] ("*As a standard summary for general use, the 5-number summary provides about the right amount of detail*"; Tukey, 1977): sono gli stessi di `summary`, **tranne la media** – e le loro etichette ☺. Se ci sono dati mancanti, aggiungete l'argomento logico `na.rm=TRUE`, per eliminarli prima di calcolare i quantili, altrimenti otterrete un NA come solo output.

[14] Le cinque statistiche contengono informazioni sul range (min, max), la posizione centrale (mediana) e la dispersione (1° e 3° quartile) della distribuzione, ritenute da Tukey più che sufficienti per descrivere e, se necessario, confrontare distribuzioni. Essendo riferite al livello di misura ordinale, possono applicarsi anche a scale a intervalli o rapporti equivalenti, e, diversamente dalla media e dalle misure di dispersione basate sulla media (varianza, deviazione standard), non sono influenzate dalla presenza di casi anomali che ne distorcono la stima (sono statistiche **robuste**): torneremo diffusamente su tutti questi aspetti.

```
fivenum(gatti$empatia_gatti)
[1]   3.0     8.0    15.0    18.5    27.0
```

Min *1° quartile* *Mediana* *3° quartile* *Max*

In realtà, il primo e il terzo valore di `fivenum` non sono esattamente corrispondenti al primo e al terzo quartile, dato che sono calcolati rispettivamente come la mediana della prima metà della distribuzione ordinata e come mediana della seconda metà della distribuzione ordinata... ma a fini pratici questa differenza è sostanzialmente irrilevante:

```
vettore<-rnorm(n=1000,mean = 0, sd=1) [distribuzione aleatoria normale: le costruiremo nel capitolo 5]
fivenum(vettore)
[1] -2.88761111
    -0.62609132
     0.06271403
     0.69631112
     3.23523883

quantile(vettore,probs =
c(.25, .75))
       25%        75%
-0.6215482  0.6953012

summary(vettore)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.88761 -0.62155  0.06271  0.03339  0.69530  3.23524
```

Notate che non dovete ordinare la distribuzione prima del calcolo: R lo fa per voi.

Prima di proseguire:

*1. Considerate **tutti** i soggetti nel dataframe `gatti`: costruite la distribuzione la distribuzione delle frequenze percentuali assolute della variabile `$autovalutazione_relazione_gatto` e commentatela: è stata una buona idea? Perché?*

2. Calcolate il primo e il terzo quartile della variabile `$autovalutazione_relazione_gatto` e interpretatene l'output.

3. Usate i quantili così individuati per creare la variabile di raggruppamento `$amiconi`, in cui i soggetti che faticano a entrare in relazione con un gatto sono individuati dal livello "scarsa relazione", quelli così così dal livello "media relazione" e quelli che pensano come un gatto dal livello "buona relazione".

4. Calcolate la densità di frequenza della variabile `$amiconi`.

Lo script per eseguire tutto quanto richiesto è in fondo alla dispensa, ma è inutile andarlo a vedere ecc. ecc.

3.2.4 Modelli per distribuzioni univariate a intervalli e rapporti equivalenti

Finalmente, possiamo usare tutte le proprietà dei numeri per dati misurati a livello intervallare e a rapporti (scale metriche): oltre a descrivere distribuzioni di frequenza e individuare moda, mediana e altri indici di posizione, in queste scale posso calcolare la media con indice di tendenza centrale e varie statistiche come modelli per rappresentare la dispersione: devianza, varianza, deviazione standard.

Più precisamente, dovremmo parlare di **medie**, al plurale, dato che possiamo stimare diversi tipi di medie:

- la **media aritmetica** μ è la sommatoria di tutte le osservazioni presenti divisa per la numerosità totale;

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- la media **quadratica** μ_q è la radice quadrata della media aritmetica (è la più influenzata dai valori estremi, molto piccoli o molto grandi, della distribuzione);

$$\mu_q = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

- la media **geometrica** di N numeri μ_g è la radice N-esima del prodotto dei numeri tra loro. Si usa in sostituzione della media aritmetica quando si deve ottenere una media di N rapporti y_i/x_i. Nella pratica si usano i logaritmi, invece della radice: la media geometrica è l'esponenziale della media aritmetica dei logaritmi dei dati;

$$\mu_g = \sqrt[N]{x_i \times x_i \times \dots x_N}$$

- la media **armonica** di N numeri μ_a è il reciproco della media aritmetica dei reciproci dei numeri stessi (si usa, per esempio, per calcolare la velocità media, intesa come media armonica delle velocità: dato il reciproco di una velocità, rappresenta il tempo necessario per percorrere una unità di spazio).

$$\mu_a = \frac{N}{\sum_{i=1}^{N} \frac{1}{x_i}}$$

Tra le medie calcolate su una stessa distribuzione, la relazione è: $\mu_a < \mu_g < \mu < \mu_q$.

Tranquilli, noi lavoreremo con la media aritmetica (da qui in avanti, se non altrimenti specificato, con "media" si intenderà media aritmetica) e qualche sua piccola modifica; se mai dovrete usare media geometrica e armonica (che non sono disponibili tra le statistiche di base di R) e non volete impicciarvi con le formule, diversi package hanno funzioni dedicate: per esempio, scaricate e installate **psych** e usate le funzioni **geometric.mean(distribuzione)** e **harmonic.mean**`Errore. Il segnalibro non è definito.`**(distribuzione)**. Ci sono anche `Gmean(distribuzione)` e `Hmean(distribuzione)` di `DescTools`, package che useremo più di `psych`.

```
geometric.mean(gatti$empatia_gatti)
[1] 11.89132
harmonic.mean(gatti$empatia_gatti)
[1] 9.811161
```

La **media** è uno dei modelli più semplici e facilmente intuitivi della statistica: è un **valore ipotetico** (**non presente** nel set di dati campionari), che **sintetizza** nella **maniera più fedele possibile** l'intera distribuzione di dati. Usiamola, allora, per approfondire meglio il concetto di goodness of fit / adattamento di un modello introdotto a pagina 49.

Per stabilire la **goodness of fit** di un modello, la cosa più ovvia è **confrontare il modello con i dati reali**: tanto più piccola è la **differenza** tra dati reali e modello, tanto migliore è quest'ultimo. Quindi, confrontiamo il modello - media rispetto alla reale distribuzione da cui è calcolata e valutiamone l'adattamento.

Facciamolo con un esempio: torniamo a usare il dataframe degli undici esperti di vita con gatti:

```
esperti
   soggetto empatia_gatti ranghi
      S16             6      1
      S17             8      3
      S21             8      2
      S74            11      4
      S48            16      5
      S33            19      6
      S20            22      9
      S34            22      8
      S73            22      7
      S19            25     10
      S28            27     11
```

La variabile `$ranghi` non ci serve più: cogliamo l'occasione per imparare come si fa a **eliminare una colonna**, cioè una variabile, da un dataframe. Si usa la struttura del dataframe, anteponendo il segno − alla colonna (o alle colonne) da eliminare. Noi vogliamo cancellare la terza variabile e mantenere tutte le righe, quindi scriveremo:

```
esperti<-esperti[,-3]
```

Infatti:

```
str(esperti)
'data.frame':11 obs. of  2 variables:
$ soggetto       : Factor w/ 79 levels "S1","S10","S11",..: 8 9 14 72 43 27 13 28 71 11 ...
$ empatia_gatti  : int  6 8 8 11 16 19 22 22 22 25 ...
```

Se volessimo **eliminare una riga**, ovvero un soggetto, o **più** righe, anteporremo il segno meno al numero di riga / di righe: `dataframe[-riga,]`. Per **eliminare una cella**, si assegna NA alla cella: `dataframe[riga,colonna]<-NA`

Ora possiamo concentrarci sulla media dell'empatia per i gatti: sappiamo già che la funzione dedicata è **mean(distribuzione)**, ma possiamo trovarla anche nell'output di `summary(distribuzione)`. Ricordiamo che se ci

fossero dati mancanti, in `mean` dovremmo aggiungere l'argomento logico **`na.rm=TRUE`**, che indica di rimuovere (`rm`) i dati mancanti (`NA`): se ce ne dimenticassimo, in output leggeremmo un frustrante "NA".

```
mean(esperti$empatia_gatti)
[1] 16.90909
```

Dato che il punteggio massimo ottenibile è 30, il punteggio medio di questi soggetti sembra descrivere una **moderata empatia** verso i gatti, non troppo bassa, ma nemmeno troppo alta. Il punto è: questo **modello-media** si adatta bene a tutti i soggetti? **Se, non conoscendo nulla di uno dei soggetti tranne la media del suo gruppo di appartenenze, facessimo previsioni sulla sua empatia, le nostre previsioni sarebbero affidabili?** Il modello – media per questa variabile è sufficientemente buono?

Verifichiamolo, confrontando l'empatia di ogni soggetto con la media: creiamo la variabile **`$differenza_media`**, che rappresenta le differenze o **scarti** o **deviazioni dalla media** di ogni punteggio:

```
(esperti$differenza_media<-(esperti$empatia_gatti-mean(esperti$empatia_gatti))
[1] -10.9090909  -8.9090909  -8.9090909  -5.9090909  -0.9090909   2.0909091
[7]   5.0909091   5.0909091   5.0909091   8.0909091  10.0909091
```

O, se è più chiaro:

```
esperti
   sogg empatia_gatti ranghi differenza_media
S16             6    1.0      -10.9090909
S17             8    2.5       -8.9090909
S21             8    2.5       -8.9090909
S74            11    4.0       -5.9090909
S48            16    5.0       -0.9090909
S33            19    6.0        2.0909091
S20            22    8.0        5.0909091
S34            22    8.0        5.0909091
S73            22    8.0        5.0909091
S19            25   10.0        8.0909091
S28            27   11.0       10.0909091
```

Per il soggetto S_{48} la media è un ottimo modello, rappresenta molto bene la sua reale empatia; però, per i soggetti S_{16}, S_{17} e S_{21} la media sbaglia gravemente, sovrastimando la loro empatia, così come sbaglia grossolanamente in senso opposto, sottostimandola, per i soggetti S_{19} e S_{28}. Per gli altri soggetti, le differenze sono meno marcate.

Nel complesso, questi **errori** nella stima / **scarti dalla media** si **annullano**, dato che per proprietà della media **la somma algebrica degli scarti è uguale a zero**:

```
round(sum(esperti$differenza_media),3)
[1] 0
```

Il modello sembrerebbe quindi **perfetto**, dato che avrebbe un errore complessivo pari a zero, ma abbiamo toccato con mano che evidentemente non è così. Quindi, per avere una stima **realistica** della **goodness of fit** del modello – media, eliminiamo i segni degli scarti **elevando al quadrato gli errori** del modello, cioè gli scarti dalla media:

```
esperti$scarti_quadrato<- esperti$differenza_media^2
esperti:
   soggetto empatia_gatti differenza_media scarti_quadrato
1       S16             6        -10.90909     119.0082446
2       S17             8         -8.90909      79.3718846
5       S21             8         -8.90909      79.3718846
11      S74            11         -5.90909      34.9173446
9       S48            16         -0.90909       0.8264446
7       S33            19          2.09091       4.3719046
4       S20            22          5.09091      25.9173646
8       S34            22          5.09091      25.9173646
10      S73            22          5.09091      25.9173646
3       S19            25          8.09091      65.4628246
6       S28            27         10.09091     101.8264646
```

La **somma degli errori al quadrato** (**Sum of Squared errors: SS**), cioè la somma degli scarti dalla media al quadrato, è la **devianza**. La devianza, **indice di dispersione** della distribuzione attorno al valore centrale media, è quindi un **indice di goodness of fit del modello**:

```
(devianza_esperti<-sum(esperti$scarti_quadrato))
[1] 562.9091
```

La devianza offre una soluzione al problema della quantificazione della goodness of fit, ma **ha un problema** quando si tratta di confrontare la qualità di modelli diversi: essendo una **somma** di errori, distribuzioni composte da pochi casi ottengono inevitabilmente devianze più piccole di quelle rilevabili in distribuzioni più numerose, anche se il modello relativo alle distribuzioni piccole fosse meno buono. Una soluzione semplice è quella di **ponderare la somma degli errori per la numerosità della distribuzione**: **dividiamo la devianza** per N, o, meglio **per i gradi di libertà [degree of freedom, df] $df = N - 1$** (**correzione di Bessel**), facendone la media: otteniamo così un altro indice di dispersione, ovvero l'indice di **goodness of fit ponderato** del modello (**Means of Squared errors: MS**), cioè la **varianza**.

```
(varianza_esperti<-devianza_esperti/(11-1))
[1] 56.29091
```

Intermezzo: cosa sono i gradi di libertà Errore. Il segnalibro non è definito. **di una distribuzione?**

I gradi di libertà (**gdl o df, degrees of freedom**) di una distribuzione corrispondono al **numero di valori indipendenti della distribuzione**. I valori indipendenti sono **quelli il cui valore non dipende da alcun altro dato**. Facciamo un esempio facile per avvicinarci al concetto. Immaginiamo che a Natale vi regalino un pacchetto ben incartato che riporta l'etichetta: "In questa scatola ci sono 5 gettoni d'oro il cui valore medio è uguale a 250 euro". Per prima cosa ringraziate calorosamente, poi cominciate a estrarre a uno a uno i gettoni, su cui è impresso il relativo valore. Non potete sapere **con certezza** quale sarà il valore del primo gettone, che scoprite essere pari a 150 euro; non potete neppure sapere quale sarà il valore del secondo gettone, che una volta estratto sarà pari a 350 euro. Ancora, non potete sapere con certezza quale sarà il valore del terzo gettone, che si rivelerà essere pari a 250 euro. A questo punto, sapete che il valore medio dei tre gettoni estratti è pari a quello annunciato dal biglietto, cioè 250 euro, ma, nuovamente, in base a questa informazione non potete sapere in anticipo quale sarà il valore del quarto gettone. Il quarto gettone si scopre valere 50 euro: **ora sì** che potete annunciare in anticipo quale sarà il valore del quinto gettone, perché il quinto gettone dovrà far rispettare il principio per cui la somma degli scarti dalla media deve essere pari a zero. La distribuzione dei quattro gettoni estratti è :

```
quattro<-c(150,350,250, 50)
```

E la somma dei loro scarti dalla media è in "debito" di 200 euro dalla media annunciata:

```
sum(quattro-250)
[1] -200
```

Quindi, l'ultimo gettone estratto deve essere di 200 euro superiore alla media per far tornare i conti, ovvero $gettone_5 = 450$ euro

```
cinque<-c(150,350,250, 50, 450)
sum(cinque-250)
[1] 0
```

Perciò, nella nostra distribuzione di cinque gettoni, **quattro di loro hanno un valore indipendente** gli uni dagli altri, ma uno no, perché è vincolato al rispetto della proprietà della media per cui la somma delle differenze dalla media deve essere $\sum(scarti\ dalla\ media) = 0$. I **gradi di libertà per la media sono quindi pari al numero di osservazioni meno 1: $N - 1 = 4$.**

Generalizzando l'esempio, quattro osservazioni che costituiscono un campione estratto da una popolazione possono assumere qualsiasi valore previsto nella popolazione. Però, se vogliamo usare questo campione di osservazioni per

calcolare la varianza della popolazione (e di solito siamo decisamente più interessati a stimare l'errore del modello rispetto alla popolazione, piuttosto che al campione) dobbiamo usare la media del campione come stima della media della popolazione. Dobbiamo quindi **tenere un parametro costante**. Diciamo che la **media del campione** è 10: quindi, assumiamo che la media della popolazione sia 10 e teniamo costante questo valore. Con questo parametro fisso, i valori delle quattro osservazioni possono variare come vogliono? No: per mantenere costante la media, solo tre ($N-1$) di essi possono variare. Se fossero 8, 9, 11 e 12 ($media = 10$), e noi cambiassimo tre di questi valori in 7, 15 e 8 ($totale = 30$), allora il quarto valore **deve** essere 10, per mantenere costante la media a 10. Di conseguenza, **se teniamo un parametro costante, allora i gradi di libertà devono essere uno in meno rispetto al numero totale**. Ecco perché, quando usiamo un campione per stimare la varianza di una popolazione, dobbiamo dividere la devianza per $N-1$, invece che per N.

Invece di fare i calcoli, con R chiederemo la varianza di una distribuzione con `var(distribuzione)`:

```
var(esperti$empatia_gatti)
[1] 56.29091
```

Tra le statistiche di base non c'è una funzione per richiedere la devianza di una distribuzione, ma, sapendo come si ricava, basta **moltiplicare** `var` **per i gradi di libertà** $\boldsymbol{df = N-1}$:

```
var(esperti$empatia_gatti)*(11-1)
[1] 562.9091
```

Possiamo quindi dare per assodato che la media **è** un semplice modello statistico che si può adattare ai dati – per alcuni soggetti meglio, per altri peggio? Allora, possiamo azzardarci ad esprimere questo concetto anche così:

$$\boldsymbol{dato\ reale_i = (modello) + errore_i}$$

Ovvero: il dato realmente osservato è dato da / è uguale a / è **predetto** dal modello statistico utilizzato (nel nostro caso, la media) più una quota di errore intrinseca al modello (nel nostro caso, lo scarto dalla media). È un'equazione tanto semplice quanto onnipresente; la ritroveremo nel capitolo 9, in cui sostituiremo "modello" non con "media" ma con i due parametri che costituiscono un modello lineare (intercetta e coefficiente angolare).
Concludiamo il nostro semplice modello-media azzardandoci ad esprimere anche il concetto di devianza e varianza come indici di goodness of fit, in questo modo:

$$\boldsymbol{devianza = \sum (dato\ reale - modello)^2}$$

Ovvero: la qualità di un modello è analizzata **valutando le deviazioni dal modello dei dati reali**: stimiamo i modelli confrontando i dati osservati con il modello che abbiamo adattato ai dati, elevando al quadrato le differenze tra dati e modello. Anche questa equazione è ubiqua; la ritroveremo per la seconda (ma non l'ultima) volta nel capitolo 9.

Facciamo un esempio concreto confrontando questo modello–media per gli esperti con lo stesso modello–media per i **soggetti non esperti**, cioè quelli che O non hanno un gatto, O non sono cresciuti con un animale domestico, O hanno 25 anni, O hanno una combinazione di due caratteristiche, ma non le possiedono tutte e tre. Usiamo il pensiero divergente per farlo in due modi, semplificando al minimo.
Cominciamo a creare una variabile di raggruppamento (fattore) nel dataframe `gatti` che chiamiamo **`$expertise`**: gli esperti saranno gli 11 soggetti che hanno più di 25 anni, sono cresciuti con animali e hanno un gatto.

```
gatti$expertise[gatti$vive_con_gatto== "si" & gatti$cresciuto_animali_domestici== "si" &
 gatti$eta >25]<-"esperti"
```

Ora ci sono 11 soggetti correttamente etichettati e 68 dati mancanti NA:

```
table(gatti$expertise,exclude=NULL)
esperti    <NA>
     11      68
```

I dati mancanti sono i soggetti non esperti: usiamo l'etichetta NA per istruire R ad assegnare loro l'etichetta "non esperti" con **is.na**: è la funzione che R usa per indicare quali dati sono NA (§2.2.6):

```
gatti$expertise[is.na(gatti$expertise)]<-"non esperti"
table(gatti$expertise,exclude=NULL)
    esperti non esperti        <NA>
         11          68           0
```

Oppure, in un solo passaggio:

```
gatti$expertise <- ifelse(gatti$vive_con_gatto== "si"  & gatti$cresciuto_animali_domestici==
   "si" & gatti$eta >25, "esperti", "non esperti")
table(gatti$expertise)
    esperti non esperti
         11          68
```

Bene: 11 esperti, di cui conosciamo già il modello – media dell'empatia, e 68 non esperti di cui ci accingiamo a calcolarlo; l'attesa è che la loro empatia media sia minore, rispetto a quella degli esperti. In analogia alla creazione del subset `esperti` fatto prima, potremmo creare un subset `non_esperti`, usando come filtro l'etichetta "non esperti" della variabile `$expertise`, e chiedere poi la media della variabile `non_esperti$empatia`:

```
non_esperti<-subset(gatti, expertise=="non esperti")
mean(non_esperti$empatia_gatti)
[1] 13.08824
```

C'è un modo **molto più rapido** e che **privilegeremo quando sarà possibile**: **tapply**Errore. Il segnalibro non è definito.**(X= misura, INDEX= fattore, FUN= funzione)** applica una funzione a ciascun gruppo di valori definito da un livello (o da una combinazione di livelli) di uno o più fattori. La funzione può essere una statistica descrittiva, come per questo esempio, ma anche un test inferenziale, come vedremo a suo tempo. I suoi argomenti sono: `X=` una misura (variabile dipendente), `INDEX=` una o più variabili `factor`, `FUN=` una qualsiasi funzione che viene applicata su `y`, separatamente per ciascuno dei livelli della variabile `factor`. Come torneremo a imparare nel capitolo dedicato all'ANOVA fattoriale, per indicare combinazioni dei livelli di più fattori scriviamo `INDEX=list(fattore1, fattore2)`: dovreste ricordare gli oggetti di classe `list`...
Se ci sono dati mancanti e la funzione richiesta in `FUN=` ha bisogno di istruzioni per gestirli, va aggiunto anche l'apposito argomento; per esempio, richiedendo una media in presenza di dati mancanti, si aggiunge `na.rm=TRUE`[15].

Applica *alla variabile $empatia_gatti,* *per ogni livello di $expertise,* *la funzione "calcola media"*

```
tapply(gatti$empatia_gatti,   gatti$expertise, mean)
    esperti non esperti
   16.90909    13.08824
```

[15]Una funzione simile nella struttura, ma che applica una funzione a un intero dataframe diviso in base a variabili `factor` è: `by(data= dataframe o matrice, x, funzione)`

Come ci aspettavamo, l'empatia media dei non esperti è più bassa. Com'è la qualità dei due modelli - media costruiti? Usiamo ancora `tapply` per richiedere la varianza dei due gruppi:

```
tapply(gatti$empatia_gatti, gatti$expertise, var)
    esperti non esperti
   56.29091    33.12643
```

La somma degli errori al quadrato ponderata per la numerosità (la varianza) è **più bassa per il gruppo dei non esperti**: **la loro media è quindi un modello migliore per descrivere l'empatia** in maniera sintetica e affidabile rispetto alla media degli esperti. Conoscere la media dei non esperti permetterà, quindi, di fare ipotesi più realistiche e scommesse più vincenti sul livello di empatia di un soggetto appartenente a questo gruppo.

Se ci fossimo basati sulla devianza, non considerando che il gruppo dei non esperti è oltre sei volte maggiore del gruppo di esperti, avremmo tratto conclusioni **opposte** – e sbagliate:

```
#varianza esperti moltiplicata per N-1
#cioè devianza esperti
56.29091*10
[1] 562.9091
```

```
#varianza non esperti moltiplicata per N-1
#cioè devianza non esperti
33.12643*67
[1] 2219.471
```

Vediamo l'ultimo degli indici di dispersione per variabili intervallari e a rapporti. Devianza e varianza sono scarti al quadrato: mentre la media è espressa nella stessa unità di misura del carattere, la varianza è il quadrato di tale unità di misura. Per descrivere i dati usando un indicatore di tendenza e un indice di dispersione con la stessa base, mettiamo la varianza sotto radice quadrata, ottenendo così la **deviazione standard**Errore. Il segnalibro non è definito. o **scarto quadratico medio**. La funzione per richiedere la deviazione standard in R è **`sd(distribuzione)`: inseriamola in `tapply` per conoscere le deviazioni standard dei due gruppi:**

```
tapply(gatti$empatia_gatti,gatti$expertise,sd)
    esperti non esperti
   7.502727    5.755556
```

L'empatia degli esperti è quindi pari a 16.9 ± 7.5, quella dei non esperti è pari a 13.09 ± 5.8 punti: l'empatia dei non esperti è più bassa, ma questo gruppo è più coeso attorno al valore centrale rispetto all'altro.

DEVIANZA	VARIANZA	DEVIAZIONE STANDARD
$D = \sum_{i=1}^{N} (x_i - \mu)^2$	$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N-1}$	$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N-1}}$

Una notazione per le formule: le statistiche riferite a uno specifico campione, si indicano con caratteri latini (m, s^2, s), mentre quando si riferiscono a popolazioni teoriche si indicano con i corrispettivi caratteri greci (μ, σ^2, σ).

Concludiamo il discorso sul modello-media esplorando un altro possibile modo di calcolare la media di una distribuzione. Se siamo in presenza di una distribuzione con casi anomali all'uno e/o all'altro estremo (**coda**) della distribuzione, cioè molto bassi o molto alti, potrebbe essere opportuno eliminarli **prima di calcolare la media**, dato che possono distorcerne il valore in maniera non irrilevante: la media è infatti sensibile ai valori estremi. Vedremo nel capitolo 9 come individuare con precisione i casi anomali (*outlier* univariati) per migliorare il fit di un modello; un'altra possibilità, più "grezza", è quella di **eliminare una quota prefissata di casi alle due estremità della distribuzione**, per esempio il 2% dei casi più bassi e il 2% dei casi più alti, indipendentemente dal fatto che siano realmente outlier. La media calcolata su questa **distribuzione troncata** alle estremità si definisce **trimmed mean** (appunto, "troncata"). R utilizza l'argomento **`trim=proporzione di casi da eliminare`** nella funzione `mean` per calcolare la media troncata, accettando una proporzione massima = 0.5 casi troncati per coda. Per esempio:

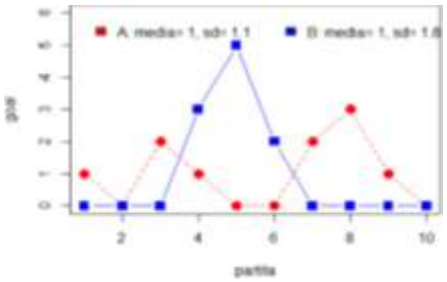
```
mean(esperti$empatia_gatti);mean(esperti$empatia_gatti,trim=.2)
[1] 16.90909
[1] 17.14286
```

Concludiamo con un *caveat* che sarà valido per il resto della materia:

> Descrivere una distribuzione con un indicatore di tendenza centrale **senza** affiancarvi l'informazione sulla dispersione è **sbagliato**. L'interpretazione necessita di entrambe le informazioni.

Facciamo un esempio del perché sia sbagliato, usando la media. Vediamo il profilo dei goal di due giocatori, A e B, nelle 10 partite di campionato che hanno entrambi giocato.

```
gioc_A<-c(1,0,2,1,0,0,2,3,1,0)
gioc_B<-c(0,0,0,3,5,2,0,0,0,0)
mean(gioc_A)
[1] 1
mean(gioc_B)
[1] 1
sd(gioc_A)
[1] 1.054093
sd(gioc_B)
[1] 1.76383
```



Entrambi hanno una media di gol $\bar{x} = 1$, ma il profilo del loro rendimento è chiaramente diverso: il giocatore A ha una minore dispersione, quindi un rendimento molto più costante, mentre il giocatore B ha una deviazione standard maggiore, indice di prestazioni decisamente imprevedibili. Impareremo a fare questo grafico nel capitolo seguente.

Prima di proseguire:

*1. Considerando **tutti** i soggetti nel dataframe `gatti`, calcolate moda, mediana e media della variabile `$AES_empatia_animali` e commentate il dato, sapendo che il punteggio minimo teoricamente ottenibile è 22 e il massimo teoricamente ottenibile è 198;*

2. a. Calcolate gli indici di dispersione della variabile `$AES_empatia_animali` per tutti i soggetti;

b. individuate i soggetti che rappresentano il 25% inferiore della distribuzione ed etichettateli come "antropocentrici", e tutti gli altri come "non antropocentrici". Verificate di aver correttamente dicotomizzato la distribuzione come richiesto

3. Calcolate il modello media della variabile `$AES_empatia_animali` per chi è cresciuto con un animale domestico e confrontatelo con quello di chi non è cresciuto con un animale domestico: quale modello si adatta meglio ai dati? Commentate i due modelli rispetto all'ipotesi che l'empatia non sia un tratto innato, ma una capacità che si può addestrare.

Lo script per eseguire tutto quanto richiesto ecc. ecc.

3.3 `DescTools: Desc`

Un package ricco di funzioni – scorciatoia, che abbiamo già assaggiato e useremo più volte nei capitoli successivi, è `DescTools`. Come suggerisce il nome, è dedicato essenzialmente alla descrizione dei dati, ma non disdegna di occuparsi di alcuni dei test inferenziali che vedremo. In questo paragrafo ci interessa soprattutto la funzione `Desc(oggetto)`, che fornisce in un solo output molte delle informazioni che abbiamo richiesto nelle pagine precedenti con le funzioni di base. Si applica a oggetti di classe diversa, e le informazioni che fornisce sono coerenti con la classe dell'oggetto: qui vediamo il caso di distribuzioni univariate di classe `character` o `factor` (`Desc` non le distingue), `numeric` o `integer`, `table` (a una via, cioè per una sola distribuzione: vedremo nel capitolo 6 le tabelle a due vie).

Cominciamo con il caso più semplice: una variabile nominale, dicotomica. Usiamo `gatti$genere` e scriviamo `Desc(gatti$genere)`. Gli argomenti opzionali sono molti, potete curiosare nell'`Help` della funzione, o, se usate `RStudio`, nei suggerimenti durante la digitazione:

```
Desc(gatti$genere)
-------------------------------------------------------------------------
gatti$genere (factor - dichotomous)

  length      n    NAs unique
      79     79      0      2
         100.0%   0.0%

   freq   perc  lci.95  uci.95'
F    48  60.8%   49.7%   70.8%
M    31  39.2%   29.2%   50.3%

' 95%-CI (Wilson)
```

← variabile genere, di classe `factor` e dicotomica

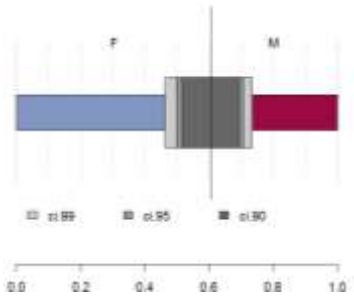
← ci sono 79 osservazioni (`length`), di cui 79 (100%) non missing (`n`) e nessuna (`0, %`) missing (`NAs`); compaiono due valori diversi (`unique`), M e F

← Ci sono 48 F e 31 M (frequenze assolute: `freq`), corrispondenti al 60.8 e 39.2% dei casi totali (`perc`).

← NON conosciamo ancora **l'intervallo di fiducia** (**CI**, confidence interval), che affronteremo nel capitolo 5: possiamo ignorarlo, per il momento.

SOLO per chi voglia anticipare un po' il *CI* o *confidence interval*: entro il range di questo intervallo possiamo trovare con una ragionevole probabilità (95%) le percentuali di donne e di uomini che dovrebbero comporre le stratificazioni per genere nella popolazione da cui abbiamo estratto il campione, in campionamenti ripetuti. In sostanza, le donne compongono il 60.8% del campione, e con il 95% di probabilità dovrebbero costituire tra il 49.7% (**`lci`**: **lower confidence interval**, limite inferiore del *CI*) e il 70.8% (**`uci`**: **upper confidence interval**, limite superiore del *CI*) della popolazione da cui abbiamo estratto i soggetti. Idem per i maschî. Il *CI* è stato calcolato con il metodo di **Wilson** (1927) per una proporzione dicotomica (binomiale), come R dichiara nel suo output (**`wilson`**).

Queste informazioni sono riassunte nel **grafico** che viene prodotto automaticamente insieme all'output di `Desc` (`plotit= TRUE`, di default). Anche se affronteremo in dettaglio i grafici nel prossimo capitolo, nulla vieta di dargli un'occhiata.



In **ascissa** troviamo il **range delle proporzioni** da 0 a1: le barre blu e rossa rappresentano la **proporzione cumulata** di donne (F) e uomini (F). La **sottile** barra verticale separa la proporzione di donne (.608) da quella degli uomini. I rettangoli nelle tre sfumature di grigio corrispondono all'ampiezza degli intervalli di fiducia, con tre diversi gradi di verosimiglianza: 90% (`ci.90`), 95% (quello riportato nell'output testuale: `ci.90`) e .99% (`ci.90`).

Ora descriviamo una variabile nominale a più di due livelli (ricordiamo che `DescTools` non fa differenza tra `character` e `factor`): proviamo con `gatti$stato_civile`. `main= "titolo"` aggiunge un titolo a output e grafico, mentre `ord= "asc"` predispone l'ordine ascendente dei livelli, come abbiamo già visto in `Freq(distribuzione)`:

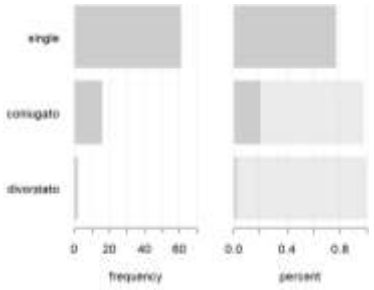
```
Desc(gatti$stato_civile,ord = "asc", main = "stato civile")
-------------------------------------------------------------------------
stato civile

  length      n    NAs unique levels  dupes
      79     79      0      3      3      y
         100.0%   0.0%

        level  freq   perc  cumfreq  cumperc
1  divorziato     2   2.5%        2     2.5%
2   coniugato    16  20.3%       18    22.8%
3      single    61  77.2%       79   100.0%
```

← variabile stato civile, di classe `character`

← ci sono 79 osservazioni (`length`), di cui 79 (100%) non missing (`n`) e nessuna (`0, %`) missing (`NAs`); compaiono tre valori diversi (`unique`), corrispondenti a tre livelli, e nessun duplicato (`dupes`)

← Ci sono 61 single, 16 coniugati e 2 divorziati (frequenze assolute: `freq`), corrispondenti al 77.2, 20.3 e 2.5% dei casi totali (`perc`). Sono anche riportate le frequenze cumulate (`cumfreq`) e le percentuali cumulate (`cumperc`)

Il **grafico** che accompagna l'output è un po' più dimesso del precedente: le barre grigie indicano, rispettivamente, le frequenze e le percentuali dei tre livelli.



Per le variabili `numeric` o `integer` (qui **l'empatia per i gatti**) le cose sono più interessanti. In rosso sono evidenziate le informazioni che studieremo poi (CI della media, asimmetria – skewness e curtosi) o non eccessivamente rilevanti):

```
Desc(gatti$empatia_gatti)
-------------------------------------------------------------------------
gatti$empatia_gatti(- integer)

length       n    NAs  unique     0s   mean  meanCI'
      79      79      0      22      0  13.62   12.25
          100.0%   0.0%           0.0%          14.99

.05   .10   .25  median    .75    .90     .95
3.00  5.60  8.00   15.00  18.50  21.00   22.00

 range   sd   vcoef   mad   IQR  skew    kurt
24.00  6.12   0.45  7.41  10.50 -0.10   -1.04

lowest : 3 (5), 4 (3), 6 (3), 7 (5), 8 (5)
highest: 21 (6), 22 (3), 23, 25, 27

heap(?): remarkable frequency (11.4%) for the
mode(s) (= 16)
' 95%-CI (classic)
```

← variabile empatia, di classe `integer`

← Qui sono riportati i **percentili**: dal 5% al 95%, passando per la mediana.

← La differenza tra minimo e massimo valore (`range`) è =24; la deviazione standard `sd` è 6.12, il range interquartilico (`IQR`) è 10.5. Non ci interessano il coefficiente di variazione (`vcoef`: media/sd) e la deviazione assoluta della mediana (`mad`), altri due indicatori di dispersione. Faremo nel prossimo capitolo l'asimmetria (skewness, `skew`) e la curtosi (`kurt`) della distribuzione.

← Sono indicati i 5 valori più alti e più bassi della distribuzione con rispettive frequenze; può essere utile conoscerli per individuare valori anomali, o decidere di calcolare una media troncata – trimmed

← La moda è una sola: corrisponde al punteggio 16, ottenuto dall'11.4% dei casi

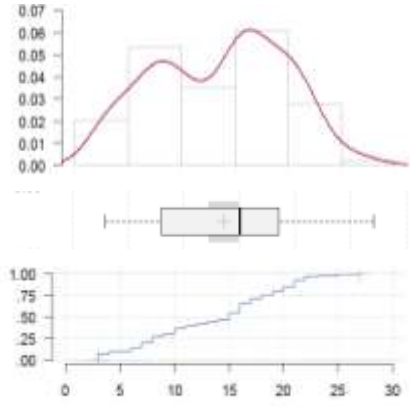
← Il CI della media (seconda riga dell'output) è stato calcolato con il metodo "classico", che impareremo nel capitolo 6.

I **grafici** sono addirittura tre, nella stessa finestra.

In altro, troviamo **l'istogramma delle densità di frequenza** cui è sovrapposta la **funzione densità di probabilità**: parleremo di questo istogramma nel capitolo 4, §4.2, e delle probabilità nel capitolo 5.

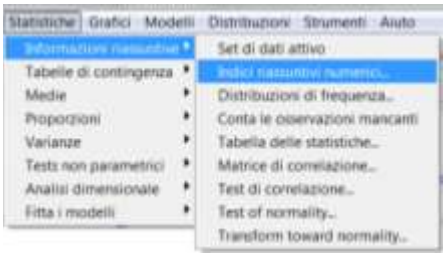
In mezzo, abbiamo il **boxplot** della distribuzione: lo affronteremo nel capitolo 4, §4.3.

In basso, troviamo il grafico delle frequenze relative cumulate.

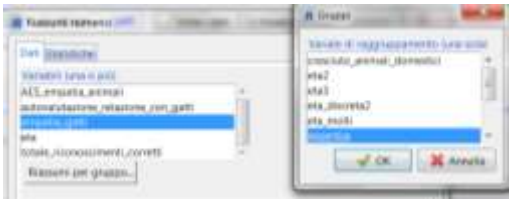


Vedremo altre applicazioni di **`Desc`**, su altre classi di oggetti, nei capitoli successivi.

In `RCommander`, parte delle statistiche descrittive univariate che abbiamo visto si trovano in `Statistiche` → `Informazioni riassuntive` → **`Indici riassuntivi numerici.`**



Nella scheda `Dati` si scelgono le misure da descrivere, eventualmente anche distinguendole in base ai livelli di un fattore (`Riassumi per gruppo`); nella scheda `Statistiche` si selezionano le statistiche descrittive: ne conosciamo solo alcune, per ora, e ne mancano un bel po'.



Per avere una descrizione in base alla combinazione dei livelli di più fattori, bisogna usare `Statistiche` → `Informazioni riassuntive` → **`Tabella delle modifiche`** : le statistiche sono davvero poche, ma si può indicare una statistica non presente nell'elenco scrivendone il nome della funzione nello slot `Altro`.

`Statistiche` → `Informazioni riassuntive` → **`Conta le osservazioni mancanti`** conta i dati NA in tutte le variabili del dataframe.

Capitolo 4
Grafici per distribuzioni univariate

"There is no statistical tool that is as powerful as a well-chosen graph"
Chambers, Cleveland, Kleiner & Tukey (1983)

"The greatest value of a picture is when it forces us to notice what we never expected to see"
Tukey[16](Exploratory data analysis, 1977)

"Excellence in statistical graphics consists of complex ideas communicated with clarity, precision and efficiency"
Tofte, 1983
"Humans are good at discerning subtle patterns that are really there, but equally so at imagining them when they are altogether absent"
Sagan, 1985

In questo capitolo abbandoniamo il dataframe ***gatti****, che recupereremo per scoprire se vivere con un gatto facilita il riconoscimento dei miagolii o se l'empatia influisce su una buona relazione con il micio (dal punto di vista dell'umano), e ci dedichiamo anima e corpo al più serio dataframe* ***attaccamento****: scaricatelo da Elly, insieme alla descrizione delle variabili che lo compongono. Leggete la descrizione, aprite il dataframe in R, e, prima di procedere oltre, fate il seguente esercizio:*

1. *Descrivete la struttura del dataframe*
2. *Descrivete il campione: da quanti soggetti è composto? Quanti hanno l'assistito in casa e quanti invece in RSA? Come sono distribuite le caratteristiche socio-anagrafiche (tranne l'età) nel campione complessivo?*
3. *Le sottoscale del CBI, insieme, possono dare origine a un punteggio totale: createlo (chiamate la variabile* `$CBI_totale`*) e fate in modo che la nuova variabile sia salvata nel file.*
4. *Anche le sottoscale del WHOQOL possono creare una dimensione complessiva: in questo caso, è data dalla media della qualità della vita nei diversi ambiti. Create la variabile (chiamatela* `$WHOQOL_media`*) e fate in modo che sia salvata nel file.*
6. *Considerate solo il sottogruppo con l'assistito in casa: quanti usufruiscono di un centro diurno? Cosa potete rilevare rispetto all'aiuto ricevuto? Quali considerazioni si potrebbero fare (e come) rispetto all'avere un aiuto e al burden totale?*

Nel capitolo precedente abbiamo usato modelli numerici per descrivere caratteristiche di **distribuzioni univariate**; in questo capitolo, useremo per lo stesso scopo dei **grafici**, per cui R è particolarmente preparato; quando affronteremo la **descrizione di distribuzioni bivariate** (due distribuzioni congiunte: test chi quadrato, correlazioni, regressioni semplici, ecc.) o **multivariate** (correlazioni parziali, regressioni multiple, ecc.), vedremo i grafici dedicati questo tipo di distribuzioni più complesse.

I grafici (Tufte,1983) "mostrano visivamente quantità misurate per mezzo dell'uso combinato di punti, linee, un sistema di coordinate, numeri, simboli, ombreggiature e colore". **L'ispezione visiva** della presentazione dei dati è **assolutamente indispensabile**, anche utilizzando più di un tipo di grafico, se necessario, sulla stessa distribuzione: il suo obiettivo primario è **comunicare**, a se stessi e agli altri, cosa è successo nella ricerca.

Secondo Schmid (1954), **alcuni** dei vantaggi dei metodi grafici sono:

[16] Oltre ad essere stato un grande statistico (lo ritroveremo nei test post hoc dell'analisi della varianza) un pioniere dell'analisi visiva dei dati, a Tukey è attribuita la creazione dei termini "software" (1958) e "bit" (1947).

1. rispetto ad altri tipi di presentazione, grafici ben fatti sono più efficaci nel creare interesse e attrarre l'attenzione del lettore o dell'uditore;
2. le relazioni visive rappresentate dai grafici sono comprese e ricordate più facilmente;
3. l'uso dei grafici fa risparmiare tempo, dato che il significato essenziale di ampie raccolte di dati può essere compreso con uno sguardo (beh, almeno molto spesso, anche se non sempre);
4. i grafici e diagrammi offrono una raffigurazione completa ed esauriente di un problema, che ne rende la comprensione più completa e più equilibrata rispetto a quella che potrebbe derivare da presentazioni tabulari o testuali dei dati;
5. i grafici aiutano a far emergere realtà nascoste e relazioni, stimolano e aiutano il pensiero analitico e l'investigazione.

Tra i packages di base scaricati al momento dell'installazione di R, `graphics`, `lattice` e `MASS` contengono molte funzioni utili per quasi tutti i grafici che ci potranno servire. Altri packages, però, sono dedicati a funzioni grafiche decisamente utili e raffinate (abbiamo già visto i grafici prodotti con `Desc(distribuzione)`): il top è probabilmente il package `ggplot2` – che, disgraziatamente, non è affatto intuitivo. RCommander è piuttosto utile per creare grafici efficaci con poca fatica: lo vedremo, quindi, all'opera.

Le funzioni che R utilizza per produrre grafici sono di tre tipi:

1. **funzioni di alto livello**: creano un nuovo grafico nella finestra Grafici: `plot`, `boxplot`, `histogram`, `pie`, `barplot`. La funzione di alto livello più generale è `plot`, adattabile a diverse scale di misura;
2. funzioni di **basso livello**: aggiungono parti a un grafico esistente: per esempio, `abline` sovrappone una linea a un grafico, secondo le coordinate indicate negli argomenti della funzione, mentre `text` inserisce stringhe di testo nel plot;
3. funzioni per **grafici interattivi**: permettono di aggiungere o di ricavare interattivamente informazioni da un grafico esistente; per esempio, la funzione `identify` consente di identificare i simboli nel grafico attribuendo loro i numeri di riga dei soggetti nel dataframe, cliccando con il mouse sulle loro coordinate. La funzione si applica a un grafico creato da `plot` per una sola distribuzione o due distribuzioni (in questo caso, abbiamo uno ***scatterplot***, o grafico a dispersione, che useremo nella correlazione e nella regressione semplice).

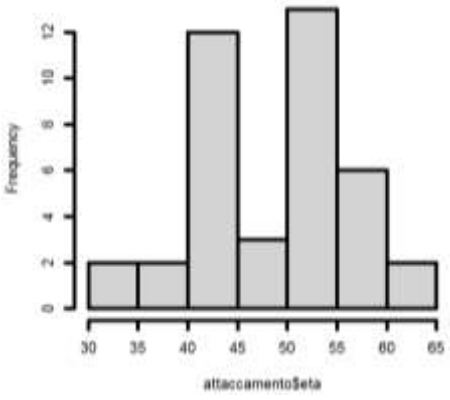
Infine, può essere impostata un'imponente quantità di parametri grafici: per l'elenco completo, chiedete l'`help` della funzione `par` (parametri); nella tabella seguente, sono elencati solo quelli di cui faremo un uso più frequente.

Scopo	Parametro	Opzioni	Risultato
Tipo di simbolo	`pch`	`= numero intero da 0 a 25`	Cambia il tipo di simbolo usato nel grafico; l'elenco è fornito nel §4.1
Grandezza del simbolo	`cex`	`=numero positivo ≥1`	I simboli del grafico sono ingranditi o rimpiccioliti
Tipo di linea	`lty`	`=1`	continua (default)
		`=2`	tratteggiata
		`=3`	punteggiata
		`=4`	punti e tratti
		`=5`	tratto lungo
		`=6`	punti e tratti
Spessore della linea	`lwd`	`=numero positivo ≥1`	Lo spessore della linea di default è =1; non tutti i sistemi operativi lavorano con numeri inferiori a 1
Colore degli elementi	`col`	`="name"`	di default è "black"; per ottenere la lista dei colori (moltissimi!), usate la funzione `colors()`
		`= "RRGGBB"`	Indicare le tre coppie di cifre esadecimali per la componente rosso, verde e blu del colore desiderato
	`col.axis, col.lab, col.main`		Con le stesse modalità di `col()`, per definire il colore degli assi, delle etichette, del titolo; di default è "black"

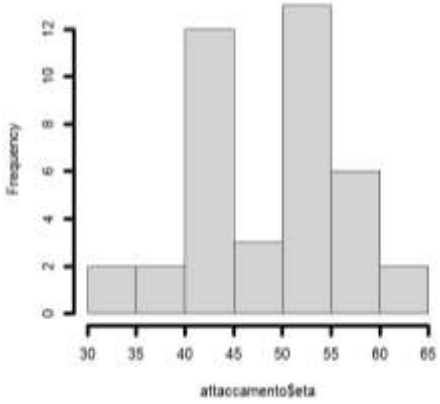
Scopo	Parametro	Opzioni	Risultato
Cambiare i limiti degli assi	`xlim`	`=c(minimo, massimo)`	Definisce il limite minimo e il limite massimo dell'asse X
	`ylim`	`=c(minimo, massimo)`	Definisce il limite minimo e il limite massimo dell'asse Y
Cambiare le etichette degli assi	`xlab`	`="nome"`	Cambia l'etichetta dell'asse X
	`ylab`	`="nome"`	Cambia l'etichetta dell'asse Y
Orientamento delle etichette degli assi	`las`	=0	Parallele agli assi (default)
		=1	Orizzontali
		=2	Perpendicolari agli assi
		=3	Verticali

I parametri grafici possono essere impostati come argomenti della funzione `par` **prima** di lanciare il comando del grafico, ma diversi tra essi possono essere inseriti **come argomenti** della funzione che crea il grafico stesso, o digitati **dopo** aver creato il grafico. In molti casi il loro effetto sul grafico è lo stesso, in altri no: per esempio, `par(lwd>1)`, che gestisce lo spessore delle linee (**`line width`**), aumenta lo spessore di **tutte** le linee che costituiscono **l'istogramma** creato **dopo** aver impostato `par`, con la funzione `hist(distribuzione numerica)`; invece, scrivendo `lwd>1` (ad esempio, `lwd=4`) come argomento della funzione `hist`, viene aumentato solo lo spessore degli **assi** del grafico.

```
par(lwd=4)
hist(attaccamento$eta)
```




```
hist(attaccamento$eta, lwd=4)
```



`plot` è la funzione più generale per creare un grafico, ed è **sensibile alla classe della variabile o delle variabili cui è applicata**: se è di classe `numeric`, crea un **grafico a dispersione** per una distribuzione bivariata o un grafico sequenziale per una sola distribuzione; se è una sola variabile di classe `factor`, crea un grafico a barre (*barplot*) che mostra le frequenze di ogni classe; se sono due variabili di classe `factor`, crea un grafico a mosaico (*mosaic plot*). Li vediamo uno per uno.

4.1 Plot per variabili numeriche o scatterplot

Il nome del dataframe `attaccamento` è lungo: per alleggerire la lettura delle funzioni dei prossimi grafici, assegniamo a un nuovo dataframe **a** le caratteristiche di `attaccamento` e usiamo questo nuovo oggetto:

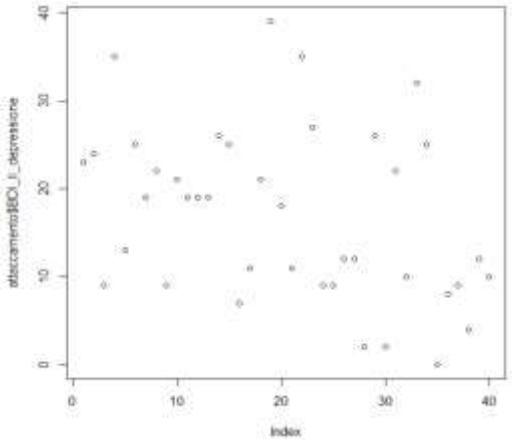
```
a<-attaccamento
```

Quando la variabile da rappresentare è di classe `numeric`, `plot` crea un **grafico sequenziale**, in cui per ogni soggetto sono rappresentate le coordinate della distribuzione in X e della distribuzione in Y. Nel caso di cui ci occupiamo qui, con **una sola distribuzione**, in ascissa (X - Index) sono elencati i casi secondo il numero di riga nel dataframe, in ordinata (Y) i valori della variabile prescelta, per ciascun soggetto.

Quindi, se vogliamo conoscere la **distribuzione dei punteggi di depressione**(`$BDI_II_depressione`) **soggetto per soggetto**, scriveremo:

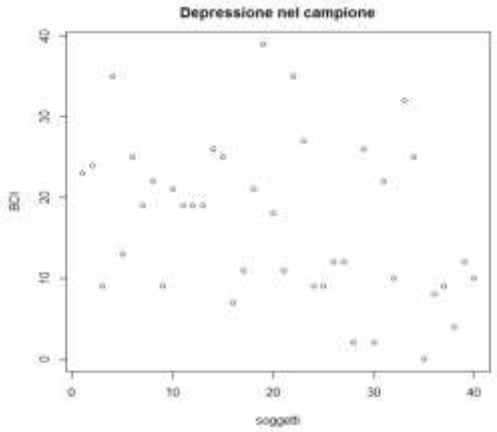
```
plot(a$BDI_II_depressione)
```

In ascissa abbiamo i 40 soggetti, secondo l'ordine delle righe nel dataframe, in ordinata i punteggi al test Beck Depression Inventory: all'interno del grafico, ogni pallino rappresenta il punteggio al test di un soggetto. Di default, X è stato etichettato Index, Y ha ricevuto il nome della variabile.

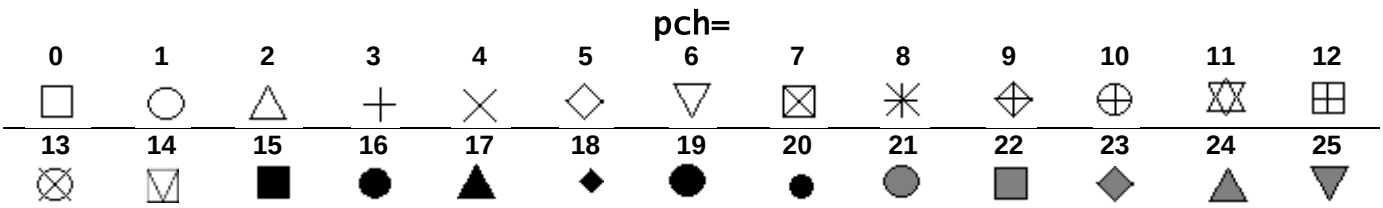


Possiamo vivacizzare un po' l'aspetto del grafico, aggiungendo argomenti opzionali a `plot`; cominciamo personalizzando i **titoli degli assi** con `xlab= "testo"` e `ylab= "testo"` per X e Y, e diamo un **titolo all'intero grafico** con l'argomento `main= "titolo del grafico"`:

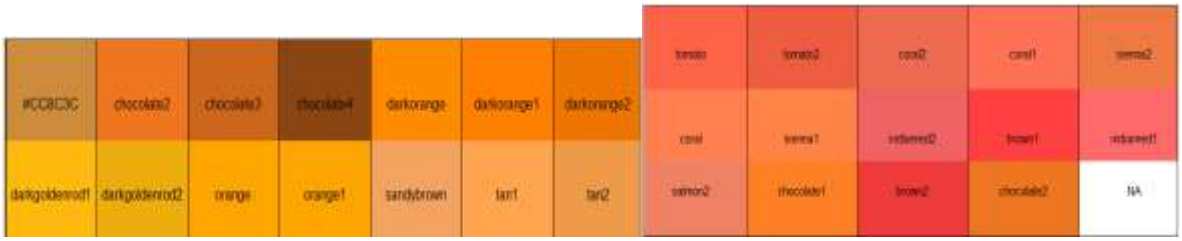
```
plot(a$BDI_II_depressione,  xlab="soggetti",  ylab="BDI",
   main="Depressione nel campione")
```



I simboli all'interno del grafico possono essere modificati scegliendo una diversa configurazione, una diversa grandezza e un diverso colore. `pch=numero` definisce il tipo di simbolo (`point character`: v. sotto la lista dei simboli disponibili), `cex=numero` definisce la grandezza del simbolo (`character expansion`: da 1, il setting di default, in su) , `col="Errore. Il segnalibro non è definito.colore"` indica il suo colore.

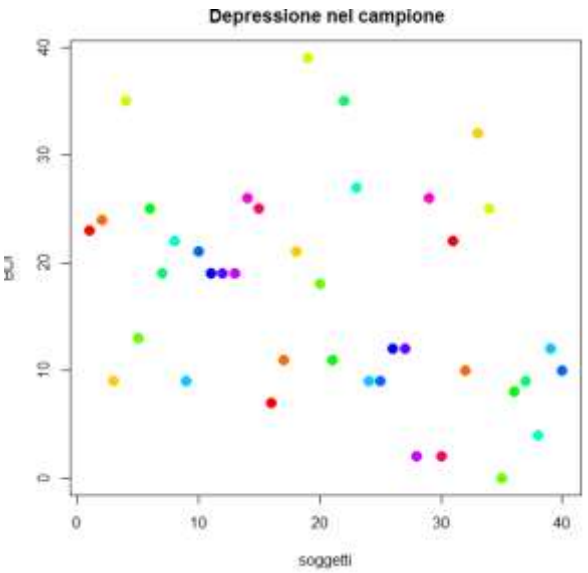


I colori possono essere indicati per nome ("red", "light blue", "purple", ecc.) o per codice esadecimale RGB (Red-Green-Blue): digitate `demo(colors)` per avere una dimostrazione di tutti i colori disponibili; per una coloritura rapida, si può usare l'argomento `col=rainbow(numero di sfumature)`. Sotto, sono raffigurate alcune delle *palette* mostrate dalla funzione `demo(colors)`:



Aggiungiamo colore al plot, ingrandiamone i simboli e cambiamo il simbolo da cerchio vuoto a cerchio pieno:

```
plot(a$BDI_II_depressione, xlab = "soggetti", ylab="BDI", main="Depressione nel campione",cex=1.5, pch=19, col=rainbow(15))
```

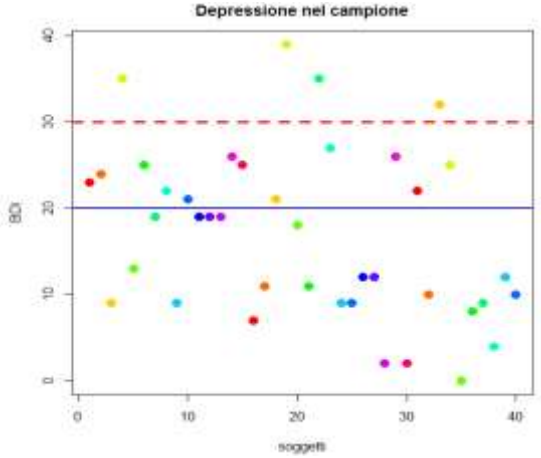


I punteggi del BDI-II compresi tra 20 e 29 indicano una depressione moderata, quelli uguali o superiori a 30 una grave depressione: per evidenziare questi cut off nel grafico possiamo **aggiungere due linee** con `abline`: una linea blu continua per la depressione moderata, una linea rossa tratteggiata per la depressione grave. In `abline` devono essere specificate le coordinate: `h=valore in Y` serve per definire la coordinata in Y per linee orizzontali; `v= valore in X` definisce la coordinata in X per linee verticali.

L'argomento `lty=valore` determina il tipo di linea `[line type]`: 1 (di default) = linea continua, 2= tratteggiata, 3= punteggiata, 4= tratti e punti, 5= tratti lunghi, 6= tratti doppi. Infine, abbiamo già incontrato `lwd= valore` per lo spessore della linea (=1 è lo spessore di default).

Nel primo comando tracciamo la **linea continua blu** per un punteggio BDI II pari a **20**, con spessore = 2.5; nel **secondo** comando tracciamo la **linea rossa tratteggiata**, con spessore = 3, per un punteggio BDI pari a **30**:

```
abline(h = 20, col="blue",lty=1, lwd=2.5)
abline(h=30, col="red", lty=2, lwd=3)
```

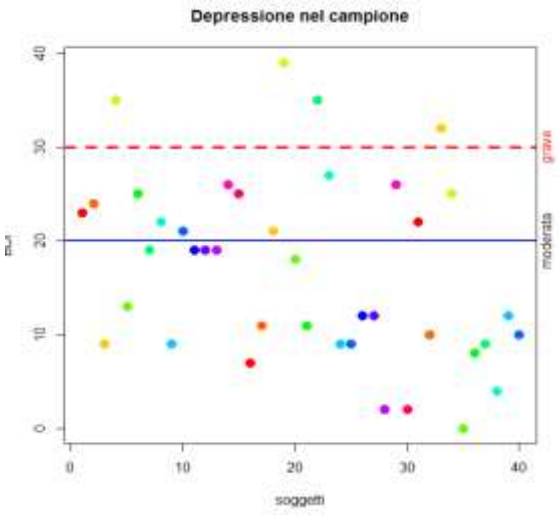


Potete anche digitare una sola funzione `abline`, usando `c` nei suoi argomenti per dare istruzioni su tutte le linee che intende tracciare:

```
abline(h=c(20, 30), col=c("blue", "red"), lty=c(1,2),
   lwd=c(2.5, 3))
```

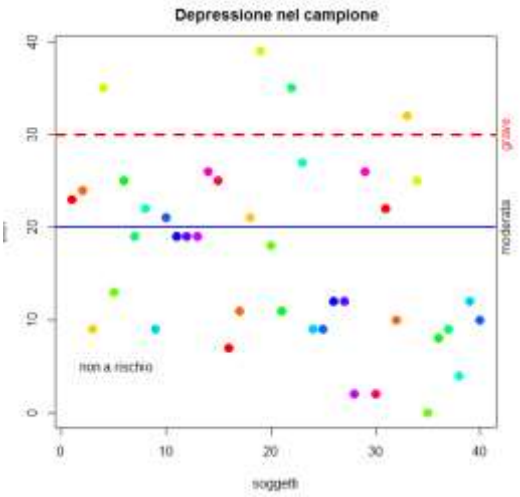
Si può **aggiungere del testo** al grafico; dato che il nostro è piuttosto pieno, **aggiungiamo all'esterno**, nei **margini**, del testo che ricordi il significato delle linee tracciate. Usiamo **`mtext`** `[margin text]`, in cui indicheremo: `text= "cose da scrivere"`; `side= valore` per indicare in quale margine scrivere (1-basso, 2-sinistra, 3-alto, 4-destra); `atErrore. Il segnalibro non è definito.= valore` per indicare il valore della coordinata in cui inserire il testo. Si può anche determinare il colore del testo (`col=`), il font, ecc. Noi scriveremo "moderata" in corrispondenza del margine destro, altezza in $Y = 20$, e "grave" in corrispondenza del margine destro, altezza in $Y = 30$, questa volta in rosso:

```
mtext(text = "moderata",side = 4,at= 20)
mtext(text = "grave", side=4, at=30, col = "red")
```



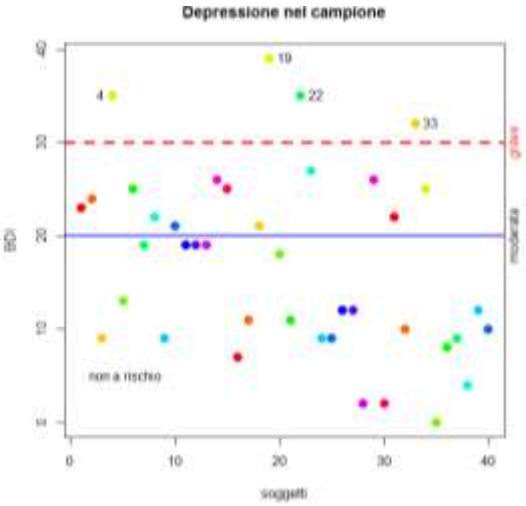
Per **aggiungere testo all'interno del grafico**, si usa `text`, i cui argomenti sono: un **punto di partenza** da cui scrivere il testo, determinato da una coordinata in X (`x= valore`) e una coordinata in Y (`y= valore`); il **testo** da scrivere: `labels= "testo"`; la **posizione** del testo rispetto alle coordinate: `pos= valore`: 1-sotto, 2-a sinistra, 3-sopra, 4-a destra. Si possono anche specificare colore e font, naturalmente. Noi scriveremo "non a rischio" nel settore del grafico inferiore alla linea blu; lo spazio libero è in fondo a sinistra, quindi indicheremo come coordinate, da cui tracciare verso destra il testo, il valore 1 in X e 5 in Y:

```
text(x= 1, y=5, labels = "non a rischio", pos = 4)
```



L' uso di `c` visto in `abline` si adatta anche a `mtext` e `text`, per scrivere contemporaneamente più cose nel plot.

Non solo: possiamo aggiungere un elemento interattivo usando la funzione `identify(variabile in X)` che ci consente di identificare chi siano i soggetti con grave depressione. Dopo aver digitato il comando e premuto Invio, R aspetta che clicchiamo con il mouse su ogni punto del grafico (`Locator active: Esc to finish`): noi cliccheremo sui quattro punti sopra la linea di riferimento della depressione grave. A fianco di ciascun punto viene stampato il numero di riga che il caso occupa nel dataframe. Terminata l'identificazione, si schiaccia il tasto Esc sulla tastiera per tornare al prompt dei comandi: in Console saranno stampati i numeri di riga dei soggetti selezionati sul grafico.



Riassumendo, lo script per ottenere quest'ultimo grafico è:

```
plot(a$BDI_II_depressione, xlab = "soggetti", ylab="BDI", main="Depressione nel campione",cex=1.5, pch=19,
col=rainbow(15))
abline(h = 20, col="blue",lty=1, lwd=2.5); abline(h=30, col="red", lty=2, lwd=3)
mtext(text = "moderata",side = 4,at= 20); mtext(text = "grave", side=4, at=30, col = "red")
text(x= 1, y=5, labels = "non a rischio", pos=4)
identify(a$BDI_II_depressione)
```

4.2 Grafici per distribuzioni di densità di frequenza e di frequenza: `hist, plot, barplot`

Nel capitolo precedente abbiamo descritto le distribuzioni di frequenza, per variabili continue e discrete, usando `table`; in questo paragrafo le descriveremo usando tre diversi tipi di grafici.

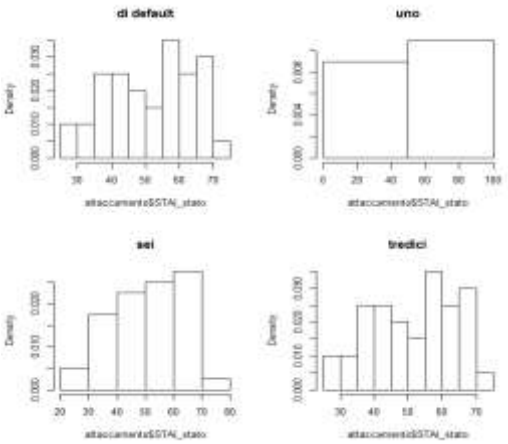
Per rappresentare la distribuzione della **densità di frequenza** assoluta (che, ricordiamo, è il **rapporto tra la frequenza di ogni classe e la sua ampiezza**) di variabili continue suddivise in classi (`bin`), si possono usare gli **istogrammi**. Un istogramma è in genere usato per contare frequenze e per mostrare la distribuzione di una variabile, anche se secondo i suoi detrattori (ad esempio, Wilkinson, 1992), "non serve né all'uno né all'altro", soprattutto a causa della sua dipendenza dalla scelta (piuttosto arbitraria) del numero di classi da rappresentare.

Per fare un istogramma, in R si usa **`hist(distribuzione continua)`**: in X è rappresentata la variabile misurata, suddivisa in classi, e in Y la sua **densità di frequenza**. È possibile indicare in ordinata la **frequenza assoluta** della classe o la sua **frequenza relativa**: per avere la densità, useremo l'argomento **`freq=FALSE;`** per avere la frequenza assoluta non scriveremo **nulla** (è l'opzione di default); per avere la frequenza relativa indichiamo l'argomento `prob=TRUE`. Se non specifichiamo nulla sul numero di classi da costituire (**`breaks=`**), R le stima da sé usando la regola di Sturges (di default: `breaks="Sturges"`)[17]: per quanto ampiamente diffuso, tuttavia, questo metodo non funziona in modo ottimale soprattutto se le osservazioni sono poche ($N < 30$) e la distribuzione è asimmetrica. È quindi una saggia regola provare a costruire più istogrammi con diverse classi, per valutare quale dia la rappresentazione migliore, modificando l'argomento `breaks=`: notate che R **accetta solo suggerimenti**, su questo punto, riservandosi di non accogliere un numero di classi evidentemente inadeguato.

Usiamo `hist` per rappresentare la distribuzione di densità di frequenza dell'ansia di stato (`$STAI_stato`) del nostro campione di caregiver; proviamo a impostare diversi numeri di classi: quello di default , $K = 1$, $K = 6$ e $K = 13$.

Dato che dobbiamo visualizzare ben quattro grafici, usiamo uno dei parametri `par` citati all'inizio di questo capitolo: **`mfrow(c(righe, colonne)`**. `mfrow` (sta per MultiFrame Row Wise) ripartisce la finestra dei grafici secondo il numero di righe e colonne specificato nei suoi argomenti; dato che predispone l'ambiente su cui saranno stampati i grafici, deve essere impostato **prima** di eseguire i grafici stessi.

```
par(mfrow = c(2,2))
hist(a$STAI_stato,freq=FALSE, main="di default")
hist(a$STAI_stato,freq=FALSE, main="uno", breaks = 1)
hist(a$STAI_stato,freq=FALSE, main="sei", breaks = 6)
hist(a$STAI_stato,freq=FALSE, main="tredici", breaks =13)
```

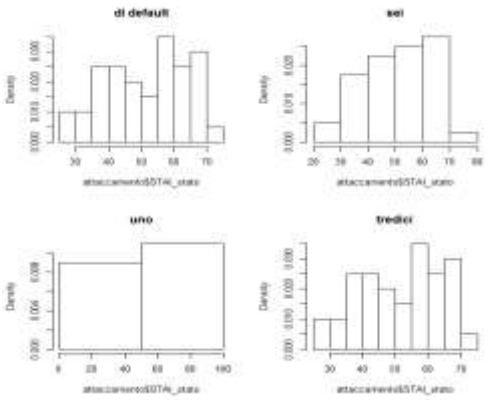


La scala in ascissa permette di ricavare i limiti delle classi calcolate da R: da 25 a 30 sono stati registrati due casi, altri due tra 30 e 35, cinque casi tra 35 e 40, et cetera. La distribuzione è unimodale (la classe modale è $55 - 40$, con 7 casi). Sembra esserci una certa **prevalenza di casi tra i punteggi più alti**: l'ansia di stato sembra un potenziale problema per i caregiver.

Notate che R accoglie solo il suggerimento "sei classi", mentre rifiuta, evidentemente perché troppo lontani dal numero ideale di classi, sia una che tredici classi. Una regola pratica per identificare il numero ideale di classi è arrotondare all'intero più vicino la radice quadrata del numero di osservazioni: nel nostro caso, $\sqrt{40 = 6.32}$, quindi "sei classi" è davvero un buon suggerimento.

[17] Numero di classi $K=1+3.322*\log_{10}(N)$

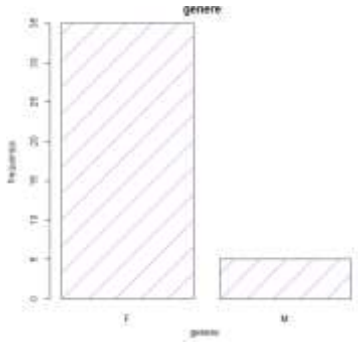
Notate anche che **par(mfcol=numero righe, numero colonne)** avrebbe predisposto il layout della finestra dei grafici ordinandoli per colonna:



Attenzione: fino alla fine della sessione di lavoro, la finestra rimane ripartita secondo le indicazioni di `par(mfrow)` o `par(mfcol)`: per tornare a visualizzare un grafico per finestra, richiedetelo espressamente:

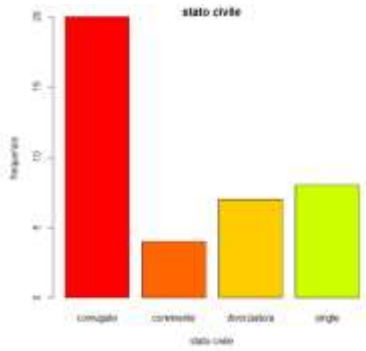
```
par(mfrow = c(1,1))
```

Per rappresentare la distribuzione della **frequenza** assoluta di una variabile continua **non suddivisa in** classi (o meglio, in classi di ampiezza unitaria), si può usare `barplot(variabile)`, ma è piuttosto difficile pensare che il grafico prodotto possa essere di qualche utilità. Molto più efficace è **barplot**`Errore. Il segnalibro non è definito.(table(variabile))`, per rappresentare la distribuzione di frequenza di variabili categoriali.
Vediamo la distribuzione del genere e dello stato civile dei caregiver. Nel primo caso, facciamo conoscenza con l'argomento grafico **density=valore**, che riempie l'area dei rettangoli con linee diagonali più o meno fitte (da 1 in su); le linee possono essere colorate con l'argomento `col="colore"` già visto.




```
barplot(table(a$genere), main="genere",
xlab="genere", ylab="frequenza", col=
"purple", density= 3)
```

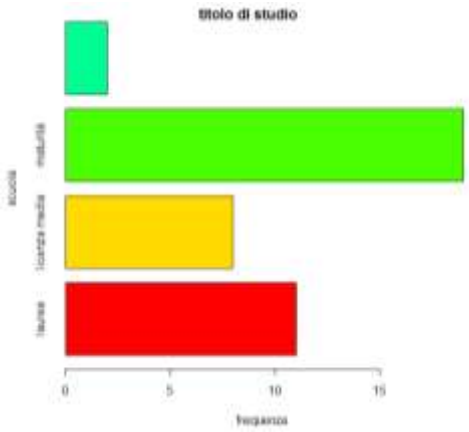
```
barplot(table(a$stato_civile), main=
"statocivile", xlab= "stato civile",
ylab= "frequenza", col= rainbow(15))
```



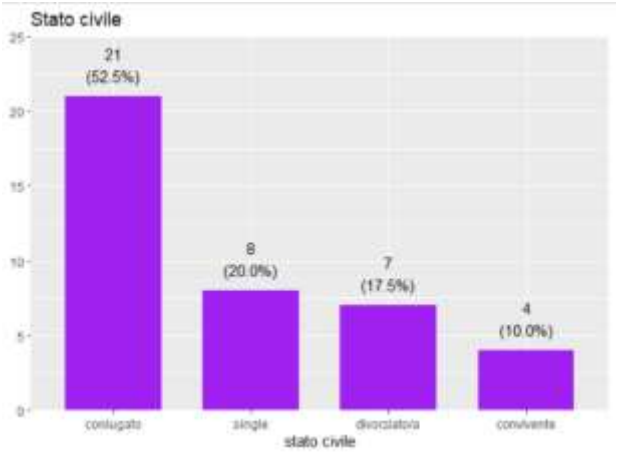
Il ruolo del caregiver non sembra certo indipendente dal genere, e la distribuzione dello stato civile vede decisamente prevalere la moda "coniugato". Notate che le barre sono **separate**, dato che rappresentano categorie discrete, e non contigue come nei grafici precedenti (di default, l'argomento è **beside= FALSE**).

Le barre possono anche essere disposte orizzontalmente: in questo caso, il grafico si definisce **grafico a pila**; basta impostare l'argomento logico **horiz= TRUE** e stare attenti alle etichette di X e Y. Applichiamolo alla distribuzione del titolo di studio dei caregiver:

```
barplot(table(a$titolo_studio),horiz = TRUE, main="titolo di
studio",xlab = "frequenza",ylab = "scuola", col=rainbow(7))
```



La funzione `plot_frq(variabile, type="bar")` di `sjPlot` produce un barplot dall'aspetto curato e consente di ordinare automaticamente le barre a seconda della frequenza, in senso ascendente o discendente. Gli altri argomenti più importanti sono: `sort.frq = "desc"` oppure `"asc"` per ordinare le frequenze, `axis.title=` "nome dell'asse X", `title=` "titolo del grafico", `geom.colors=` "colore".

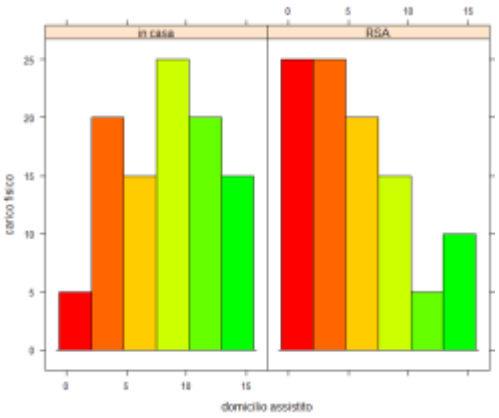



```
plot_frq(a$stato_civile,title = "Stato civile",
   sort.frq = "desc", type = "bar", axis.title =
   "stato civile", geom.colors = "purple")
```

Il package **lattice**, che viene installato insieme a R (ma va comunque caricato quando s'intende usarlo) contiene **`histogram(~variabile|fattore)`**[18], che è comoda se si è interessati a **confrontare le diverse distribuzioni di frequenza nei livelli di un fattore**.

Per esempio, se siamo interessati alla distribuzione di frequenza dello stress derivante dalla fatica fisica nei caregiver che assistono il paziente a casa rispetto a quella dei caregiver con pazienti ricoverato, scriveremo:

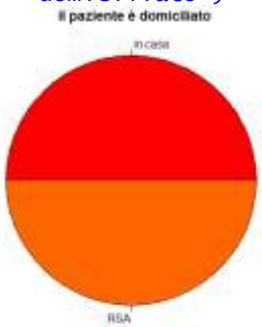
```
histogram(~a$CBI_burden_fisico|a$domicilio_assistito,
   xlab="domicilio assistito", ylab="carico fisico",
   col=rainbow(15))
```



È facile osservare che i caregiver con paziente in RSA mostrano prevalentemente bassi punteggi di burden fisico (asimmetria positiva), mentre tra chi assiste il paziente in casa ci sono pochissimi punteggi bassi e una prevalenza di punteggi alti.

Un'altra modalità grafica per presentare distribuzioni nominali è il grafico a torta o ***pie***, in R `pie(table(variabile))`, in cui le dimensioni dei settori del cerchio sono proporzionali alle frequenze che rappresentano (l'area del cerchio corrisponde al totale dei casi). I settori rappresentano in senso orario (`clockwise=TRUE`, di default) le etichette nel loro ordinamento alfanumerico

```
pie(table(a$domicilio_assistito),
col=rainbow(15), main="il paziente è
domiciliato")
```




```
pie(table(a$occupazione_attuale),
col=rainbow(15), main="professione del
caregiver")
```

Esattamente metà del campione ha ricoverato per un periodo di sollievo il proprio assistito in una RSA, mentre l'altra metà lo assiste in casa propria. La distribuzione dell'occupazione del caregiver, in particolare la moda della distribuzione, vi suscita qualche riflessione?

[18] Sulla ***tilde*** "~" si veda a pag. 82

Per quanto visivamente immediati, i grafici a torta sono stati pesantemente criticati: “le tabelle di contingenza sono preferibili per dati poco numerosi. Una tabella è sempre preferibile a uno stupido grafico a torta; l’unica presentazione grafica peggiore di un grafico a torta è una lunga serie di grafici a torta... non dovrebbero mai essere usati” (Tufte, 1983); “il grafico a torta è completamente inutile” (Bertin, 1981); “i grafici a torta sono i meno utili tra tutte le forme grafiche” (Wainer, 1977). Insomma, il solo fatto che sia possibile crearli facilmente non ne raccomanda l’uso.

4.3 Grafici per dati ordinali e intervallari

Naturalmente, le distribuzioni di densità di frequenza e di frequenza assoluta o relativa si usano per rappresentare anche variabili ordinali e intervallari, che in più hanno varie opzioni per rappresentare graficamente gli indici di tendenza centrale, di dispersione e posizione.

Uno dei grafici che più useremo è il **grafico a scatola e baffi** (*box and whisker*), o **boxplot**, come lo chiameremo d’ora in poi (Tukey, 1977). Nel boxplot sono rappresentati:

- il **range interquartilico - IR**: il primo quartile traccia il bordo inferiore della scatola, il terzo quartile traccia il bordo superiore della scatola. L’area della scatola è quindi proporzionale al range interquartilico: distribuzioni compatte attorno alla mediana produrranno scatole strette, distribuzioni con molti valori dispersi produrranno scatole più larghe. Il range interquartilico è quindi un indicatore efficace della dispersione della distribuzione, con il vantaggio di non risentire della presenza di valori estremi che, invece, incidono su misure di dispersione quali varianza e deviazione standard. Come avevamo anticipato parlando della funzione `fivenum`, i bordi inferiore e superiori sono ***fourths*** (così li definisce Tukey), non esattamente quantili (***quantiles***): la mediana della prima metà della distribuzione ordinata e la mediana della seconda metà della distribuzione ordinata. Ma la differenza nella pratica è trascurabile.
- la **mediana**, indicata dalla linea spessa interna alla scatola.
- **due** “baffi”, o meglio **whiskers**Errore. Il segnalibro non è definito.: il whisker **superiore** (se preferite, Valore Adiacente Superiore) è calcolato **aggiungendo** al **terzo quartile** una **volta e mezzo il range interquartilico**; il whisker **inferiore** (Valore Adiacente Inferiore) è calcolato **sottraendo al primo quartile** una **volta e mezzo il range interquartilico**:

$$whisker_{inferiore} = Q1 - 1.5(Q3 - Q1); \qquad whisker_{superiore} = Q3 + 1.5(Q3 - Q1)$$

In R, l’estensione del “baffo” dalla scatola è regolata dall’argomento `range= valore`; di default è predisposta appunto a 1.5. Come abbastanza consueto, questo valore è stato proposto da Tukey come scelta di buon senso: pare che a un suo allievo, che gli aveva chiesto la motivazione del valore 1.5, abbia risposto “***Because 1 would be too small and 2 would be too large***” (De Veaux, Velleman e Book, 2008). Nulla, perciò, se non la convenzione, vieta di modificare l’estensione dei baffi.

Se i valori dei whiskers così calcolati **eccedono** **il valore minimo e/o il valore massimo della distribuzione**, nel **grafico sono fissati al valore minimo e/o al valore massimo**. In caso contrario, nel grafico possono essere rappresentati i casi che rappresentano **potenziali outlier (outside values)**, cioè casi i cui valori sono maggiori del wisker superiore o minori del whisker inferiore: sono indicati da **cerchietti** sopra o sotto i whiskers. È **importante evidenziare gli outlier[19]/outside values**, sia da un punto di vista statistico che interpretativo. Dal punto di vista

[19] Gli outlier univariati e multivariati di cui parleremo più avanti sono identificati dai loro scarti dalla media (due o più deviazioni standard dalla media), mentre il riferimento degli outside values è il range interquartilico. Nessuno si scandalizzerà, comunque, se useremo il termine più diffuso “outlier” per identificare valori come “molto, molto anomali”

statistico, la loro presenza può danneggiare il fit di un modello: è abbastanza facile intuire che per questi casi gli errori del modello (gli scarti dalla media, per esempio) saranno gravi, e che la loro eliminazione dal dataframe potrebbe ridurre gli errori e produrre modelli migliori. D'altronde, ci sono regole piuttosto rigide per eliminare gli outlier da un modello: ne parleremo diffusamente nel capitolo 9. Dal punto di vista interpretativo, i casi anomali possono rappresentare errori di campionamento (potrebbero essere soggetti che appartengono a una popolazione diversa dagli altri, ad esempio soggetti con disturbi emotivi non diagnosticati reclutati in un campione normativo), o i casi all'estremo di una coda della popolazione che ci interessa; potrebbero essere soggetti particolarmente dotati, o al contrario particolarmente a rischio, da segnalare al committente della ricerca. Attenzione a **non farvi ingannare dal grafico**: se ci sono **più outlier con lo stesso punteggio**, nel grafico vedremo **un** solo cerchietto: prima di ulteriori indagini, potremo quindi solo dire non che esiste **un** outlier nel campione, ma che esiste **almeno un** outlier.

Oltre a dare informazioni sulla più o meno ampia variabilità attorno alla mediana della distribuzione, scatola e "baffi" consentono di interpretare la distribuzione rispetto alla normale teorica: in distribuzioni ragionevolmente affini alla normale, le distanze tra ogni quartile e la mediana sono uguali (la mediana è esattamente a metà del box, cioè la distribuzione è **simmetrica**), e i "baffi" hanno uguale lunghezza.

Vediamone due esempi di boxplot, uno senza e uno con outlier. Rappresentiamo prima il burden (ovvero il carico assistenziale) per il caregiver **derivante dall'avere poco tempo per** sé, sottratto dagli impegni dell'assistenza: è la variabile `$CBI_burden_restrizione_tempo`. Vediamo con `summary` gli elementi che compariranno nel boxplot: primo e terzo quartile, mediana, minimo e massimo:

```
summary(a$CBI_burden_restrizione_tempo)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0     8.0    15.0    12.9    18.0    20.0
```

Potremmo anche usare la funzione **`fivenum`**, già esplorata a pag. 63, che riporta i cinque descrittori proposti da Tukey: però l'output non riporta le loro etichette; quindi, dobbiamo affidarci alla memoria per ricordare chi è cosa; scegliete liberamente la funzione che preferite tra `fivenum` e `summary`:

```
fivenum(a$CBI_burden_restrizione_tempo)
[1]  0  8 15 18 20
```

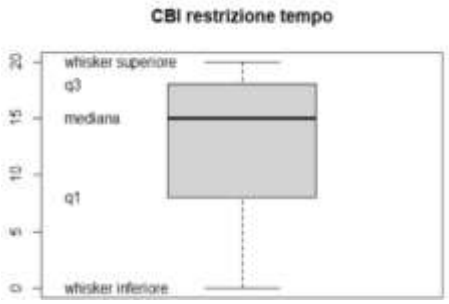
Sappiamo che vedremo il limite inferiore della scatola a 8 (Q_1), il limite superiore a 18; la riga che rappresenta la mediana a 15. Possiamo calcolare i whiskers:

```
IR<-18-8
inferiore<-8-(1.5*IR)
superiore<-18+(1.5*IR)
c(inferiore,superiore)
[1] -7 33
```

Il whisker inferiore è sotto il minimo della distribuzione ($min = 0$), quello superiore è sopra il massimo ($max = 20$): nel grafico, i wisker saranno quindi impostati ai valori minimo e massimo, e non ci saranno outlier.

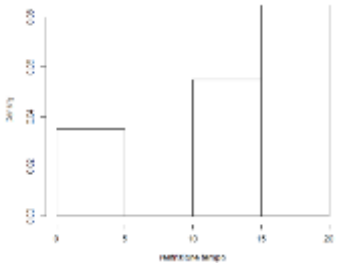
Vediamo se abbiamo ragione, usando `boxplot(distribuzione)`; aggiungiamo la descrizione degli elementi del grafico con `text`.

Scrivete gli script che hanno prodotto il boxplot a fianco e i grafici successivi, compreso il testo all'interno.



Scommessa vinta. **Lo stress** derivante dall'avere poco tempo per sé sembra un **problema rilevante per molti: la distribuzione è ampia e variabile,** ma la **maggior parte dei punteggi sembra addensarsi nella parte superiore** della scala: ricordiamo che la mediana divide la distribuzione in modo simmetrico, per cui metà dei soggetti ha punteggi compresi (anzi, compressi) tra 15 e 20 (asimmetria negativa). La distribuzione non sembra avere molte chance di essere normale.

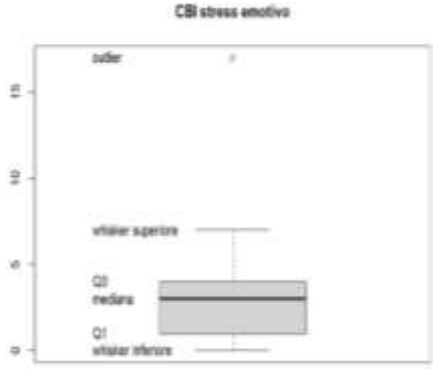
Potremmo anche verificare la distribuzione di densità con il noto `hist`.



Vediamo come si presenta la distribuzione del **burden emotivo**, cioè la risonanza emotiva negativa dovuta a comportamenti imprevedibili e inappropriati dell'assistito: `$CBI_burden_emotivo`. Vediamo `summary` e calcoliamo i whiskers:

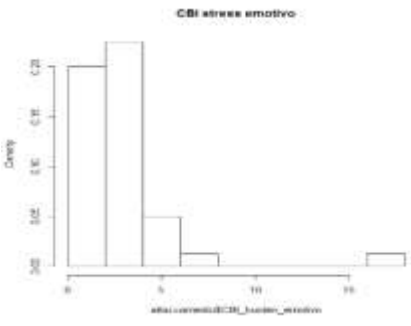
```
summary(a$CBI_burden_emotivo)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   3.000   3.075   4.000  17.000
IR<-4-1
inferiore<-1-(1.5*IR)
superiore<-4+(1.5*IR)
c(inferiore, superiore)
[1] -3.5  8.5
```

Il whisker inferiore, essendo più basso del valore minimo rilevato nella distribuzione ($min = 0$), sarà impostato a 0 nel grafico; evidentemente, per almeno un caregiver i comportamenti inappropriati dell'assistito non rappresentano un problema. D'altronde, il wisker superiore è pari a 8.5, ma nella distribuzione troviamo almeno un valore a lui superiore, cioè il valore massimo $max = 17$.



La distribuzione è molto più compatta di quella precedente, con punteggi prevalentemente bassi: i caregiver sembrano in gran maggioranza non lamentare stress per questo aspetto specifico – tranne naturalmente il/i caregiver lassù in alto.

Vediamo con `hist`:



Possiamo sapere chi è / chi sono gli outlier in molti modi, per esempio con **`which`**: la conosciamo, identifica il caso che soddisfa i requisiti specificati nel suo argomento. Noi dobbiamo identificare i casi con valore `$CBI_burden_emotivo` superiori a 7, perciò:

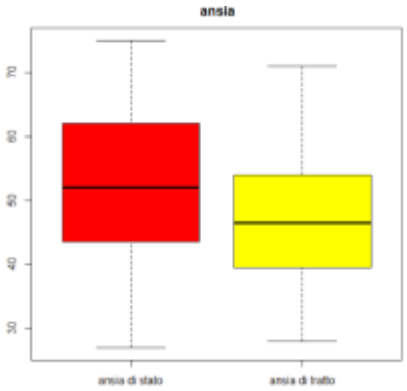
```
which(a$CBI_burden_emotivo>7)
[1] 2
```

*La persona con maggior carico emotivo è il **caregiver numero 2**: potete descrivere le sue caratteristiche, cioè **tracciare un suo profilo**? Segnala disagio emotivo di altro tipo? A quale gruppo appartiene? Chi lo/la aiuta?*

In uno **stesso grafico possono essere rappresentate fianco a fianco due o più distribuzioni**, ma ha senso farlo se le distribuzioni hanno la **stessa unità di misura** e lo **stesso range teorico** minimo-massimo: basta **separare le due distribuzioni con una virgola**. Dovremo, però, specificare i nomi delle variabili rappresentate con l'argomento `names=c("nome1","nome2")`, perché altrimenti in ascissa non sarà indicato nulla.

Rappresentiamo nello stesso grafico i boxplot dell'ansia di stato e dell'ansia di tratto (`$STAI_stato` e `$STAI_tratto`):

```
boxplot(a$STAI_stato, a$STAI_tratto, main="ansia",
col=rainbow(6), names = c("ansia di stato","ansia di tratto"))
```



Si direbbe che l'ansia dovuta alla situazione contingente prevalga, rispetto alla predisposizione ansiosa.

È possibile, e lo faremo molte volte, anche rappresentare in un boxplot la **distribuzione di una misura in funzione di una variabile di raggruppamento** / fattore, ovvero rappresentare una misura per tutti i livelli della variabile di raggruppamento. Il modo in cui facciamo capire a R che vogliamo "una cosa in funzione di un'altra" è il carattere **tilde ~**, che compone una **formula** del tipo $Y \sim X$ (Y **in funzione di** X), con cui lavoreremo per i mesi a venire. La tilde non è un carattere di tastiera, ma può essere richiamato come carattere ASCII. In Windows si ottiene con la combinazione di tasti Alt + 126, in MacOS con Alt+5. Altrimenti, si può inserire come simbolo in un editor di testi, copiarlo e incollarlo in uno script di R da tenere pronto per copiare e incollare ~ quando serve. Infine, potete creare l'oggetto tilde:

```
(tilde<-rawToChar(as.raw(126)))
[1] "~"
```

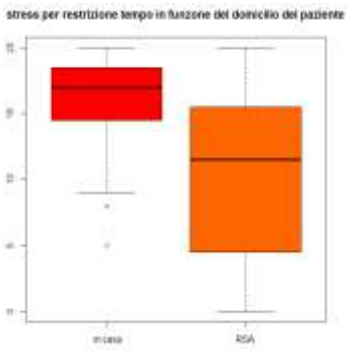
e copiare e incollare il suo output nelle formule.

Per esempio, esploriamo la distribuzione dello stress derivante dall'avere poco tempo per sé (`$CBI_burden_restrizione_tempo`) in funzione del luogo in vive l'assistito (`$domicilio_assistito`), ovvero a casa con il caregiver o in RSA. Per usare descrittori numerici, sappiamo di poter usare `tapply`:

```
tapply(a$CBI_burden_restrizione_tempo,a$domicilio_assistito,summary)
$` in casa`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   14.75   17.00   15.60   18.25   20.00
$RSA
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    4.75   11.50   10.20   15.25   20.00
```

Per visualizzare la distribuzione nei due livelli del fattore `$domicilio`, usiamo il boxplot:

```
boxplot(a$CBI_burden_restrizione_tempo~a$domicilio_assistito,
col=rainbow(15), main="stress per restrizione tempo in funzone del
domicilio del paziente")
```



Si direbbe che il ricovero in RSA abbia ragionevolmente diminuito lo stress derivante dalla mancanza di tempo. Notiamo gli outlier nel livello "in casa", minori rispetto al baffo inferiore:

```
14.75-1.5*(18.25-14.75)
[1] 9.5
```

... e se vogliamo sapere chi sono, possiamo usare `which`, specificando entrambi gli attributi: una misura inferiore a 9.5 **e** l'appartenenza al livello "in casa":

```
which(a$CBI_burden_restrizione_tempo<9.5 & a$domicilio_assistito=="in casa")
[1]  1  2 20
```

Ops, **sono tre**, non due come sembra dal grafico. I loro punteggi sono:

```
a[c(1,2,20),10]
[1] 5 9 8
```

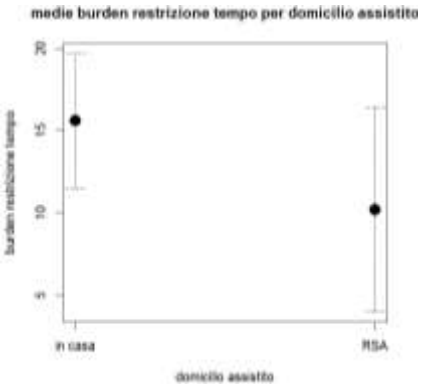
I cerchietti corrispondente a 8 e a 9 sono sovrapposti.

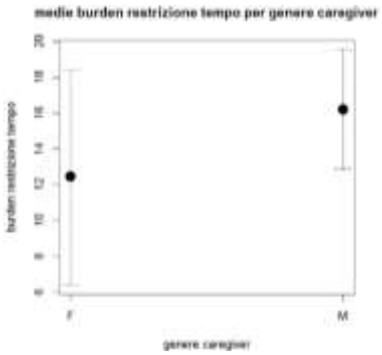
*Verificate la distribuzione delle **altre variabili di burden nei due livelli della variabile** **`$domicilio_assistito`**: anche le altre dimensioni di stress sembrano risentire del gruppo di appartenenza? Ricoverare temporaneamente l'assistito allevia anche altre dimensioni di carico assistenziale? Perché?*

Se avete scaricato il package `Rcmdr` con le sue dependencies per utilizzare `RCommander`, potete usare i grafici installati nella dependency `RcmdrMisc` per ampliare i grafici a disposizione con un grafico per visualizzare le **medie**, con relativa dispersione, dei livelli di un fattore. Se non avete intenzione di usare `Rcommander`, potete scaricare solo `RcmdrMisc`, che tornerà utile anche nell'analisi della varianza fattoriale. Useremo **`plotMeans(response= misura, factor= fattore, error.bars`** = `indici di dispersione)`: i suoi argomenti delineano la variabile da rappresentare (`response`), il fattore per i livelli dei quali saranno rappresentate le medie della variabile (`factor1`) e un indice di dispersione (`error.bars`), per cui potete scegliere tra deviazione standard `="sd"`, l'errore standard della media (`="se"`) o l'intervallo di fiducia (`="conf.int"`): di questi ultimi parleremo nel capitolo 6. I punti che rappresentano le medie sono di default connessi da una linea, che può essere omessa con `connect=FALSE`.

Vediamo come si presentano le medie del burden derivante dal poco tempo a disposizione, di cui abbiamo rappresentato le mediane con il boxplot; come indice di dispersione, usiamo la **deviazione standard**:

```
plotMeans(response=  a$CBI_burden_restrizione_tempo,  factor1=
  a$domicilio_assistito, pch = 19, xlab="domicilio assistito",
  ylab="burden restrizione tempo", main="medie burden restrizione
  tempo per domicilio assistito",error.bars = "sd",connect=FALSE)
```





Così invece si presentano le medie dell'ansia di stato per genere (provate a scrivere la funzione da soli).

4.4 Indici di forma

In questo paragrafo uniremo i grafici appena imparati (`hist`, in particolare) a indici numerici (**indici di forma**) per completare il quadro delle tecniche usate per la comprensione delle caratteristiche di una distribuzione statistica, dopo aver visto ***medie e mediane*** avere un'idea dell'ordine di grandezza del fenomeno e gli ***indici di dispersione*** per segnalare il grado di diversità tra le singole manifestazioni del fenomeno.

4.4.1 "The supreme law of Unreason": la distribuzione normale

Le distribuzioni di frequenza possono assumere molte forme diverse, ma è importante avere una descrizione generale che si adatti ai più comuni tipi di distribuzione. I dati di una variabile **continua** potrebbero essere distribuiti simmetricamente attorno alla tendenza centrale di tutti i punteggi: tracciando una linea verticale attraverso il centro della distribuzione, essa sarebbe speculare ai due lati della linea. Questa è una caratteristica della **distribuzione normale**.

È molto probabile che nessun'altra astrazione matematica abbia avuto un'influenza sulla psicologia e le scienze sociali pari a quella della curva a campana o normale, nome che Pearson ha importato in queste discipline. I primi passi della distribuzione normale si devono a **Laplace** (1812), che, lavorando sulle distribuzioni di probabilità (capitolo 5) ha interpretato la curva come **legge dell'errore**, dimostrando che poteva essere applicata a risultati variabili in maniera impredicibile (**aleatori**) in multiple osservazioni: la sua prima applicazione al di fuori del gioco d'azzardo (questa è la sua poco nobile origine) fu la stima degli errori nelle previsioni astronomiche, e qualche anno dopo furono i problemi di puntamento nel fuoco di artiglieria. Successivamente, **Gauss** (1855) sistematizzò le osservazioni di Laplace nella famosa funzione della "curva gaussiana" o distribuzione di Laplace - Gauss, che nel corso del XIX secolo trovò ampia fama e utilizzo, anche grazie alla crescita delle compagnie assicurative (assai interessate al concetto di "errore di previsione" e "probabilità di sopravvivenza") e all'applicazione di un approccio statistico alle scienze biologiche e sociali.

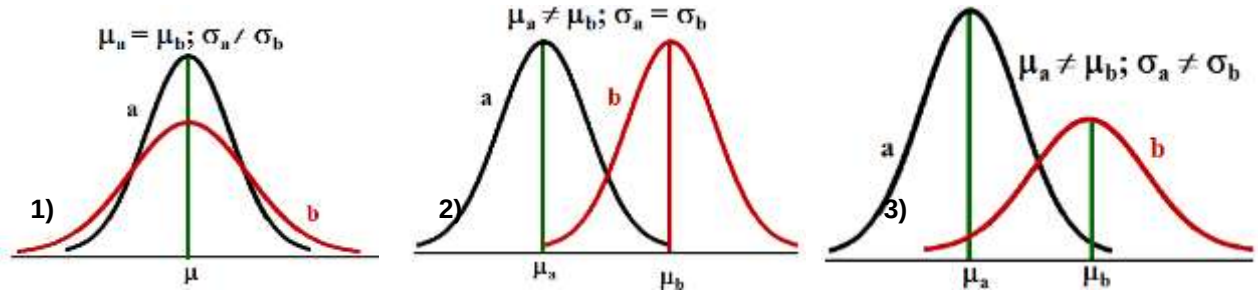
Il passaggio alla denominazione della funzione gaussiana come "curva normale" si deve all'osservazione che molte variabili biologiche, **se misurate in grandi gruppi** di individui e rappresentate graficamente come distribuzioni di frequenza, mostrano una stretta **approssimazione** alla curva normale: in realtà, la "vera" **distribuzione normale è solo teorica**: l'istogramma di frequenza, per quanto piccole possano essere le classi, è una curva ***discontinua***, non continua**.** La responsabilità, o il merito, dell'estensione dell'uso di calcoli nati per stimare gli errori di previsione nel gioco d'azzardo alla valutazione di caratteristiche umane si deve a **Quetelet** (1835; peraltro, era un astronomo). Quetelet descrisse il suo concetto di *homme moyen* o **uomo medio**: è l'idea che la Natura ha di come deve essere un uomo, un **ideale che corrisponde a un valore misurato medio**. Ma la **Natura commette errori**, e così facendo, cioè mancando il bersaglio, **crea la variabilità osservata** nei tratti fisici e nelle caratteristiche psicologiche ("morali", avrebbero detto i contemporanei di Quetelet). **L'estensione e la frequenza di questi errori** della Natura si **conformano** alla legge della frequenza degli errori, ovvero alla **distribuzione normale**.

Galton (le sue note caratteristiche, per i curiosi, sono abbozzate nell'Appendice III) fu grandemente impressionato dal lavoro di Quetelet, le cui osservazioni si adattavano a molti dei suoi dati: "Conosco ben poche cose così capaci di colpire l'immaginazione come la meravigliosa forma di ordine cosmico espressa dalla "legge di frequenza degli errori". La legge sarebbe stata personificata dai Greci e deificata, se l'avessero conosciuta. Regna con serenità e in volontaria discrezione in mezzo alla più selvaggia confusione. Tanto più smisurata è la moltitudine e tanto più grande è l'anarchia, quanto più perfetta appare la sua regola. **È la suprema legge dell'Irrazionale**. Ogni volta che una grande massa di elementi caotici viene raccolta, e questi elementi sono schierati secondo l'ordine della loro grandezza, un'insospettabile e splendida forma di regolarità dimostra di essere stata latente fin dall'inizio"[20]. Questa concezione quasi mistica, ed

[20] *Natural Hineritance*, 1889, pag.66

estremamente vittoriana nella sua sensibilità, di un ordine all'opera sotto il caos non ha retto alla prova del tempo (almeno per i più), ma la rappresentazione della curva normale come raffigurazione di come "devono andare le cose", secondo un supposto ordine naturale, si è insinuata nelle scienze sociali, anche grazie al lavoro di **Pearson**, allievo e sodale di Galton.

Noi affronteremo la descrizione della curva "normale" con un approccio più "laico". La distribuzione normale è in realtà una **famiglia** di distribuzioni, definite da due **parametri:** la media μ e la deviazione standard σ: nelle figure sottostanti vediamo tre esempi di distribuzioni 1) con uguale media e diversa deviazione standard; 2) con diversa media e uguale deviazione standard; 3) con diversa media e diversa deviazione standard.



in ascissa i valori di X, in ordinata la loro frequenza

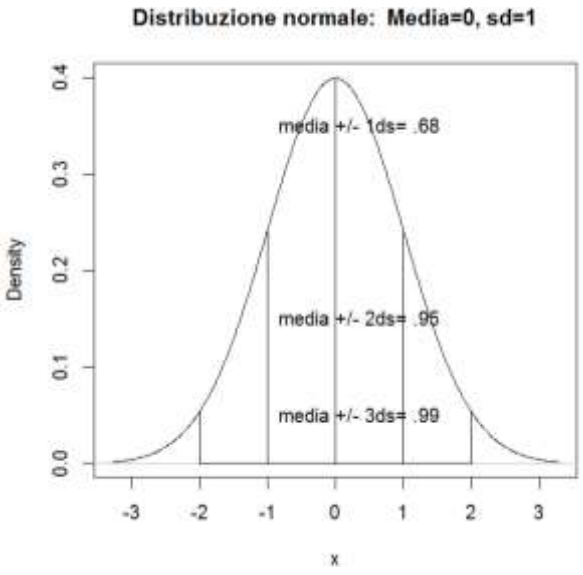
La distribuzione normale è *asintotica*: non tocca mai l'ascissa X (si riferisce a popolazioni infinite), se non in corrispondenza di $\pm \infty$. **Moda, mediana e media coincidono nel valore centrale**; ogni metà della curva presenta punti di flesso, in cui la curva cambia direzione, corrispondenti a $\pm\sigma$.

Il 100% dei casi della popolazione è compreso nell'area delimitata dalla curva: l'area sottesa dall'intera curva è quindi $area = 1$. Per **qualunque valore di μ e σ**, in una distribuzione normale l'area corrispondente a intervalli definiti / la **proporzione di casi compresi sotto la curva è sempre la stessa:**

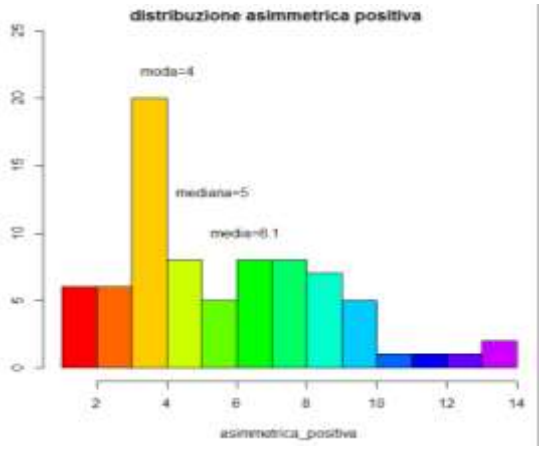
$$\mu \pm 1\sigma = .68; \mu \pm 2\sigma = .95; \mu \pm 3\sigma = .99$$

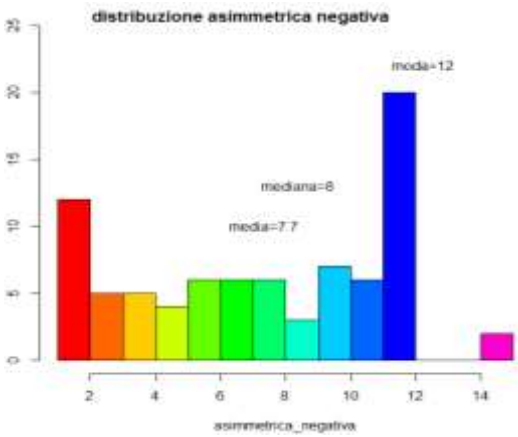
Il grafico a fianco si può ottenere o con RCommander → Distribuzioni → Distribuzione continue → distribuzione normale → disegna distribuzione normale cui si aggiunge `text` nello script, oppure così:

```
x <- seq(-3.291, 3.291, length.out=1000)
plotDistr(x, dnorm(x, mean=0, sd=1), cdf=FALSE, xlab="x",
  ylab="Density", main=paste("distribuzione normale:
  Mean=0, Standard deviation=1"), regions=list(c(-2, -1),
  c(-1, 0), c(0,1), c(1,2)), col=c('#FFFFFF',
  '#FFFFFF','#FFFFFF','#FFFFFF'), legend=FALSE)
text(x =-1, y=c(0.35, 0.15, 0.05), labels = c("media +/-
  1ds= .68", "media +/- 2ds= .95", "media +/- 3d s= .99"),
  pos = 4)
```



Dato che la curva normale è simmetrica, l'area compresa tra $-\infty$ e μ è $= .50$, come l'area compresa tra μ e $+\infty = .50$. Gli indici di forma della distribuzione normale teorica fungono da paragone per gli indici di forma delle altre distribuzioni. È la coincidenza di moda, mediana e media in un unico valore a rendere la curva simmetrica: quando i tre indici non sono equivalenti, la distribuzione è **asimmetrica**. L'**asimmetria** (**skewness**) può essere positiva (> 0) o negativa (< 0). Quando $media > mediana > moda$, **l'asimmetria è positiva**: la distribuzione presenta il **maggior numero di casi verso i valori più piccoli e una coda – tail – più lunga a destra**. Quando $media < mediana < moda$, **l'asimmetria è negativa**: la distribuzione presenta il maggior numero di casi verso i valori più alti e una **coda** più lunga a **sinistra**.



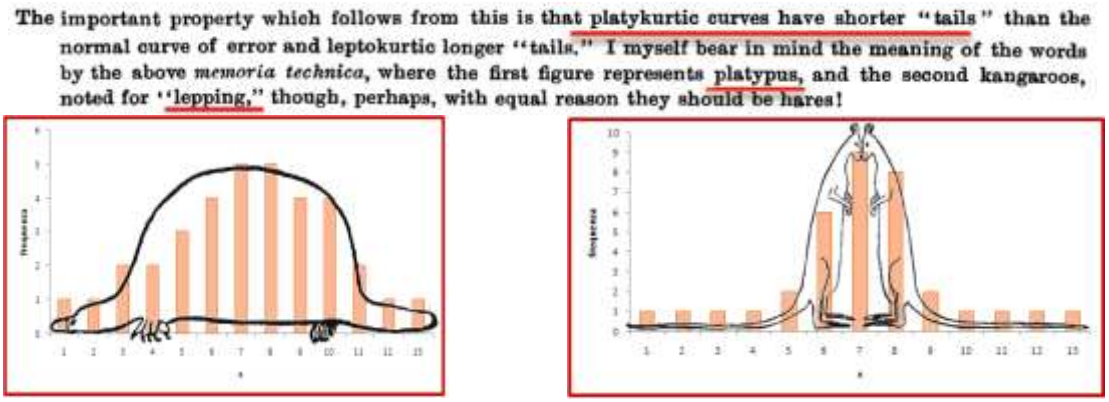



```
mean(asimmetrica_positiva)
[1] 6.089744
median(asimmetrica_positiva)
[1] 5
table(asimmetrica_positiva)
asimmetrica_positiva
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
 2  4  6 20  8  5  8  8  7  5  1  1  1  2
```

```
mean(asimmetrica_negativa)
[1] 7.731707
median(asimmetrica_negativa)
[1] 8
table(asimmetrica_negativa)
asimmetrica_negativa
 1  2  3  4  5  6  7  8  9 10 11 12 15
 4  8  5  5  4  6  6  6  3  7  6 20  2
```

L'altro indice di forma è la **curtosi** (**kurtosis**): indica quanto le code della distribuzione siano più o meno addensate attorno ai valori centrali rispetto alle code di una distribuzione normale. Nelle formule originali di Pearson[21], asimmetria e curtosi di una distribuzione normale teorica risultavano pari a 3; le formule sono state successivamente riadattate in modo che entrambi gli indici di forma della distribuzione normale teorica fossero $= \mathbf{0}$, facilitandone l'interpretazione. In questa modalità, una curtosi $kur = \mathbf{0}$ indica perfetta **coincidenza con la gaussiana** (distribuzione **mesocurtica**: *mesos*= normale). Una curtosi **negativa** ($kur < 0$) indica un eccesso relativo di osservazioni nelle zone intermedia a destra e a sinistra del centro e scarsità relativa di osservazioni al centro e nelle code estreme: la distribuzione è definita **platicurtica** (*platùs*= piatto), con **code scarsamente differenziate** dai valori centrali. Una curtosi **positiva** ($kur > 0$) indica scarsità relativa di osservazioni a destra e a sinistra del centro e un eccesso relativo nella zona centrale: la distribuzione è definita **leptocurtica** (*leptòs*: sottile), con **code chiaramente differenziate** dai valori centrali. Convenzionalmente, asimmetria e curtosi comprese nell'intervallo tra -1 e +1 (o anche tra -1.5 e +1.5) indicano deviazioni trascurabili rispetto alla simmetria e alla curtosi di una distribuzione normale teorica; quanto più i valori si allontanano da questi limiti, tanto più le alterazioni della forma rispetto alla normale sono marcate.

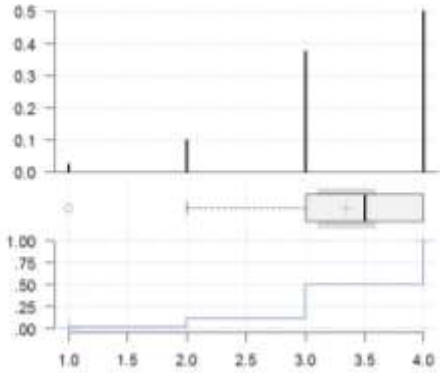
Curiosità: il termine "code" della distribuzione deriva da un'originale "mnemotecnica" di Gosset (*alias* Student, 1927):


The important property which follows from this is that platykurtic curves have shorter "tails" than the normal curve of error and leptokurtic longer "tails." I myself bear in mind the meaning of the words by the above *memoria technica*, where the first figure represents platypus, and the second kangaroos, noted for "lepping," though, perhaps, with equal reason they should be hares!

[21] $asimmetria = 3\frac{\sum(\bar{x}-mediana)}{s_x}$; $curtosi = \frac{\sum(x_i/s)^4}{N-3}$

In R, ahimè, *skewness* e kurtosis non sono disponibili tra le statistiche di base, ma le funzioni per richiederle sono presenti in un gran numero di package. Abbiamo già usato `DescTools`: se vi servono **solo** le informazioni sui due descrittori di forma, potete usare le sue funzioni `Skew(distribuzione)` e `Kurt(distribuzione)`; se invece servono anche altre descrittive e amate i grafici, usate `Desc`: quando il suo oggetto è una variabile `numeric` o `integer`, tra le molte informazioni troviamo anche *skewness* e curtosi, oltre ai tre grafici prodotti con l'output. Applichiamola alla variabile `$QOL_salute_fisica` e vediamo come si distribuisce la soddisfazione dei caregiver per la propria salute:

```
Desc(attaccamento$QOL_salute_fisica)
-------------------------------------------------------
attaccamento$QOL_salute_fisica (integer)

  length       n     NAs  unique      0s    mean  meanCI'
      40      40       0       4       0    3.35     3.10
          100.0%   0.0%           0.0%            3.60

     .05     .10     .25  median    .75     .90     .95
    2.00    2.00    3.00    3.50   4.00    4.00    4.00

   range      sd   vcoef     mad     IQR    skew    kurt
    3.00    0.77    0.23    0.74    1.00   -0.99    0.39
```

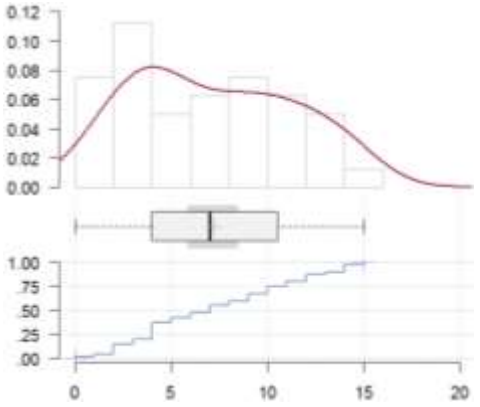


[omissis]

L'asimmetria è **negativa**: i soggetti tendono ad accumularsi verso i valori alti della distribuzione e la coda si prolunga a sinistra; quindi, i caregiver hanno in maggioranza alti punteggi di qualità della vita per la dimensione di salute; la curtosi è positiva, quindi la distribuzione è leptocurtica, con una coda piuttosto visibile.

Vediamo, invece, la distribuzione dello stress derivante dall'onere prettamente fisico dell'assistenza (`$CBI_burden_fisico`):

```
Desc(attaccamento$CBI_burden_fisico)
-------------------------------------------------------
attaccamento$CBI_burden_fisico (integer)

  length       n     NAs  unique      0s    mean  meanCI'
      40      40       0      16       1    7.17     5.84
          100.0%   0.0%           2.5%            8.51

     .05     .10     .25  median    .75     .90     .95
    1.95    2.00    4.00    7.00   10.25   13.10   14.00

   range      sd   vcoef     mad     IQR    skew    kurt
   15.00    4.19    0.58    4.45    6.25    0.21   -1.21
```

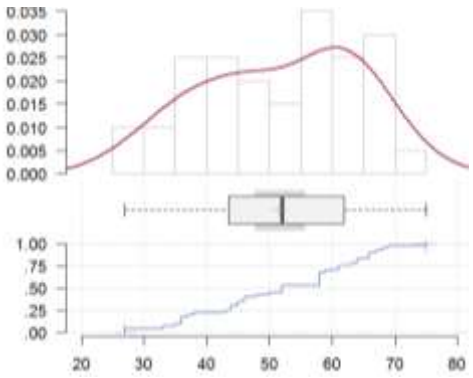


[omissis]

L'asimmetria è **positiva**, quindi i soggetti tendono a riportare soprattutto i valori più bassi della distribuzione; la curtosi è negativa, quindi la distribuzione è "piatta", con code scarsamente differenziate rispetto ai valori centrali: il carico fisico non sembra rappresentare un grave problema per la maggioranza dei caregiver.

Infine, vediamo la distribuzione dell'ansia di stato:

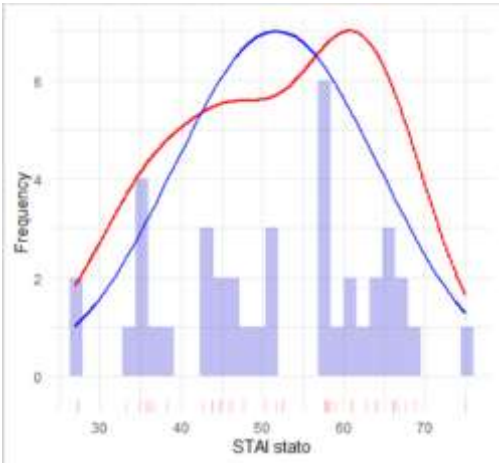
```
Desc(attaccamento$STAI_stato)
------------------------------------------------------------
attaccamento$STAI_stato (integer)

  length       n     NAs  unique      0s    mean  meanCI'
      40      40       0      23       0   51.73    47.69
          100.0%   0.0%            0.0%           55.76

     .05     .10     .25  median     .75     .90     .95
   32.70   35.90   43.75   52.00   61.50   66.10   68.05

   range      sd   vcoef     mad     IQR    skew    kurt
   48.00   12.62    0.24   13.34   17.75   -0.23   -1.07
```



[omissis]

Asimmetria e curtosi sono entrambe negative: tendono a prevalere i punteggi di ansia di stato più alti, con coda a sinistra, e la distribuzione è leptocurtica, con code differenziate dai valori centrali.

Per visualizzare un istogramma di frequenze che associa la distribuzione delle frequenze osservate con la distribuzione normale, per un più facile confronto, potete usare `normalHist(vector= distribuzione, normalCurve= TRUE, distCurve= TRUE)` del package `ufs`; di default, sono mostrate sia la distribuzione empirica (`distCurve`), come fa `Desc`, sia la normale teorica (`normalCurve`).




```
normalHist(vector=  a$STAI_stato,  normalCurve=  TRUE,
  distCurve = TRUE, normalColor = "blue", distributionColor
  = "red", xLabel = "STAI stato")
```

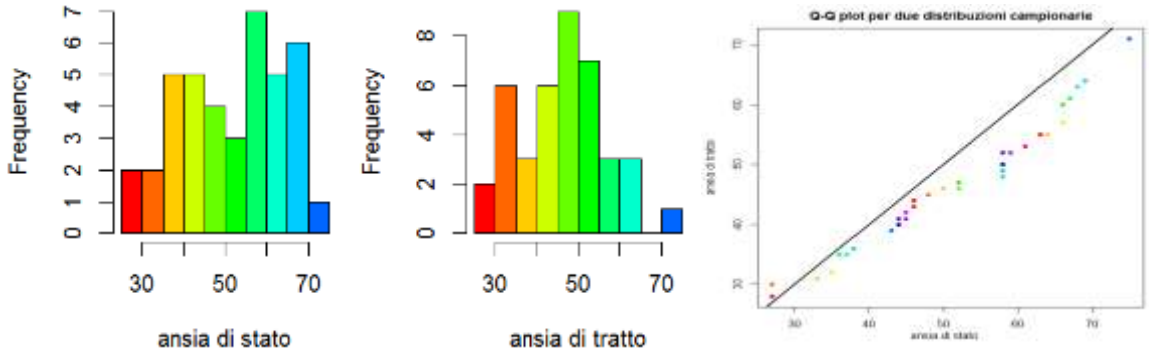
Esiste un particolare grafico che aiuta la valutazione sulla sovrapposizione di una distribuzione di frequenza campionaria a una distribuzione normale teorica: il **Q-Q plot**, ovvero grafico quantile - quantile.

In effetti, il Q-Q plot mette genericamente a confronto i **quantili** di due distribuzioni X e Y: ogni punto del Q-Q plot ha come coordinata l'N-esimo quantile di X e il corrispondente quantile di Y: **se le due distribuzioni X e Y hanno un andamento simile, i punti del Q-Q plot si dispongono approssimativamente su una retta $x = y$.** È quindi possibile usare il Q-Q plot per confrontare la forma di **due distribuzioni campionarie**, così come per **confrontare la forma di una distribuzione campionaria con una attesa** (**normale**, perlopiù). In entrambi i casi, la logica del confronto è la stessa: tanto più le distribuzioni si assomigliano, tanto più i punti del Q-Q plot si disporranno ordinatamente sulla retta. Il modo in cui **non** si dispongono sulla retta dà informazioni sulla natura della distorsione: asimmetria destra o sinistra, curtosi positiva o negativa.

In R, `qqplot(distribuzione 1, distribuzione 2)` confronta la forma di due distribuzioni campionarie; per aggiungere la retta di riferimento, si aggiunge `abline`Errore. Il segnalibro non è definito.`(a=0,b=1)`, dove `a=0` esprime l'intercetta b_0 della retta (il punto in cui passa per Y: in questo caso l'origine degli assi) e `b=1` esprime il coefficiente angolare b_1, cioè la variazione unitaria in Y al variare di una unità in X, che determina la pendenza della retta: se $b_1 = 1$, per ogni punto in più di X, Y aumenta di 1 punto (diremo tutto quello che vorrete sapere su intercetta e coefficiente angolare, e anche qualcosa di più, nel capitolo 9).

Confrontiamo, ad esempio, le distribuzioni di ansia di stato e ansia di tratto:

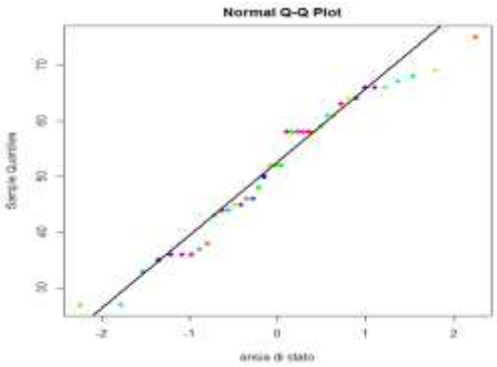
```
qqplot(a$STAI_stato,a$STAI_tratto, pch=19, col=rainbow(15), xlab = "ansia di stato", ylab="ansia
  di tratto", main="Q-Q plot per due distribuzioni campionarie")
abline(0,1, lwd=2)
```



La forma delle due distribuzioni non sembra molto analoga.

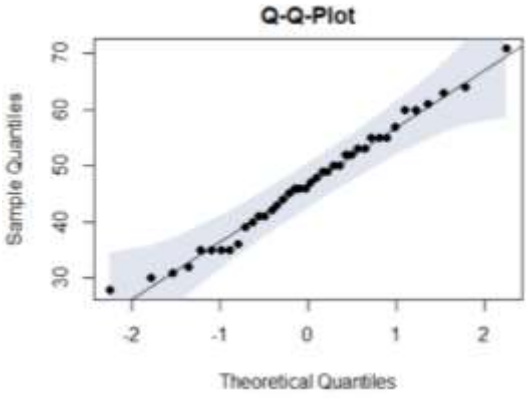
Nel Q-Q plot applicato al **confronto tra una distribuzione campionaria e una distribuzione teorica normale**, le osservazioni sono plottate rispetto ai quantili di una distribuzione normale standardizzata, con $\mu = 0$ e $\sigma = 1$: sotto l'assunzione di normalità, il grafico dovrebbe approssimarsi a una linea retta, in diagonale dall'origine degli assi. In R, usiamo **`qqnorm(distribuzione)`**, aggiungendovi la funzione **`qqline`**Errore. Il segnalibro non è definito.**`(distribuzione)`** per tracciare la retta di riferimento: in ascissa avremo i quantili della variabile empirica, in ordinata i quantili della distribuzione normale teorica: tanto più gli elementi del grafico si disporranno sulla retta, tanto più affine sarà la distribuzione alla normale.

```
qqnorm(a$STAI_stato, pch=19, col=rainbow(15), xlab =
  "ansia di stato")
qqline(a$STAI_stato,lwd=2)
```



La variabile `$STAI_ansia_stato` non è ben sovrapposta alla retta di riferimento.

Per confrontare una distribuzione campionaria con una teorica (anche diversa dalla distribuzione normale) potete usare `PlotQQ(x= distribuzione, qdist = quantili della distribuzione teorica)` di `DescTools`, che aggiunge la linea di riferimento di default. Per il confronto con la distribuzione normale, `qdist= qnorm`. Oltre alla retta di riferimento si delinea anche il ***confidence band*** della retta: lo vedremo bene nella regressione, per ora potete interpretarlo come il range entro cui, in campionamenti ripetuti, si dovrebbe trovare la distribuzione delle coordinate, con una verosimiglianza prefissata).



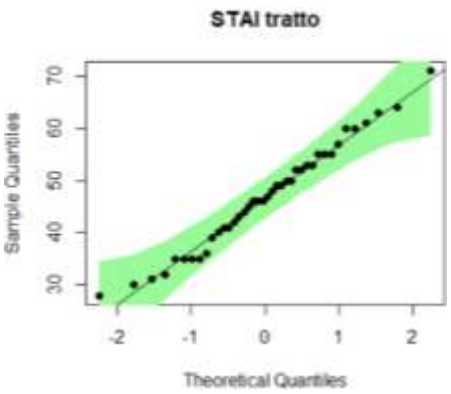

```
PlotQQ(a$STAI_tratto, qdist = qnorm, pch=19)
```

La particolare conformazione dei punti che non si sovrappongono alla retta dà informazioni su asimmetria e curtosi:

Andamento Q-Q plot	Forma della distribuzione
a U, concavità verso l'alto	Asimmetria a destra
A ∩, concavità verso il basso	Asimmetria a sinistra
Parte iniziale sotto la linea, parte finale sopra la linea	Code molto lunghe, curtosi positiva
Parte iniziale sopra la linea, parte finale sotto la linea	Code molto corte, curtosi negativa

Nel grafico precedente non è chiaramente evidenziabile una conformazione a U o ∩, mentre è abbastanza evidente la curtosi negativa (confrontate l'interpretazione del grafico con i valori di asimmetria e curtosi calcolati in precedenza).

```
PlotQQ(a$STAI_tratto, qdist = qnorm, pch=19, args.cband = list(col="pale green"), main = "STAI tratto")
```



La distribuzione dell'ansia di tratto è decisamente più sovrapponibile a quella di una distribuzione normale, il che spiega l'insoddisfacente Q-Q plot delle due distribuzioni campionarie. Infatti, se osserviamo gli indici di curtosi e asimmetria, confermiamo l'impressione tratta dal Q-Q plot:

```
Skew(a$STAI_stato); Kurt(a$STAI_stato)
[1] -0.2313227
[1] -1.071975
Skew(a$STAI_tratto);Kurt(a$STAI_tratto)
[1] 0.1389546
[1] -0.7128255
```

4.6 Correggere le distribuzioni: trasformazioni non lineari

Come abbiamo visto nel paragrafo precedente, le distribuzioni dei dati possono (e lo fanno con inquietante frequenza) non aderire in maniera soddisfacente alla distribuzione normale teorica. Di per sé, questo non costituisce certo un problema per la descrizione dei dati: anzi, può capitare che proprio i dati anomali siano i casi più interessanti da approfondire. Tuttavia, l'assunzione di distribuzione normale è un prerequisito per l'applicazione di molti test inferenziali che affronteremo dal capitolo 6 in poi: sono i cosiddetti test parametrici, le cui conclusioni sono affidabili se la distribuzione (in alcuni test la distribuzione dei dati del campione, in altri la distribuzione degli **errori** del modello) segue determinati **parametri**, cioè se è analoga alla normale. Nel caso in cui la distribuzione non rispetti questo prerequisito di applicabilità, le opzioni sono:

a) **cambiare la distribuzione di probabilità di riferimento e, di conseguenza,** il **test** inferenziale che s'intende usare: invece di usare un test parametrico in condizioni che non rispettano l'assunto di normalità, si possono usare test non parametrici o **robusti**, cioè non influenzati da casi anomali (outlier: §4.7) e asimmetria nei dati. Vedremo (alcuni dei) test non parametrici, le statistiche robuste di Wilcox e la regressione logistica nei capitoli successivi;

b) **trasformare la distribuzione** per renderla più "normale": di questo ci occupiamo in questo paragrafo.

Trasformare la forma della variabile non significa "manipolare i dati per far emergere quel che si attende", come qualche malizioso potrebbe pensare. In realtà, nella trasformazione non cambiano le relazioni tra le variabili (le differenze relative tra i soggetti per una data variabile restano le stesse), ma cambiano le differenze tra diverse variabili, perché cambiano le unità di misura. D'altronde, la trasformazione dei dati resta un punto controverso in letteratura, con posizioni a confronto (acceso) tra pro e contro. Per non affrontare in dettaglio la *querelle*, limitiamoci a citare i punti elencati da Games (1984):

1. il teorema centrale del limite (capitolo 5) ci dice che per grandi campioni la distribuzione campionaria tenderà comunque a essere normale; quindi, il **dibattito sulla necessità della trasformazione è realmente importante solo per campioni piccoli.** Le prime ricerche avevano dimostrato che in campioni con $N \geq 40$ la forma della

distribuzione sarebbe stata normale come predetto, ma i loro dati erano basati su distribuzioni leptocurtiche, con code "sottili", e ricerche successive hanno dimostrato che con distribuzioni platicurtiche la numerosità del campione dovrebbe essere decisamente > 40 per affidarsi al teorema centrale del limite (Wilcox, 2005). Per distribuzioni di questo tipo, le trasformazioni potrebbero essere necessarie per tendere alla normalità.

2. Trasformando i dati **si cambiano le ipotesi che vengono testate**: per esempio, usando una trasformazione logaritmica su due distribuzioni e confrontandone le medie, staremmo confrontando medie geometriche, e non medie aritmetiche: l'interpretazione della differenza tra queste medie sarebbe ovviamente diversa (Gelman e Hill, 2007).
3. In piccoli campioni è problematico stimare la normalità, qualsiasi modalità si usi.
4. Le **conseguenze sul modello statistico derivanti dall'applicazione di una trasformazione inadeguata sarebbero peggiori delle conseguenze derivanti dall'analisi su dati non trasformati**.

Quindi, **per questo esame**, quando avremo distribuzioni inadeguate e sarà disponibile un test robusto al posto del corrispettivo test parametrico (e R ne mette a disposizione moltissimi, soprattutto grazie al lavoro di Wilcox), privilegeremo i test robusti. Tuttavia, poiché in letteratura e nella pratica sono molti i casi in cui è lecito applicare trasformazioni alla distribuzione, vediamone almeno le principali: sono elencate nella tabella seguente, insieme alle caratteristiche della distribuzione che ne raccomandano l'uso, e brevemente descritte, con le funzioni di R, nei sottoparagrafi a seguire. Vale la pena ricordare che, una volta scelta la trasformazione, la stessa trasformazione deve essere applicata a **tutte** le variabili oggetto d'analisi!

Trasformazione non lineare	Apporta correzioni per:
Logaritmo in base naturale: log(X_i)	Asimmetria positiva
Radice quadrata: $\sqrt{X_i}$	Asimmetria positiva
Reciproco: 1/X_i	Asimmetria positiva
Esponenziale: X_i^n	Asimmetria negativa

4.6.1 Trasformazione logaritmica: $X_t = lg_{X_i}$

La trasformazione in logaritmo "schiaccia" la coda destra della distribuzione, e quindi può funzionare nel caso di un'asimmetria positiva. Se avete bisogno di un rapidissimo ripasso sui logaritmi (che ritroveremo nella regressione logistica), date un'occhiata all'Appendice VI. Con R, usiamo la funzione `log(X)`; se non si specifica la base, per default sono calcolati i logaritmi in base naturale:

```
log(1); log(10); log(50)
[1] 0
[1] 2.302585
[1] 3.912023
```

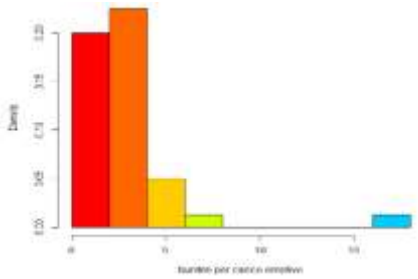
Tuttavia, i logaritmi **non si possono calcolare su 0 e numeri negativi**:

```
log(0); log(-1)
[1] -Inf
[1] NaN
```

Quindi, se la distribuzione originaria contiene questi valori essi devono essere prima trasformati, aggiungendo una costante a tutti i valori prima di trasformare la variabile.

Vediamo la distribuzione dello stress per le componenti emotive dell'attività di assistenza `a$CBI_carico_emotivo`:

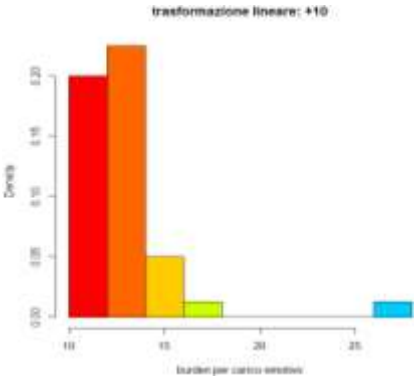
```
Skew(a$CBI_burden_emotivo)
[1] 2.749263
```



È (fortunatamente per i caregiver) **decisamente asimmetrica positiva**, tranne per un caregiver in evidente grave sofferenza. A parte l'umana simpatia per il caregiver in questione, un'asimmetria così forte pone problemi allo statistico: incide sull'affidabilità di test statistici parametrici per la verifica delle ipotesi, che assumono distribuzioni normali, ovvero simmetriche e unimodali.

Proviamo a normalizzare (anzi, a **log-normalizzare**) la distribuzione usando la **trasformazione logaritmica in base naturale** della variabile `$CBI_burden_emotivo`. Nel grafico vediamo che ci sono punteggi = 0:

```
summary(a$CBI_burden_emotivo)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   3.000   3.075   4.000  17.000
```



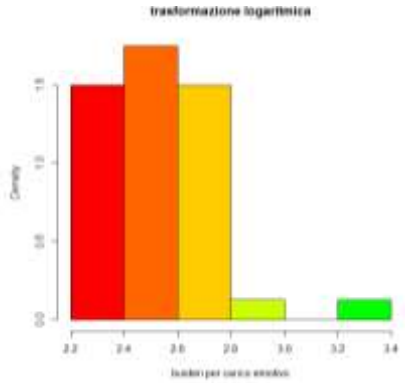
Quindi trasformiamo la variabile **linearmente**, aggiungendo un valore **costante** (ad esempio, $+10$) a tutti i dati, **prima** di trasformarla **NON linearmente** nel suo logaritmo.

```
burden_emotivo_corretto<-a$CBI_burden_emotivo+10
```

È evidente dal grafico che questa prima **trasformazione lineare** **lascia intatta la forma della distribuzione**.

Ora applichiamo la trasformazione logaritmica con la funzione **`log(variabile)`**:

```
burden_emotivo_log<-log(burden_emotivo_corretto).
Skew(burden_emotivo_log)
[1] 1.43035
```



La forma è decisamente cambiata: l'asimmetria positiva si è ridotta, anche se resta ancora lontana dal desiderato $skewness = 0$ della distribuzione normale teorica.

Attenzione: la base del logaritmo è irrilevante, dato che i dati trasformati saranno diversi solo per una **costante** moltiplicativa.

```
burden_emotivo_log2<-log(burden_emotivo_corretto, base = 2)
Skew(burden_emotivo_log2)
[1] 1.43035
burden_emotivo_log10<-log(burden_emotivo_corretto, base = 10)
Skew(burden_emotivo_log10)
[1] 1.43035
```

4.6.2 Trasformazione in radice quadrata: $X_t = \sqrt{X_i}$

Questa trasformazione, spesso applicata a distribuzioni di conteggi, è un caso particolare di **trasformazione esponenziale**, in cui la variabile da trasformare è elevata a potenza ½ : $X_t = X_o^{1/2}$. Calcolare la radice quadrata della distribuzione (**`sqrt(variabile`**) ha un effetto diverso su ogni caso: la trasformazione sarà maggiore per i valori di

partenza più grandi e minore per i valori più piccoli. Di conseguenza, la radice quadrata porterà i **valori più grandi verso il centro della distribuzione**, e questo potrebbe migliorare l'asimmetria positiva.

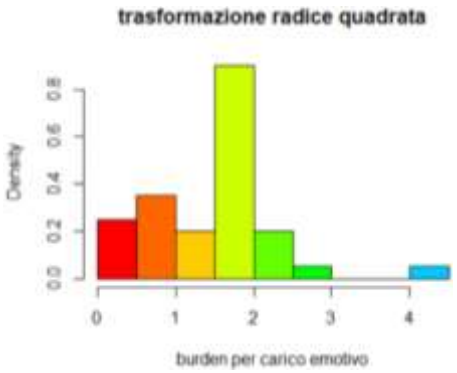
```
sqrt(1); sqrt(10); sqrt(50)
[1] 1
[1] 3.162278
[1] 7.071068
1-1; 10-3.162278; 50-7.071068
[1] 0
[1] 6.837722
[1] 42.92893
```

Come nel caso della trasformazione logaritmica, però, gli eventuali valori negativi dovranno prima essere trasformati in positivi, aggiungendo una costante.

Dato che la variabile `$CBI_burden_emotivo` non contiene valori negativi e la radice quadrata di 0 è un legittimo =0, possiamo trasformarla direttamente:

```
burden_emotivo_radice<- sqrt(a$CBI_burden_emotivo)
Skew(burden_emotivo_radice)
[1] 0.05412116
```

La forma è cambiata, e in modo diverso dalla trasformazione logaritmica: la riduzione dell'asimmetria positiva è decisamente molto più efficace.



Se la distribuzione presentasse valori $x_i = 0$ e una media $\bar{X} < 1$, sarebbe opportuno modificare leggermente la trasformazione, aggiungendo 0.5 al valore grezzo prima di operare la trasformazione: $X_t = \sqrt{X_i + 0.5}$

4.6.3 Trasformazione in reciproco: $X_t = \frac{1}{X_i}$

Dividere 1 per ciascuno dei dati riduce l'impatto dei valori più grandi, che tenderanno a 0. Oltre a moderare l'asimmetria positiva, la trasformazione in reciproco stabilizza la varianza della distribuzione. Nell'interpretazione, è essenziale ricordarsi che la trasformazione rovescia i punteggi: i valori più grandi diventano più piccoli, ma i punteggi che originariamente erano piccoli diventano grandi; è quindi conveniente tornare ai valori originari, per la loro descrizione.

```
1/1; 1/10; 1/50
[1] 1
[1] 0.1
[1] 0.02
```

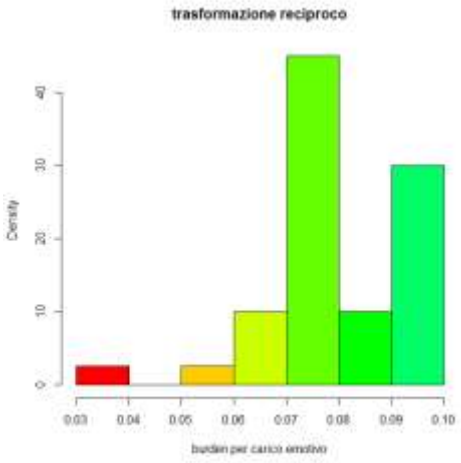
Poiché:

```
1/0
[1] Inf
```

, dovremo nuovamente usare la variabile `burden_emotivo_corretto` in cui è stato eliminato il punteggio 0:

```
burden_emotivo_reciproco<-1/(burden_emotivo_corretto)
Skew(burden_emotivo_reciproco)
[1] -0.4599006
```

La forte asimmetria positiva ha lasciato il posto a una meno intensa asimmetria negativa.

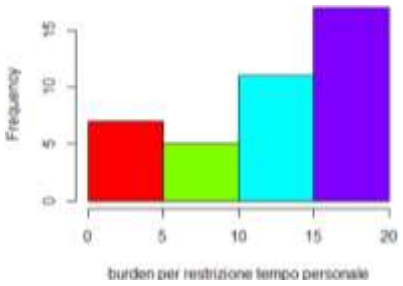


4.6.4 Trasformazione esponenziale: $X_t = X_i^n$

Per ridurre l'asimmetria **negativa**, si ricorre in genere a trasformazioni **esponenziali**, perlopiù elevando alla seconda la variabile da trasformare; più raramente si elevano alla terza, con esponenti > 3 o < 2, o negativi (v. §4.6.5).

Vediamo il burden dovuto alla restrizione del tempo personale dei caregiver:

```
Skew(a$CBI_burden_restrizione_tempo)
[1] -0.7002986
```



L'asimmetria è negativa, di moderata intensità.

Vediamo l'effetto di una elevazione alla seconda e alla terza della variabile originaria:

```
burden_tempo_esp2<-
   a$CBI_burden_restrizione_tempo^2

Skew(burden_tempo_esp2)
[1] -0.185458
burden_tempo_esp3<-
   a$CBI_burden_restrizione_tempo^3

Skew(burden_tempo_esp3)
[1] 0.1589908
```

L'elevazione alla seconda riduce l'asimmetria negativa, quella alla terza la sposta addirittura nel campo positivo.

Attenzione: le trasformazioni logaritmica, quadratica, nel reciproco possono ridurre l'asimmetria negativa sui dati grezzi **invertiti**: per invertili, ogni dato viene sottratto al valore più alto $+ 1$ (in questo modo il valore più alto invertito non sarà $= 0$, che è un problema per la trasformazione logaritmica e il reciproco). Per esempio:

```
burden_emotivo_inverso<-18-(burden_emotivo_corretto)
```

È importante ricordarsi di aver eseguito questa operazione prima di interpretare i dati, o invertirli nuovamente al momento di commentarli!

4.6.5 Scegliere la trasformazione non lineare migliore

Attenzione: il contenuto di questo paragrafo richiede elementi di teoria che faremo in Tecniche di Analisi di dati II. Il background teorico non sarà, quindi, oggetto dell'esame di TAD I, ma la funzione in R è molto semplice, e potete già usarla.

Quelle descritte nei paragrafi precedenti sono solo **alcune** di molte trasformazioni non lineari, applicabili per casi diversi: trasformazione angolare (arcoseno, seno inverso, seno inverso iperbolico: per proporzioni e percentuali), tangente iperbolica inversa (per distribuzioni da -1 a $+1$), log-log e log-complementare (per analisi di sopravvivenza) , probit (affine all'angolare), ed altre ancora. Si pone perciò il problema di individuare la trasformazione non lineare che meglio normalizza la distribuzione, dove la consuetudine adottata in ricerche similari non è d'aiuto.

Un metodo per individuare quale sia la migliore trasformazione non lineare di una distribuzione è quello di **Box-Cox**, (Box e Cox, 1964), che valuta la verosimiglianza di diversi esponenti positivi e negativi da applicare alla X da trasformare: con un metodo iterativo che accenneremo nella regressione logistica (metodo della massima verosimiglianza[22]) viene individuato il parametro lambda (λ) il cui valore corrisponde all'esponente cui elevare la variabile per ottenere la migliore normalizzazione possibile: $X_T = X^{\lambda}$**.** Il valore lambda è inserito in un **intervallo di**

[22] Per la precisione: è il valore che massimizza la funzione di verosimiglianza – log-likelihood function LL)

fiducia CI (capitolo 5), all'interno del quale si sceglie l'esponente che garantisce la trasformazione migliore. Di solito si sceglie il **valore lambda intero più prossimo al valore lambda ottenuto**, corrispondente a una tra le trasformazioni più comuni, ma si può anche usare il valore lambda esatto:

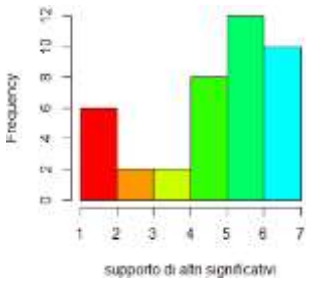
λ = -3: $\frac{1}{X^3}$	λ = -2: $\frac{1}{X^2}$	λ = -1: $\frac{1}{X}$	λ = -.5: $\frac{1}{\sqrt[2]{X}}$	λ = $-1/3$: $\frac{1}{\sqrt[3]{\sqrt{X}}}$	λ =0: $\log X$	λ =.5: $\sqrt[2]{\sqrt{X}}$	λ =1/3: $\sqrt[3]{X}$	λ =1: X	λ =2: X^2	λ =3: X^3
Asimmetria positiva					Asimmetria fortemente positiva	Asimmetria positiva	Asimmetria debolmente positiva	Trasformazione lineare: non serve trasformare X	Asimmetria negativa	Asimmetria fortemente negativa

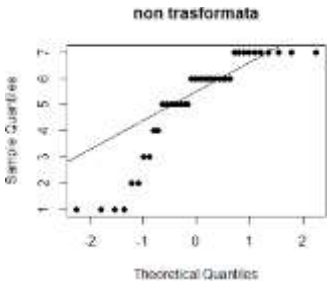
Se i dati da trasformare contengono 0, va aggiunta una costante a tutti i dati (0.5 o 1).

In R usiamo `boxcox(variabile~1, plotit=TRUE)` del package di base `MASS`. L'elemento `~1` va usato perché l'oggetto della funzione è un modello lineare ($Y \sim X$), che nel caso di distribuzioni univariate è in realtà un modello **nullo**, contenente la sola intercetta (`~1`: Capitolo 9). L'argomento `plotit`, di default `=TRUE`, produce un grafico che consente di individuare il lambda ottimale all'intero di un range di valori ugualmente verosimili (intervallo di fiducia, Capitolo 5). Di default, il range dei valori λ va da -2 a $+2$; se necessario, si può ampliare con `seq(da= limite negativo, a= limite positivo, 1/10)`, in cui $1/10$ indica il passo del range.

Vediamo un esempio con la distribuzione `$zimet_altri_significativi`:

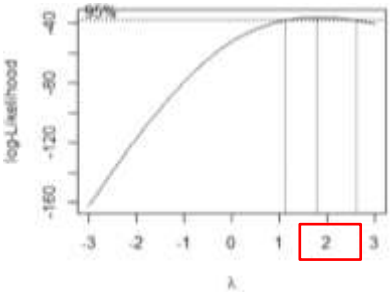
```
Skew(a$Zimet_supporto_altri_significativi);
Kurt(a$Zimet_supporto_altri_significativi)
[1] -0.9876344
[1] -0.243172
```





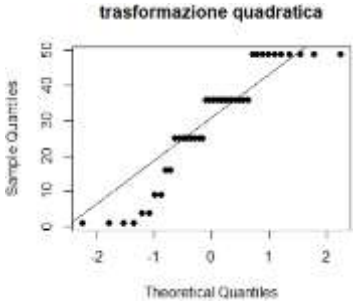
Dovremo ridurre un'asimmetria negativa, quindi `boxcox` ci proporrà un esponente positivo (trasformazione esponenziale con esponente intero).

```
altri<-boxcox(a$Zimet_supporto_altri_significativi~1, lambda =
   seq(-3, 3, 1/10))
altri$x[which.max(altri$y)]
[1] 1.787879
```



Il lambda ottimale è prossimo a 2, e nell'intervallo di fiducia non sono compresi altri valori interi: possiamo quindi usare una trasformazione esponenziale quadratica (o usare come esponente $\lambda = 1.788$). Vediamo:

```
qqnorm(a$Zimet_supporto_altri_significativi^2, main="trasformazione
   quadratica", pch=1);
   qqline(a$Zimet_supporto_altri_significativi^2)
```



Effettivamente, il QQ plot conferma un avvicinamento alla normalità, anche se l'ideale è sempre lontano.

4.7 Centrare le distribuzioni: trasformazioni lineari

In particolari casi (ne abbiamo visto un esempio nel paragrafo precedente) può essere utile cambiare la scala o l'origine delle distribuzioni (**traslazione**), in genere per semplificare i calcoli o la leggibilità dei dati: si usa allora una trasformazione **lineare**, che può essere **moltiplicativa** (ogni dato è moltiplicato per un valore costante, o costante di conversione) o **sommativa** (a ogni dato è sottratto un valore costante), o una combinazione di moltiplicativa e additiva. In qualsiasi caso, una trasformazione lineare **non altera la forma della distribuzione** originaria. Vedremo esempi di una trasformazione lineare anche nella regressione lineare (capitolo 9), quando discuteremo dell'opportunità di centrare le variabili indipendenti di un modello.

Una particolare e importantissima forma di traslazione, che in realtà **oltre a spostare il centro della distribuzione cambia anche l'unità di misura**, è la **standardizzazione di una variabile**. Standardizzare una variabile significa trasformarla in modo tale che, qualunque siano l'unità di misura e il range dei punteggi grezzi della variabile, la distribuzione trasformata avrà media $\bar{X} = 0$ e deviazione standard $s = 1$. Per prima cosa, si **centrano i dati attorno a zero**, sottraendo a ciascun dato la media della distribuzione; in questo modo, la media della nuova distribuzione sarà $= 0$. Poi, si divide ogni dato così centrato per la deviazione standard del campione, ottenendo così che la **deviazione standard della nuova distribuzione sia pari a 1**: la **nuova unità di misura della variabile standardizzata è quindi la deviazione standard**.

I dati trasformati sono chiamati **punteggi z (*z scores*)**:

$$z = \frac{x_i - \bar{X}}{s}$$

Standardizzare una variabile con R è facile: vediamo un esempio, prima con tutti i passaggi del calcolo e poi con la funzione dedicata.

Standardizziamo la variabile `a$STAI_stato`, con $\bar{X} = 51.72$ e $s = 12.62$:

```
mean(a$STAI_stato); sd(a$STAI_stato)
[1] 51.725
[1] 12.61661
```

Creiamo la distribuzione **`ansia_z`**: prima **centriamo** la variabile sulla media **sottraendo a ogni dato la media della distribuzione**:

```
ansia_z<-a$STAI_stato-mean(a$STAI_stato)
round(mean(ansia_z),2);sd(ansia_z)
[1] 0
[1] 12.61661
```

La distribuzione `ansia_z` ha $\bar{x} = 0$ (centrata sulla media), ma ancora la vecchia deviazione standard; completiamo la standardizzazione **rapportando i dati centrati alla deviazione standard** della variabile `$STAI_stato`, **cambiando così l'unità di misura della variabile**:

```
ansia_z<-ansia_z/sd(attaccamento$STAI_stato)
round(mean(ansia_z),2);sd(ansia_z)
[1] 0
[1] 1
```

Confrontiamo le distribuzioni grezza e standardizzata dell'ansia: **`Desc(distribuzione)`** ha tutto quello che serve:

```
Desc(a$STAI_stato)
```

length	n	NAs	unique	0s	mean	meanCI'
40	40	0	23	0	51.73	47.6
	100.0%	0.0%		0.0%		55.76

.05	.10	.25	median	.75	.90	.95
32.70	35.90	43.75	52.00	61.50	66.10	68.05

range	sd	vcoef	mad	IQR	skew	kurt
48.00	12.62	0.24	13.34	17.75	-0.23	-1.07

```
Desc(ansia_z, digits = 2)
```

length	n	NAs	unique	0s	mean	meanCI'
40	40	0	23	0	0.00	-0.32
	100.0%	0.0%		0.0%		0.32

.05	.10	.25	median	.75	.90	.95
-1.51	-1.25	-0.63	0.02	0.77	1.14	1.29

range	sd	vcoef	mad	IQR	skew	kurt
3.80	1.00	-1.64e+16	1.06	1.41	-0.23	-1.07

Notate *skewness* e curtosi della distribuzione grezza e di quella standardizzata: la standardizzazione, essendo una **trasformazione lineare** (a ogni dato è sottratta la stessa quantità e il risultato è diviso per la stessa quantità), ha cambiato la scala dei dati **lasciando inalterata la forma** della loro distribuzione.
R ha la velocissima funzione **`scale(variabile)`** che elimina ogni passaggio del calcolo: i suoi argomenti sono `center = TRUE` (default) e `scale = TRUE` (default), che seguono esattamente la logica del calcolo appena vista. Se lasciati entrambi `TRUE`, la trasformazione della variabile è in punti z – altrimenti, si ottengono altri tipi di trasformazioni:

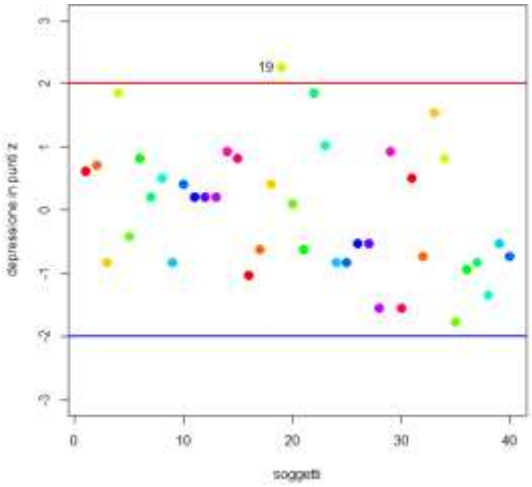
```
ansia_z<-scale(a$STAI_stato)
```

Ora la nuova unità di misura è la deviazione standard, per cui il **livello di ansia** dei soggetti **è interpretato secondo il numero di deviazioni standard sopra o sotto la media**.

La trasformazione in punti z rende molto facile individuare i soggetti **outlier univariati**[23]: sono definiti outlier univariati i soggetti il cui punteggio si discosta di **almeno $|2|$ sd dalla media** (per alcuni autori, |3|), e quindi i soggetti con $z \geq 2$ o $z \leq -2$. **Se** la distribuzione segue un andamento approssimativamente normale, per proprietà della distribuzione normale il caso outlier cadrà nel 2.5% superiore o inferiore della totalità dei casi. L'individuazione degli outlier è un passaggio importante della descrizione dei dati, sia dal punto interpretativo, sia per la valutazione del fit della distribuzione alla forma teorica attesa (perlopiù, quella normale), ovvero a un **modello**. Infatti, l'eliminazione degli outlier della distribuzione può facilitare l'adeguamento della distribuzione dei dati e della distribuzione degli errori alla normale: nel primo caso, il soggetto outlier determina una coda "anomala" rispetto alla distribuzione normale. Nel secondo, gli **outlier rappresentano i casi per cui il modello compie gli errori più grandi**: eliminandoli, la capacità del modello di descrivere la realtà, cioè il suo fit, dovrebbe migliorare.

Vediamo per esempio la distribuzione standardizzata della depressione, **trasformando in punti z i punteggi al test BDI II**;

```
depressione_z<-scale(a$BDI_II_depressione)
 plot(depressione_z, col=rainbow(15), pch=19, cex=1.5,
 ylim=c(-3,3), xlab="soggetti", ylab="depressione in
 punti z")
abline(h=2, col="red", lwd=2)
abline(h=-2, col="blue", lwd=2)
identify(depressione_z)
```



Il soggetto 19 ha un punteggio di depressione di oltre due deviazioni standard superiore alla media: probabilmente, è un outlier.

Se il campione è sufficientemente grande e la distribuzione è fortemente asimmetrica, è possibile sostituire l'uso dei punti z con la **disuguaglianza di Chebyshev**, che **calcola la probabilità di un dato, distante k deviazioni standard dalla media della popolazione, di appartenere alla popolazione** da cui abbiamo estratto il campione.

$$P < \frac{1}{k^2} \times 100$$

Se questa probabilità è molto piccola, probabilmente il soggetto cui si riferisce il dato è un outlier, appartenente a una popolazione diversa dal resto del campione. Naturalmente, la disuguaglianza prevede che siano note media e sd della

[23] Tratteremo nel capitolo 9 gli outlier **bivariati**

popolazione, in realtà non sempre disponibili, ed **è meno potente** nell'individuazione degli outliers rispetto ai punti z, il che limita la sua utilità come *detector*.

Manteniamo come esempio il Beck II, che nella validazione italiana di Montano e Flebus (2006) presenta una media normativa $\mu = 8.10$ con $\sigma = 6.43$. Qual è la probabilità che il caregiver 19, individuato dai punti z come outlier, appartenga alla popolazione normativa? Il suo punteggio grezzo è:

```
a$BDI_II_depressione[19]
[1] 39
```

Calcoliamo k (sd del punteggio 39 dalla media) e inseriamolo nella formula della disuguaglianza di Chebyshev:

```
k<-(39-8.1)/6.43
p<-(1/(k^2))*100
p
[1] 4.33017
```

Il caso 19 ha meno del 4.3% di probabilità di appartenere a una popolazione normativa senza sintomi depressivi: la probabilità è decisamente piccola, confermando la diagnostica dei punti z.

L'uso dei punti z è correntemente usato nella ricerca, ma non è esente da critiche, soprattutto legate a **campioni piccoli**: infatti, il **valore massimo teorico $|z|$ dipende da N**, in quanto il limite massimo raggiungibile da un punteggio z è dato da $\frac{N-1}{\sqrt{N}}$ (Shiffler, 1988). Quindi, con campioni di $N = 10$, $N = 30$ e $N = 50$ avremmo dei limiti massimi pari a:

```
(10-1)/sqrt(10); (30-1)/sqrt(30); (50-1)/sqrt(50)
[1] 2.84605
[1] 5.294651
[1] 6.929646
```

Lo stesso valore x_i, quindi, In un campione di 10 soggetti non potrà mai rispettare il criterio $z \geq |3|$ ed essere indicato come outlier, mentre potrebbe esserlo solo aumentando la numerosità, a parità di media e sd del campione.

Una volta individuato il caso / i casi presumibilmente outliers, questi possono:

a) essere eliminati dal campione, se si è ragionevolmente certi che rappresentino un errore di campionamento (per esempio, un paziente con sintomatologia clinicamente rilevante, ma non diagnosticata, inserito in un campione non clinico);
b) mantenerli nel campione usando tecniche di analisi che "depotenzino" la capacità dell'outlier di distorcere il valore medio della distribuzione: si possono usare la mediana invece della media, test inferenziali basati sui ranghi (test non parametrici) invece di test inferenziali basati sulla media; si può sostituire la media aritmetica con la media *trimmed* (l'abbiamo trovata nel §3.2.4) o con la media ***winsorized***, in cui i valori estremi sono sostituiti da valori meno estremi (la rivedremo in Tecniche di analisi di dati II, §10.1.3).

4.6 Grafici, illusioni e distorsioni

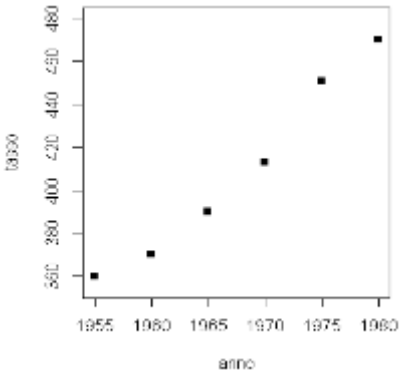
Come dichiarato all'inizio di questo capitolo, la presentazione dei dati mediante grafici è estremamente utile, e anzi irrinunciabile per comprendere e comunicare agli altri quanto accaduto nella propria ricerca. Però, proprio la loro accattivante chiarezza può portare l'incauto lettore (o creatore!) dei grafici a equivoci interpretativi. Ricordiamo qui i peggiori.

Quando la **scala dell'ordinata Y è eccessivamente compressa**, l'informazione del grafico viene distorta. Vediamo un esempio con i dati proposti da Everitt (2001), relativi ai tassi di mortalità per cancro al seno registrati dagli anni Cinquanta

agli Anni Settanta negli Stati Uniti: **se non specifichiamo alcun limite ai valori dell'ordinata**, il tasso espresso dal plot esprime un'ascesa vertiginosa.

Usiamo R:

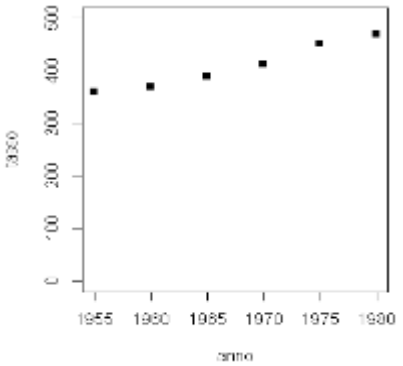
```
anno<-c(1955,1960,1965,1970, 1975, 1980)
tasso<-c(360, 370, 390 ,413, 451, 470)
plot(anno, tasso, ylim=c(355,480), pch=15)
```



Quando non sono specificati da noi, i limiti di Y calcolati da R sono 360 e 480: il range è quindi pari a 120.

Cambiamo i limiti di Y usando l'argomento `ylim= c(limite inferiore, limite superiore)`: facciamo **partire Y da 0**, allargandone il range, e vediamo come cambia l'aspetto del grafico relativo agli stessi dati.

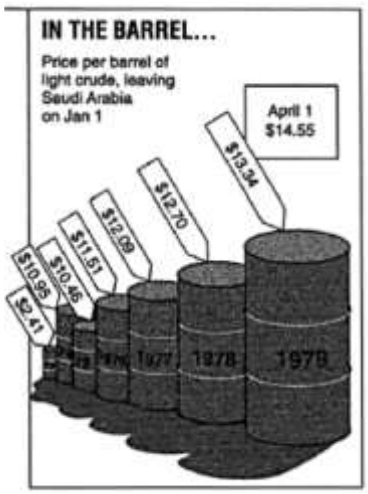
```
plot(anno, tasso, ylim=c(0,500), connect=TRUE, pch=15)
```

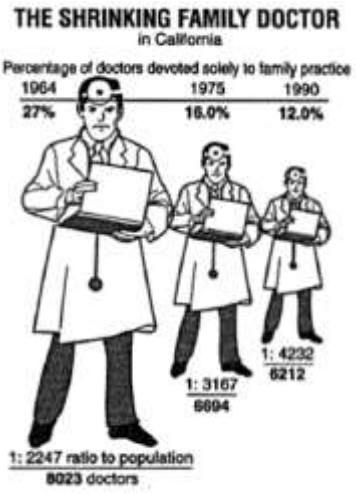


Le conclusioni " a occhio" sarebbero diverse, no?

Un'altra distorsione comune introdotta nei grafici o nei diagrammi usati negli articoli di divulgazione avviene quando **entrambe le dimensioni di una figura bidimensionale sono variate simultaneamente** in risposta ai **cambiamenti di una singola variabile**. Tufte (1983) chiama questa distorsione "fattore di menzogna" (**lie factor**) di un grafico, definendolo come: $lie\ factor = \frac{dimensione\ dell'effetto\ mostrato\ nel\ grafico}{dimensione\ dell'effetto\ nei\ dati}$

Le due figure sottostanti sono usate da Tofte (1983) come esempio di *lie factor*:

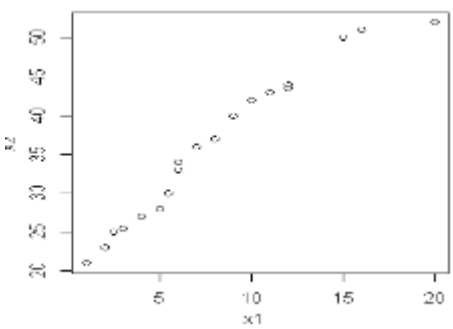




Se il *lie factor* è prossimo a 1, il grafico rappresenta in maniera realistica e accurata i numeri sottostanti; quello della figura con i barili di petrolio è $= 9.4$, dato che l'incremento reale nella misura, pari al **454**%, è raffigurato come se fosse un incremento del **4280**%; quello della figura con i medici è $= 2.8$.

Anche un semplice plot può indurre equivoci. Vedremo nel capitolo 8 che per rappresentare la relazione tra due variabili continue si usa un plot (grafico a dispersione) in cui i valori di X_1 sono in ascissa e quelli di X_2 in ordinata. Se tra due variabili esiste una relazione positiva, al crescere dell'una cresce anche l'altra: tanto più è forte la relazione (il valore del coefficiente di correlazione tende a 1), tanto più i punti, che rappresentano i valori X_1X_2 di ogni soggetto, **tendono a disporsi lungo una retta** immaginaria.

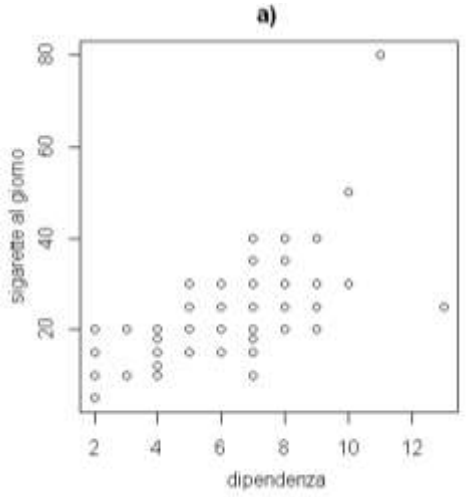
Per esempio, il grafico a fianco rappresenta la relazione di due variabili il cui coefficiente di correlazione è quasi 1: $r = .98$



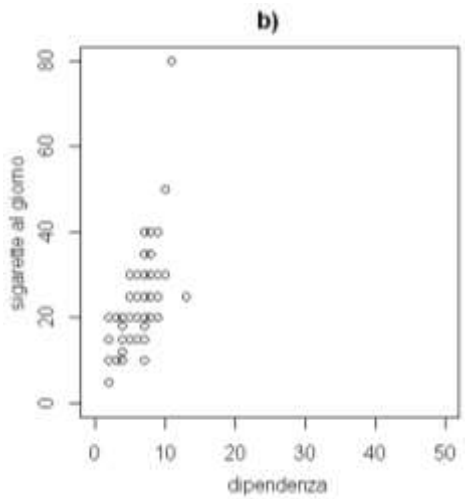

```
x1<-c(1,2,2.5,3,4,5,5.5,6,6,6,7,8,9,10,11,12,12,15,16,20)
x2<-c(21,23,25,25.5,27,28,30,33,34,34,36,37,40,42,43,43.5,44,50,51,52)
```

Ebbene, il **giudizio visivo sulla forza della correlazione tra le variabili può essere distorto allargando l'area** in cui sono raffigurati i punti.

.Nel capitolo 6 useremo il dataframe `fumo`, che contiene dati relativi a un campione di pazienti che si sono rivolti a un Centro antifumo per smettere di fumare. Al loro ingresso in trattamento, la relazione positiva tra il numero di sigarette fumate e la loro dipendenza da nicotina è discreta: $r = .628$. Nel grafico a) la relazione tra le due variabili è presentata senza manipolazioni dell'area del grafico: i limiti di Y e di X sono scelti da R. Nel grafico b), la relazioni tra le due variabili è mostrata **fissando i limiti di Y e di X (`ylim` e `xlim`) in modo da allargare l'area**. All'impatto visivo, la relazione positiva mostrata nel grafico b) **sembra più forte:**




```
plot(fumo$Fagerstrom,fumo$sigarette, xlab="dipendenza", ylab="sigarette al giorno", main="a)")
```




```
plot(fumo$Fagerstrom,fumo$sigarette, xlab="dipendenza", ylab="sigarette al giorno", main="b)", ylim=c(0,80),xlim=c(0,50))
```

Capitolo 5

Prevedere eventi: dalla frequenza alla probabilità del fenomeno

*"Probabilities are **very slippery** things"*
K. Pearson, lettera a Edgeworth

"As to what probability is and how it is connected with statistics, there has seldom been such complete disagreement and breakdown of communication since the Tower of Babel"
Savage, 1954

Un altro modo per ragionare sulle distribuzioni di frequenza non è nei termini di quanto spesso un fenomeno, per esempio un punteggio, si verifica, ma di **quanto è verosimile** che tale punteggio si verifichi, ovvero di quanto sia **probabile**. Una variabile può assumere diverse modalità, e ogni modalità ha una certa probabilità di manifestarsi: a ogni modalità è quindi associabile la sua probabilità, come, in una distribuzione di frequenza, a ogni modalità è associata la sua frequenza. La probabilità di un evento è quindi interpretabile come **facilità dell'evento di realizzarsi.**
Il legame tra calcolo della probabilità e le statistiche descrittive si realizza attraverso le **variabili casuali o variabili aleatorie o variabili stocastiche**: tali variabili sono quantità il cui valore dipende dall'esito di un esperimento casuale[24].
Un **esperimento casuale** è ogni atto o processo la cui singola esecuzione (detta ***prova:*** l'esperimento è quindi ripetibile) dà luogo ad un ***risultato non prevedibile***: tirare una moneta, lanciare due dadi...; al contrario, negli **esperimenti deterministici**, se le condizioni sperimentali non mutano e sono assenti altre circostanze perturbatrici (covariate), ogni singola prova dà lo stesso risultato. Tuttavia, se **ripetiamo** molte volte un **esperimento casuale** è possibile rilevare **regolarità** nei risultati, ovvero nella distribuzione di un certo fenomeno statistico (esempio tipico: buttare in aria una moneta e contare quante volte esce testa: si vedano gli esperimenti di **Buffon** e Pearson nel §5.1), a lungo - molto a lungo...- andare si può concludere che le frequenze relative osservate con cui si realizzano di valori delle distribuzioni di frequenza siano assimilabili alle probabilità con cui questi valori possono realizzarsi.

L'insieme dei valori assumibili da un fenomeno statistico e le probabilità che gli attribuiamo costituiscono un modello, cioè una descrizione di quello che può verificarsi e delle probabilità con cui aspettiamo che questi eventi si verifichino. **Le variabili aleatorie sono proprio tali modelli**.

5.1 Una breve storia naturale della probabilità

Storicamente, giungere a una definizione condivisa di cosa s'intenda per probabilità è stato faticoso.
La **definizione classica** è stata data da Pascal (1623-1662), che si è occupato di probabilità su sollecitazione dell'amico cavaliere de Méré, appassionato di gioco d'azzardo: "La **probabilità** di un evento è il **rapporto tra il numero dei casi favorevoli all'evento e il numero dei casi possibili**, purché questi siano **tutti ugualmente probabili**". Questa **definizione** è però **circolare**: per definire la probabilità occorre sapere preliminarmente che cosa significa che due casi sono ugualmente probabili, cioè sapere già cos'è la probabilità. Inoltre, essa è applicabile solo se gli eventi hanno tutti la **medesima probabilità**, e **presuppone un numero finito di casi possibili**: per esempio, se volessimo sapere qual è la probabilità di un estrarre a caso un numero naturale **pari** (evento favorevole, insieme infinito…) dall'insieme dei

[24] Per i precisini: una variabile casuale X è una ***funzione*** definita nello spazio campionario S che associa un numero reale $X(e) = x$ a ogni evento elementare di S

numeri naturali (casi possibili, insieme altrettanto infinito), ci troveremmo di fronte a $p\left(\frac{n.pari}{n.naturali}\right) = \frac{\infty}{\infty}$, ben diverso dalla risposta "ingenua" che sarebbe "un mezzo".

L'insoddisfazione per la definizione classica portò a rovesciare questo concetto, **costruendo la probabilità sulle frequenze stesse**, invece di considerarla come qualcosa di esterno col quale confrontare le frequenze. La **legge empirica del caso** o **definizione frequentista della probabilità** afferma che, in una successione di prove fatte nelle stesse condizioni, la frequenza di un evento si avvicina alla probabilità dell'evento stesso e l'approssimazione tende a migliorare con l'aumentare del numero delle prove → la **probabilità è il valore costante intorno al quale tende a stabilizzarsi la frequenza relativa di un evento al crescere del numero di prove di un dato esperimento**, ovvero il limite cui tende la frequenza relativa di successi man mano che cresce indefinitamente il numero di prove. Anche questa definizione non è esente da problemi: le frequenze non costituiscono una successione numerica data mediante legge, ma sono numeri rilevati sperimentalmente, per i quali il concetto di limite non è chiaro. Inoltre, si deve prendere in considerazione, evidentemente, una successione di prove fatte nelle stesse condizioni e ripetute indefinitamente, e ciò restringe l'applicabilità della definizione a situazioni ben delimitate, come i lanci successivi di un dado. Per fare un esempio storico di mirabile pazienza, Buffon (1708-1788) tirò 4040 volte una moneta ottenendo 2048 volte testa: la probabilità di ottenere testa è quindi data dal rapporto tra la frequenza di testa (2048) e il numero totale di lanci (4040): $2048 \div 4040 = 0.0506$. Però, l'ancor più paziente Pearson (1857-1936) ha tirato tirò per 24000 volte una moneta ottenendo 12012 volte testa: la sua probabilità di ottenere testa è quindi diversa, dato che $10212 \div 24000 = 0.5005$.

Notate che l'esperimento di Pearson comprende sei volte più lanci di quello di Buffon: dato che l'approssimazione tende a migliorare aumentando il numero di prove, secondo definizione, la sua stima della probabilità di ottenere testa è più accurata di quella di Buffon: in effetti, se consideriamo la probabilità come il rapporto tra l'evento atteso o favorevole (testa) rispetto al numero di eventi ugualmente possibili (testa e croce), **la probabilità di ottenere testa è $\boldsymbol{P(testa) = 0.50}$**. Tuttavia, quando settemila lanci di dodici dadi, tirati da un impiegato dello University College su incarico di un collaboratore di Pearson, non hanno ottenuto il valore teoricamente più probabile previsto, **Pearson ha rifiutato il risultato empirico** e avviato un ampio dibattito sulle discrepanze tra empiria e teoria, da cui il suo frustrato commento citato a inizio capitolo.

La migliore approssimazione ottenuta dalle monete di Pearson non è stata un caso, ma una regolarità definita **legge dei grandi numeri** (la dobbiamo a Bernoulli): **per un numero tendente all'infinito di ripetizioni identiche di un esperimento aleatorio, la probabilità di un evento tende a coincidere con la sua frequenza**. In altre parole[25], possiamo dire che **la media calcolata di una sequenza di prove, indipendenti ed equiprobabili, di una variabile aleatoria, ripetute per un numero di volte tendente a infinito, è sufficientemente vicina alla media vera, ovvero quella calcolabile teoricamente**.

Al di là delle definizioni divergenti, c'è accordo sul **calcolo** delle probabilità: la prima vera formalizzazione della teoria del calcolo delle probabilità è dovuta a **Kolmogorov** (1903-1987), cui si devono i tre assiomi fondamentali della probabilità[26] (**approccio assiomatico**).

$$P(A) = \frac{P(A)}{P(\neg A)}$$

La probabilità dell'evento A, $P(A)$, sommata alla probabilità che si verifichi l'evento $non\ A$, $P(\neg A)$, definisce **tutti** i possibili esiti della prova: **probabilità dell'evento certo $\boldsymbol{P = 1}$**. La probabilità di un evento **varia da 0** (l'evento non può in alcun caso verificarsi) **a 1** (**evento certo**); se vi risulta più semplice interpretarle in termini **percentuali** ("una probabilità del 10% di verificarsi", invece di "una probabilità pari a 0.1 di verificarsi) va bene lo stesso – ma ricordate che una probabilità espressa come percentuale è una probabilità trasformata dall'originale range 0-1.

[25] Parole piuttosto superficiali, in realtà: esistono una legge forte e una legge debole dei grandi numeri, con relative dimostrazioni, ma le lasciamo agli amanti della materia.

[26] Se l'evento *A* è certo *P*(*A*) = 1; *P*(*A*) ≥ 0 per ogni evento *A*; se *A* e *B* sono due eventi incompatibili *P*(*A* U *B*) = *P*(*A*) + *P*(*B*).

Due eventi si dicono **mutualmente escludentisi** (**incompatibili**), se il **verificarsi dell'uno non consente il** verificarsi dell'altro: ad esempio, tirando un solo dado, il verificarsi della faccia "cinque" impedisce il verificarsi delle altre cinque possibilità. La probabilità del verificarsi di due eventi mutualmente escludentisi è uguale alla **somma** della probabilità di verificarsi dei singoli eventi: **teorema della probabilità totale** o **regola della somma**→ $\boldsymbol{P(A\ o\ B) = P(A) + P(B) = 1}$, ovvero $P(A \cup B) = P(A) + P(B) - \boldsymbol{P(A \cap B)}$. Quindi, la probabilità che si verifichi almeno uno tra gli eventi A e B $P(A \cup B)$ è data dalla somma della probabilità dell'evento A e dell'evento B, meno **la probabilità che i due eventi si verifichino insieme** ($\boldsymbol{P(A \cap B)}$: per gli eventi incompatibili, naturalmente, $P(A \cap B) = 0$. La probabilità che si verifichi almeno uno tra A e B è =1, cioè è un evento certo che uno degli eventi si verifichi.
Per esempio, tirando un solo dado, il verificarsi dell'evento faccia "1" O faccia "3" O faccia "5" (cioè vedere una faccia dispari) è uguale alla somma della probabilità di "1", "3" e "5", ovvero:

$$P(1 \cup 3 \cup 5) = {}^{1}/_{6} + {}^{1}/_{6} + {}^{1}/_{6} - 0 = {}^{3}/_{6} = .\mathbf{50} \rightarrow 50\%$$

Come si declina la regola della somma nel caso di eventi **compatibili,** invece, in cui il verificarsi dell'uno **non esclude** il verificarsi dell'altro? In maniera del tutto simile, ma poiché $P(A \cap B) \neq 0$, non potremo ignorarla. Per esempio, tirando un solo dado, qual è la probabilità di ottenere "faccia dispari" (A: "1", "3", "5") **o** un multiplo di 3 (B: "3")? C'è un evento, la faccia "tre", che soddisfa contemporaneamente A – dispari e B – multiplo di 3. Quindi,

$$P(dispari \cup multiplo_3) = {}^{3}/_{6} + {}^{1}/_{6} - {}^{1}/_{6} = .\mathbf{50} \rightarrow 50\%$$

Due eventi si definiscono **indipendenti** se il verificarsi dell'uno **non influenza il verificarsi dell'altro**: tirando due dadi, il fatto che uno segni la faccia "cinque" non influenza il risultato dell'altro dado. La probabilità del verificarsi **contemporaneo o in successione** di due eventi **indipendenti** è uguale al **prodotto** della probabilità di verificarsi dei singoli eventi: **regola del prodotto (o delle probabilità composte)**→ $\boldsymbol{P(A\ e\ B) = P(A)\ x\ P(B)}$, ovvero $P(A \cap B) = P(A) \times P(B)$. Per esempio, tirando due dadi, il verificarsi dell'evento "somma uguale a due" (dado 1: "faccia uno" **e** dado 2: "faccia uno"), è uguale al prodotto della probabilità di avere "uno" con il primo dado **e** "uno" con il secondo, ovvero:

$$P(\Sigma_2) = \frac{1}{6} \times \frac{1}{6} = \frac{\mathbf{1}}{\mathbf{36}} = .\mathbf{028}$$

L'evento favorevole è 1 su 36 eventi **possibili, ovvero una delle 36 possibili combinazioni delle sei facce di due dadi**; nell'Appendice I trovate un breve ripasso sul **calcolo combinatorio**, che è certamente utile, ma piuttosto fuori tema per il nostro programma, se vi serve rinfrescare la memoria su combinazioni, disposizioni e permutazioni.

Alcuni eventi sono **dipendenti**: il **verificarsi del primo evento altera la probabilità di verificarsi dei successivi**. Ne è un esempio l'estrazione dei numeri del lotto o della tombola, in cui primo numero estratto ha probabilità $1/90$; il secondo, poiché il bussolotto non viene rimesso con gli altri (**estrazione senza rimpiazzo**), ha probabilità $1/89$; il terzo ha probabilità $1/88$, e così via. Quindi, se il l'evento "estrazione di un numero pari" ha probabilità $45/90$ alla prima estrazione, ha probabilità $45/89$ alla seconda se il primo estratto è dispari e $44/89$ se il primo estratto è pari. **Attenzione**: all'interno della **medesima estrazione** gli eventi **non sono indipendenti**, ma **tra** un'estrazione e l'altra gli eventi **sono indipendenti**! Il fatto che 90 sia estratto sulla ruota di Venezia è del tutto ininfluente sulla probabilità che sia estratto sulla ruota di Cagliari, o che sia estratto la settimana successiva sulla stessa ruota.

Quando gli eventi sono dipendenti, il principio del prodotto cambia: **probabilità condizionata o composta** di B, una volta che si sia verificato l'evento A: $P(A\ e\ B) = P(A) \times P(B|A) = P(B) \times P(A|B)$, **P**, ovvero $P(A \cap B) = P(A) \times P(B|A)$.
Per esempio, la probabilità che i primi due numeri dell'estrazione del lotto siano **entrambi pari** sarà uguale a:

$$P(primi\ 2\ numeri\ pari) = \frac{45}{90} \times \frac{44}{89} = \frac{1980}{8010} = .247$$

Dichiariamo conclusa questa breve presentazione della probabilità, sufficiente per i nostri scopi, e vediamola all'opera negli esperimenti un po' più complessi.

Abbiamo, un po' semplicisticamente, definito una variabile aleatoria come il risultato numerico di un esperimento casuale. A seconda che si **manifestino modalità discrete o continue**, si hanno **distribuzioni di probabilità discrete** o **distribuzione di probabilità continue**.

5.2 Variabili aleatorie discrete

Se una **variabile X può assumere solo valori e interi e positivi** $x_1, x_2, x_3, \dots x_k$, rispettivamente con probabilità $P_1, P_2, P_3, \dots P_k$ (la cui somma è $\sum(P) = 1$), la **variabile X ha una distribuzione di probabilità discreta**.
Pensiamo all'esperimento di **due lanci di una moneta** non truccata, in cui ogni faccia della moneta, testa -T - e croce - C - ha la **stessa probabilità** di manifestarsi in ogni lancio**:** $P(T) = P(C) = 1/2 =.\mathbf{50}$. **L'insieme dei risultati possibili**, cioè lo **spazio campionario** dell'esperimento, è: che esca due volte testa, che al primo lancio esca testa e al secondo croce, che al primo lancio esca croce e al secondo testa, che in entrambi i lanci esca croce. Formalizzando, diremo che lo spazio campionario **Ω** (omega) è:

$$\boldsymbol{spazio\ campionario} = \boldsymbol{\Omega(TT, TC, CT, CC)}$$

Se consideriamo come **evento atteso** la faccia "***T***" e contiamo il numero di T possibili nello spazio campionario, otteniamo $(2, 1, 1, 0)$, cioè associamo a ogni **prova possibile ω** dell'insieme Ω un numero $X(\omega) = x$. Costruiamo così la tabella:

ϖ	TT	TC	CT	CC
P(ϖ)	1/4	1/4	1/4	1/4
X (ϖ)=*x*	**2**	**1**	**1**	**0**

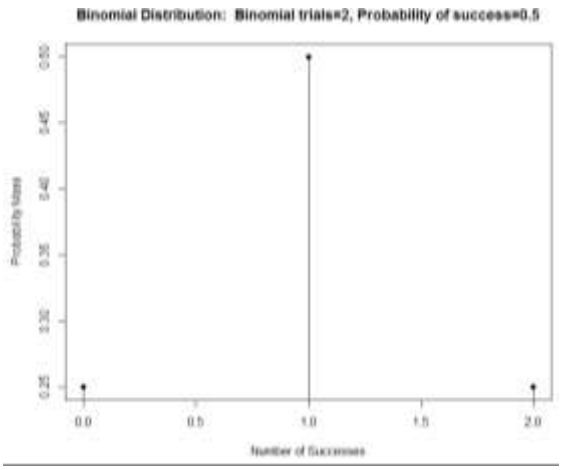
X è dunque il risultato dell'esperimento. La tabella si può sintetizzare **raccogliendo i valori distinti assunti da X** $(\mathbf{0}, \mathbf{1}, \mathbf{2})$ e le **relative probabilità**. Il valore $X(0)$ si presenta una sola volta nella prova $\varpi\ CC$, con probabilità $1/4$; la probabilità di $X(0)$ è quindi $P(0) = \mathbf{1/4}$. Il valore $X(1)$ si presenta due volte, nelle prove $\varpi\ TC$ e CT, ciascuna delle quali con probabilità $1/4$: la probabilità di $X(1)$ è quindi uguale a $P(1) = 1/4 + 1/4 = \mathbf{1/2}$. Infine, il valore $X(2)$ si presenta una sola volta, nella prova $\varpi\ TT$, con probabilità $= 1/4$: la sua probabilità è quindi $P(2) = \mathbf{1/4}$.
Creiamo quindi la nuova tabella:

$X = x$	0	1	2	Totale
	1/4	1/2	1/4	1
$P(X = x)$	**.25**	**.50**	**.25**	**1**

Abbiamo appena creato un **modello matematico che descrive i possibili risultati numerici di un esperimento casuale** con **l'attribuzione delle probabilità** di verificarsi dei singoli eventi: $\sum_{i=1}^{k} \boldsymbol{P(X = x_i) = 1}$

$P(X = x)$ è la **FUNZIONE DI PROBABILITÀ** (o **funzione di massa di probabilità**Errore. Il segnalibro non è definito. o **densità discreta**), cioè la **funzione che assegna a ogni possibile valore di X la probabilità del possibile esito** dell'esperimento casuale.

Una distribuzione di una variabile aleatoria discreta come quella che abbiamo appena costruito, i cui valori possono essere solo dicotomici (testa-croce, giusto-sbagliato, promosso-bocciato) si definisce **distribuzione binomiale** (§4.2.1)Errore. Il segnalibro non è definito.. La **rappresentazione** della **distribuzione** di probabilità binomiale è **un grafico a barre**; quello della nostra X è:



In ascissa abbiamo il numero di eventi attesi, o favorevoli (nel nostro esempio, le frequenze dell'evento Testa), in **ordinata** la **massa di probabilità**[27], ovvero la **probabilità di ciascuno degli eventi**.

Il grafico è stato costruito con `Rcommander` → `Distribuzioni`→ `discrete` → `binomiale` → **`Disegna la distribuzione binomiale`**.

Una volta che sia nota la distribuzione di probabilità di una variabile aleatoria, possiamo calcolare la **probabilità del tipo $P(X \leq x)$,** ovvero la **probabilità di un evento minore o uguale a x_i**: basta calcolare le **probabilità cumulate**, analoghe alle frequenze cumulate, per cui si sommano le probabilità degli eventi inferiori o uguali a x_i.

Questa funzione si chiama **funzione di ripartizione**, perché ripartisce la distribuzione delle probabilità in due parti: fino a x_i – oltre x_i). La sua formula è:

$$P(X \leq x_i) = \sum_{xi \leq x} P(X = x_i)$$

Usiamo come esempio un esperimento casuale che consiste in un **lancio di due dadi** ($trial = 2$; eventi possibili: $6 \times 6 = 36$): ci interessa stabilire la **probabilità delle somme possibili** dei due dati, quindi da evento "2" (1+1) a evento "12": 6+6

$X = x$	$\sum ab = 2$	$\sum ab = 3$	$\sum ab = 4$	$\sum ab = 5$	$\sum ab = 6$	$\sum ab = 7$	$\sum ab = 8$	$\sum ab = 9$	$\sum ab = 10$	$\sum ab = 11$	$\sum ab = 12$	TOT
Combina-zioni dadi	1-1	1-2,2-1	1-3,3-1, 2-2	1-4,4-1, 2-3,3-2	1-5,5-1, 2-4,4-2, 3-3	1-6,6-1 ,2-5,5-2, 3-4,4-3	2-6,6-2, 3-5,5-3, 4-4	3-6,6-3, 4-5,5-4	4-6,6-4,5-5	5-6,6-5	6-6	
f	1	2	3	4	5	6	5	4	3	2	1	36
$P(X = x)$	1/36 .028	2/36 .083	3/36 .111	4/36 .139	5/36 .167	6/36 .194	5/36 .167	4/36 .139	3/36 .111	2/36 .083	1/36 .028	1

La **probabilità** di **ottenere una somma pari o minore a 6** è uguale alla somma delle probabilità dei valori X fino a 6 compreso; facciamoci aiutare da R:

```
cumulata_sei<-1/36+2/36+3/36+4/36+5/36
cumulata_sei
[1] 0.4166667
```

[27] Più precisamente, infatti, si dovrebbe dire "**massa di probabilità**" per variabili discrete e "**densità di probabilità**" per variabili continue.

La probabilità di ottenere una somma uguale o minore a 6 è $P(\sum \leq 6) = .42$

Intuitivamente, per conoscere la **probabilità di ottenere un dato maggiore o uguale a x_i,** cioè $\boldsymbol{P(X > x)}$, si **sottrae alla probabilità totale la probabilità cumulata per i valori inferiori o uguali a $\boldsymbol{x_i - 1}$**. Nel nostro esempio, la probabilità di ottenere una somma **uguale o maggiore di 7** è uguale a

```
cumulata_sette_più<-1-(cumulata_sei)
cumulata_sette_più
[1] 0.5833333
```

Abbiamo costruito il grafico della più semplice delle distribuzioni di probabilità discrete, quella binomiale: approfondiamola.

5.2.1 Distribuzione di probabilità binomiale o bernoulliana

Il lancio della moneta per N volte, usato per descrivere la funzione di probabilità, è un esempio di **distribuzione di probabilità discreta binomiale**: è usata per calcolare la **probabilità di ottenere X successi** (ad esempio Testa) in N **prove** dello stesso esperimento aleatorio, quando:

- **ogni prova ha due soli esiti possibili** e reciprocamente **escludentisi** (testa o croce, giusto o sbagliato),
- le N **prove** sono **indipendenti**;
- la **probabilità di successo** resta **costante** in ogni prova.

L'attribuzione del teorema binomiale è stato oggetto di contesa tra **Newton** (che l'avrebbe scoperto per primo e annunciato per lettera, ma non pubblicato: era uno scienziato tanto brillante quanto una persona decisamente complicata...) e **Leibniz**, più veloce a pubblicarlo: è comunque probabile che le loro scoperte siano state indipendenti.

La **funzione di probabilità binomiale** usa le **combinazioni** (C è il **coefficiente binomiale**: vedi Appendice **I**) per attribuire una probabilità a ciascuna delle modalità della variabile; N è il numero di prove, k è il numero di volte ($\leq N$) in cui si verifica l'evento atteso nella ripetizione delle prove, p è la probabilità che si manifesti l'evento atteso in ogni prova.

Questa è la **formula di Bernoulli** per il coefficiente binomiale:

$$P(X) = C_k^N \times p^k \times (1-p)^{N-k}$$

Comunque, R ci mette a disposizione funzioni dedicate che ci permetteranno di ignorare la formula nel calcolo: le vediamo per la prima volta con questa distribuzione, e le ritroveremo, con poche modifiche, nelle altre distribuzioni discrete e continue (di seguito sono elencate solo quelle che tratteremo):

`d`distribuzione	`p`distribuzione	`q`distribuzione	`r`distribuzione
Calcola la massa di probabilità per le distribuzioni discrete, e la densità per quelle continue `dbinom,dpoisson, dhyper,dnorm,dt,dchi,df`	Calcola la funzione di ripartizione `pbinom,ppoisson, phyper,pnorm,pt,pchi,pf`	Individua il quantile di una distribuzione corrispondente a una probabilità cumulata `qbinom,qpoisson, qhyper,qnorm,qt,qchi,qf`	Genera una distribuzione di valori casuali (aleatoria) che segue la distribuzione di probabilità indicata `rbinom,rpoisson, rhyper,rnorm,rt,rchi,rf`

La funzione **`dbinom(x, size, prob)`** è quella corrispondente alla formula: consente di **calcolare la probabilità di elementi** di una distribuzione binomiale[28]. `x=` è un vettore aleatorio o una singola modalità di una distribuzione binomiale, `size=` esprime il numero di trial/prove, `prob=` esprime la probabilità dell'evento atteso in ogni trial.

[28] **d** sta per **density**, vedi nota precedente.

Nell'esempio della moneta, in cui abbiamo una variabile con valori da 0 a 2, due lanci, e una probabilità di ottenere Testa in ogni lancio pari a 1/2, scriveremo:

```
dbinom(0:2, size = 2, prob = .50)
[1] 0.25 0.50 0.25
```

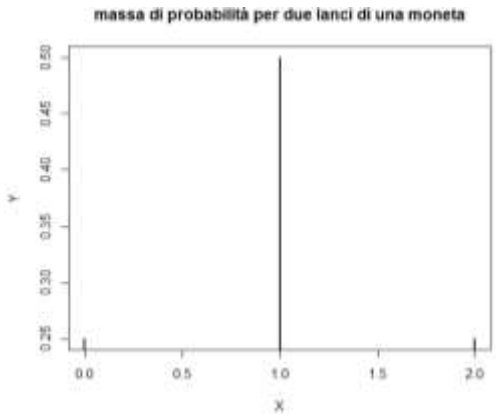
, ottenendo infatti le probabilità che abbiamo calcolato precedentemente per ogni x_i.

Se volessimo invece calcolare qual sarebbe la probabilità di avere un **solo figlio maschio** (evento atteso) **su tre nascite** (trial=3, probabilità=.50 [biologicamente un po' imprecisa, ma possiamo sorvolare]), scriveremmo:

```
dbinom(1,size = 3,prob = .50)
[1] 0.375
```

Se salviamo la massa di probabilità come Y e il vettore $0{:}2$ come X, **possiamo plottarli**, specificando che il tipo di plot è di tipo "h" ("`histogram type`": traccia linee verticali)

```
X<-0:2
Y<-dbinom(0:2, size = 2, prob = .50)
plot(X,Y,type="h", lwd=2, main= "massa di probabilità per due
lanci di una moneta")
```



Per calcolare in R la **funzione di ripartizione** di una distribuzione binomiale, usiamo **`pbinom(q, size, probabilità, lower.tail= TRUE/FALSE)`**, dove **`p`** sta per *partition* e gli argomenti sono: `q=` quantile della variabile aleatoria che identifica la partizione (inferiore o uguale a), `size=` numero di trial, `prob=` probabilità dell'evento atteso in ogni trial. `lower.tail` è un argomento logico: se **TRUE** (**default**) viene considerata la parte inferiore della distribuzione, e quindi è **riportata la probabilità cumulata fino al quantile compreso**; se **FALSE** viene ignorata la parte inferiore e **riportata la probabilità cumulata dal quantile (escluso) in su**.

Nella minuscola distribuzione dei due lanci di una moneta, la probabilità di ottenere testa 0 o 1 volta è uguale a:

```
pbinom(1,size = 2,prob = .5, lower.tail=TRUE)
[1] 0.75
pbinom(1,size = 2,prob = .5, lower.tail = FALSE)
[1] 0.25
```

La probabilità di ottenere il quantile 1 o un quantile inferiore a 1 è $P(\leq 1) = .75$; la probabilità cumulata di ottenere un quantile superiore a 1 è $P(> 1) = 0.25$.

L'inverso della funzione di ripartizione `pbinom` è **`qbinom(p, size, prob, lower.tail= TRUE/FALSE)`**, che restituisce il quantile della distribuzione aleatoria corrispondente alla probabilità cumulata indicata nella funzione. Gli argomenti sono `p=` probabilità cumulata, `size=` numero di trial, `prob=` probabilità dell'evento atteso in ogni trial. L'argomento logico `lower.tail` si usa come in `pbinom`: se `TRUE` (default), viene restituito il quantile corrispondente alla probabilità cumulate $\boldsymbol{P(X \leq x)}$; se `FALSE`, viene restituito il quantile corrispondente alla probabilità cumulata $\boldsymbol{P(X > x)}$.

```
qbinom(p = .75, size =2, prob = .50, lower.tail = TRUE)
[1] 1
qbinom(p = .75, size =2, prob = .50, lower.tail = FALSE)
[1] 0
```

Il quantile corrispondente a una probabilità cumulata fino a .75 è $q(P_{cum} \leq .75) = 1$; il quantile corrispondente a una probabilità cumulata superiore a .75 è $q(P_{cum} > .75\ ) = 0$.

Infine, è possibile generare una variabile aleatoria composta da numeri casuali che assumono una distribuzione binomiale usando la funzione **`rbinom(n, size, prob)`**, in cui `n=` numero di osservazioni, `size=` numero di trial, `prob=` probabilità dell'evento atteso in ogni trial.

Riassumiamo con un esempio realistico: supponiamo che la prova d'esame sia composta da trenta domande a scelta multipla; le opzioni di risposta comprendono per ogni domanda un'alternativa corretta e due alternative errate: si assegna un punto alla risposta corretta e zero punti alla risposta sbagliata. L'evento atteso è "risposta corretta" *versus* "risposta sbagliata", quindi la distribuzione è binomiale. Sappiamo che la probabilità dell'evento atteso è, per ogni domanda, $1/3 = 0.33$, e supponiamo che il docente sia stato tanto bravo da rendere realmente equiprobabili le risposte giuste e sbagliate per chi risponda completamente a caso. Quanto è probabile che uno studente, rispondendo completamente a caso, ottenga la sufficienza, ovvero 18?

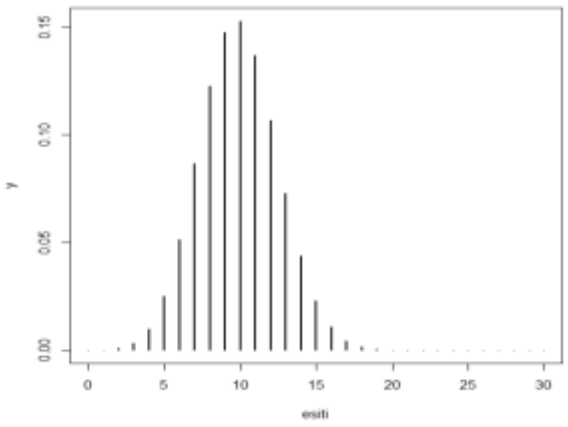
```
dbinom(x = 18, size = 30, prob = .333)
[1] 0.001700154
```

Basarsi solo sul caso per puntare alla sufficienza è una cattiva strategia: si ha meno dell'1% di probabilità di superare l'esame. Se il docente fosse stato più pigro e avesse previsto solo due alternative di risposta per domanda, le chance dello studente random sarebbero aumentate parecchio, pur restando decisamente scarse:

```
dbinom(x = 18, size = 30, prob = .50)
[1] 0.08055309
```

Possiamo costruire **il plot della massa di probabilità** per vedere qual è il risultato più probabile e quali quelli praticamente impossibili: in ascissa costruiamo il vettore degli esiti possibili, da 0 risposte corrette a 30, e in ordinata la massa di probabilità della distribuzione binomiale:

```
esiti<-c(0:30)
y<-dbinom(esiti, 30, .33)
plot(esiti, y, type="h", lwd=2)
```



L'esito più probabile rispondendo a caso è di ottenere un punteggio pari a 10.

Potremmo anche chiedere quale sia la probabilità massima riscontrata nella distribuzione binomiale:

```
max(dbinom(esiti, 30, .33))
[1] 0.1529001
```

La probabilità di rispondere correttamente per caso al massimo a 12 domande (funzione di ripartizione) è uguale a:

```
pbinom(q = 12, size = 30, prob = .33,lower.tail = TRUE)
[1] 0.1562739
```

Invece, la probabilità di rispondere correttamente per caso a 10 o più domande è uguale a:

```
pbinom(q = 9, size = 30, prob = .33,lower.tail = FALSE)
[1] 0.447119
```

ovvero:

```
1-pbinom(q = 9, size = 30, prob = .33,lower.tail = FALSE)
[1] 0.447119
```

5.2.1 Altre distribuzioni di probabilità discrete

Quando il **campione è estratto da una popolazione finita**, **senza possibilità di reinserire** nella popolazione gli elementi via via estratti (come nel caso della tombola o dell'estrazione del lotto), la distribuzione binomiale non è applicabile: **gli eventi non sono indipendenti**, dato che, come descritto nel §4.1, l'estrazione di un elemento del campione condiziona le probabilità di estrazione dei successivi. In questo caso, si utilizza la **distribuzione ipergeometrica**, che misura la **probabilità dei successi / eventi attesi X in un campione di N elementi, presi a caso e senza reinserimento da una popolazione di N element**i. Possiamo usare `dhyper(x=, m=, n=, k=)`, analoga a `dbinom`, per stimare la probabilità di un evento da una distribuzione ipergeometrica. **Attenzione agli argomenti**: `x=` evento atteso, `m=` numero di successi nella popolazione finita, `n=` numero dei non successi nella popolazione, `k=` numerosità del campione estratto.

Per esempio, la probabilità di selezionare casualmente 2 palline bianche (x) da un'urna che contiene 5 palline bianche (m) e cinque palline nere (n), estraendo senza reinserimento 6 palline dall'urna (k) è uguale a:

```
dhyper(x = 2,m = 5,n = 5,k = 6)
[1] 0.2380952
```

Oppure, la probabilità di estrarre sette numeri pari nelle prime 20 estrazioni di una tombola è uguale a.

```
dhyper(x = 7,m = 45,n = 45,k = 20)
[1] 0.06498521
```

Quando il campione estratto è molto piccolo rispetto alla popolazione (meno del 5%), il mancato reinserimento ha scarso effetto nella probabilità di successo di ogni prova: la distribuzione binomiale è allora una buona approssimazione della distribuzione ipergeometrica. Nell'esempio della tombola, sette estrazioni sono poche (il 7.8%) rispetto alla popolazione = 90; in effetti, la distribuzione binomiale stimerebbe una probabilità non troppo dissimile:

```
dbinom(x = 7,size = 20,prob = .50)
[1] 0.07392883
.07392883-.06498521
[1] 0.00894362
```

Ma per 4 numeri pari (4.4% della popolazione), la differenza è ancora più piccola:

```
dhyper(x = 4,m = 45,n = 45,k = 20)
[1] 0.001889814
dbinom(x = 4,size = 20,prob = .50)
[1] 0.004620552
0.004620552- 0.001889814
[1] 0.002730738
```

Come per la binomiale, possiamo calcolare la funzione di ripartizione con `phyperErrore. Il segnalibro non è definito.(q= , m= , n= ,k= , lower.tail= TRUE/FALSE)`, dove `q=` quantile in corrispondenza del quale effettuare la ripartizione, `m=`, `n=` e `k=` danno le stesse indicazioni di `dhyper`. L'inverso della funzione di ripartizione si ottiene con `qhyper(p= ,m= ,n= ,k= ,lower.tail=TRUE/FALSE)` e una distribuzione casuale di numeri che segue la distribuzione ipergeometrica si ottiene con `rhyper(nn= ,m= ,n= ,k= )`: `nn=` indica il numero di osservazioni. Ritroveremo la distribuzione ipergeometrica nel **test della probabilità esatta di Fisher** (§7.2.3).

Per conoscere la probabilità di un dato **numero di eventi attesi / successi per unità di tempo**, se gli eventi sono **indipendenti** e il numero medio di successi per unità di tempo si mantiene costante, si usa la **distribuzione di Poisson** (da Poisson, che la definì a inizio Ottocento: ***legge dei piccoli numeri***). La distribuzione di Poisson può essere usata

come approssimazione della distribuzione binomiale quando il numero di prove N è grande e la probabilità che si verifichi l'evento è piccola (**evento raro**). Un evento è considerato raro se il numero delle prove N è almeno pari a 50, mentre $(N \times p) =< 5$. La distribuzione di Poisson ha avuto un grande spettro di applicazioni: controllo qualità (numero di difetti per unità prodotte), numero di pazienti sofferenti di particolari malattie, terremoti, numero di telefonate a un destinatario sbagliato, numero giornaliero di bersagli colpiti a Londra dai bombardamenti durante la Seconda Guerra Mondiale, errori di stampa nei libri, e diversi altri.

La formula è poco piacevole alla vista: $\boldsymbol{X}$ **è il numero** di **successi**, $P(X)$ è la sua probabilità, λ (**lambda**) è il **numero medio** di successi per unità di tempo ed e è la base del sistema di logaritmi naturali: $e = 2.71828$.

$$P(X) = \left(\lambda^X \times e^{-\lambda}\right)/X!$$

R ci toglie tutti i problemi: per esempio, se volessimo calcolare la probabilità di ricevere solo 2 mail di lavoro in un'ora a Ferragosto, sapendo che la media oraria negli altri giorni dell'anno è pari a 5, potremmo chiedere **`dpois(x=, lambda =)`**, in cui `X` è il numero di successi e `lambda=` è la media per unità di tempo:

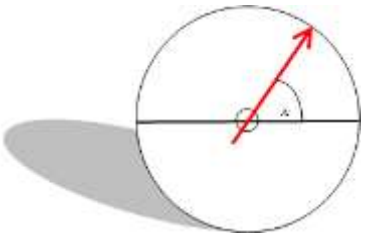
```
dpois(x = 2,lambda = 5)
[1] 0.08422434
```

Analogamente, per la funzione di ripartizione usiamo **`ppois`**`(q =, lambda= ,lower.tail=TRUE/FALSE)`, dove `q=` è il quantile in corrispondenza del quale effettuare la ripartizione e `lambda=` è come in `dhyper`. L'inverso della funzione di ripartizione si ottiene con **`qpois`**`(p= ,lambda=, lower.tail=TRUE/FALSE)` e una distribuzione casuale di numeri che segue la distribuzione di Poisson si ottiene con **`rpois`**`(n = , lambda = )`.

5.3 Variabili aleatorie continue

Alcuni esperimenti casuali non generano valori discreti, o comunque in quantità numerabile: quando la variabile X assume un insieme continuo di valori, è chiamata **variabile casuale continua**.

Pensiamo a una sorta di "ruota della fortuna", in cui la freccia centrale è fatta ruotare finché terminerà casualmente la corsa in una certa posizione, che registreremo come **numero di gradi x_1** da una linea di riferimento prefissata. In un secondo giro, si otterrà un nuovo valore in gradi x_2, e così via. In sostanza, possiamo ottenere un numero infinito di diversi valori x_i, quindi l'esperimento ci porta a uno spazio campionario $\boldsymbol{\Omega}$ che corrisponde a tutte le possibili posizioni assumibili dalla freccia.



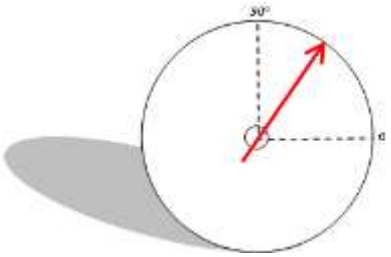
Indichiamo con X la variabile aleatoria che registra i gradi x_i: X può assumere tutti i valori dei gradi nell'intervallo $[0,360]$. Proviamo a calcolare la probabilità con cui la variabile aleatoria X assume un particolare valore x_i. Dato che la freccia non è truccata e la sua rotazione è del tutto casuale, possiamo usare la nota formula del caso di eventi elementari (casi di un esperimento aleatorio) equiprobabili:

$$\boldsymbol{P(X = x_i) = \frac{casi\ favorevoli}{casi\ possibili} = \frac{1}{\infty} = 0}$$

Questo significherebbe che **ciascuno dei possibili valori di X ha probabilità nulla di verificarsi**! Quindi, è un problema concettualizzare una variabile aleatoria continua come nel caso discreto. Però, non è impossibile calcolare le probabilità relative a eventi che coinvolgono variabili aleatorie continue.

Per esempio, con quale probabilità la variabile casuale X (l'ago) misurerà **un'inclinazione compresa tra 0 e 90°**? Dato che il settore $0 - 90°$ rappresenta la **quarta** parte di un cerchio, è intuitivo rispondere che:

$$\boldsymbol{P(0 \leq X \leq 90) = P(X \leq 90) = \frac{1}{4} = .25}$$



Perciò, nel caso di una variabile aleatoria continua X dovremo valutare quanto **cambia la probabilità da un valore all'altro di un intervallo**, anche infinitesimale: $(\boldsymbol{x, x + dx})$, in cui dx esprime l'incremento di X; se preferite, dovremo valutare la probabilità che $\boldsymbol{X}$ **assuma determinati valori nell'intervallo di riferimento** (rapporto incrementale).

Si usa la funzione $\boldsymbol{f(\bullet)}$, detta **funzione di densità di probabilità**Errore. Il segnalibro non è definito.: è la funzione che a ogni $\mathbb{R}$ associa il limite, per dx tendente a 0, del rapporto tra la probabilità che X assuma valori nell'intervallo $(x, x + dx)$ e l'ampiezza dx.

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)\mathrm{d}u$$

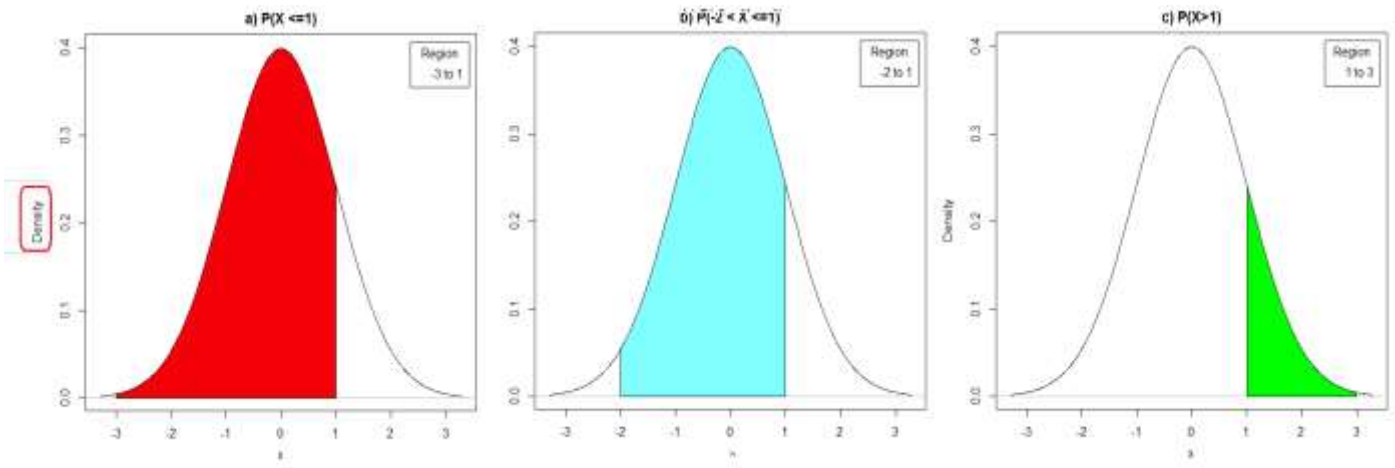
Consente, quindi, di calcolare quanto valga la **probabilità attorno a x_i in rapporto all'ampiezza dell'intervallo** – ed ecco perché si definisce "**densità".**

L'aspetto della funzione di densità farebbe tremare un troll norvegese, ma per fortuna non la useremo mai direttamente. Consideriamola un'estensione della formula che abbiamo visto applicata alle variabili aleatorie discrete: $\boldsymbol{P(X \leq x_i) = \sum_{x_i \leq x} P(X = x_i)}$, in cui l'integrale sostituisce la sommatoria e la funzione di densità $f(\bullet)$ sostituisce i termini $\boldsymbol{P(X \leq x_i)}$.

È utile dare **un'interpretazione grafica** della densità: **calcolare l'integrale della funzione di densità corrisponde a calcolare l'area sotto la curva $\boldsymbol{f(\bullet)}$ stessa**. Di seguito, abbiamo disegnato tre distribuzioni aleatorie continue, di forma normale (capitolo 2) con `RCommander` (`Distribuzioni` → `distribuzioni continue` → `distribuzione normale` → `Disegna distribuzione normale`).

In effetti:

- per il grafico a), abbiamo calcolato la funzione di densità da $-\infty$ a $+1 \rightarrow P(X \leq 1) = \int_{-\infty}^{1} f(u)\mathrm{d}u = (a)$;
- per il grafico b), abbiamo rappresentato il valore dell'area sotto $f(\bullet)$ compresa tra -2 e $+1$, cioè $P(-2 < X \leq 1)$ e infine,
- per il grafico c), abbiamo rappresentato l'area a destra del punto $+1$, cioè $P(> 1)$.



È facile intuire che, come nel caso discreto, la somma delle probabilità in corrispondenza di tutti i valori assumibili dalla variabile aleatoria continua sia $= 1$. Questa proprietà per le variabili casuali continue corrisponde a che **l'area totale sotto la curva $f(\bullet)$ sia $= 1$.**

Riassumiamo:

Variabili aleatorie discrete	**Variabili aleatorie continue**
Una variabile aleatoria discreta è un modello matematico $\{x,P(X=x_i),\ i=1,...k\}$, che associa ai valori $x_1,x_2,...x_k$ le rispettive probabilità $\{P(X=x_i),\ i=1,...k\}$. La **funzione di probabilità** $\{P(X=x_i),\ i=1,...k\}$ è tale per cui: $\sum_{i=1}^{k} P(X = x_i) = 1$	Una variabile aleatoria continua è un modello matematico $\{x,f(x)\}$. La **funzione di densità di probabilità** $f(\bullet)\geq 0$ è tale per cui: $P(X \leq x) = \int_{-\infty}^{x} f(u)\mathrm{d}u;\ P(X \leq x) = \int_{-\infty}^{+\infty} f(u)\mathrm{d}u = 1$ dove $P(X \leq x)$ rappresenta l'area sotto la curva fino al punto x. ***ATTENZIONE:*** ***per le variabili aleatorie continue,*** $P(X \leq 1) = 0$

Non vi stupirà che le funzioni che abbiamo visto applicabili in R a una distribuzione discreta binomiale si possono applicare anche a variabili continue, con qualche cambiamento negli argomenti dovuto alle specificità delle distribuzioni.

5.3.1 Distribuzione di probabilità normale

Cominciamo con la distribuzione normale teorica che già conosciamo: per calcolare la funzione di densità si usa la funzione **`dnorm(x, μ, σ)`**, che restituisce **l'altezza della curva (la densità di probabilità) normale**, con media = μ e deviazione standard = σ, per ogni x_i. Se μ e σ non sono specificate, R assume che la distribuzione normale sia standardizzata, con $\mu = 0$ e $\sigma = 1$. Ricordiamo ancora una volta**:** mentre per **variabili discrete** **`dbinom`**, **`dpois`** e **`dhyper`** danno **la probabilità del valore** x_i**,** per **variabili continue la probabilità** è sempre restituita per **intervalli tra X, non per singoli** x_i**.** Per esempio, la densità di probabilità di +2 in una distribuzione normale con $\mu = 0$ e $\sigma = 1$ è:

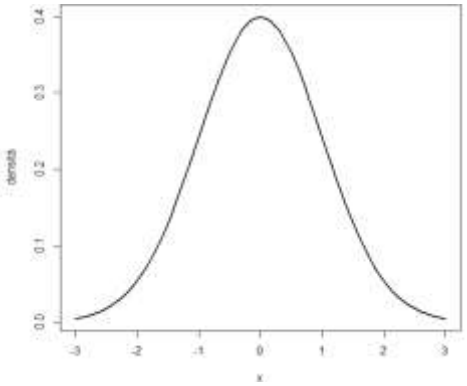
```
dnorm(x = 2,mean = 0, sd = 1)
[1] 0.05399097
```

Dato che la curva normale è simmetrica **NON DEVE** stupirci che la densità di probabilità di -2 sia identica:

```
dnorm(x = -2,mean = 0, sd = 1)
[1] 0.05399097
```

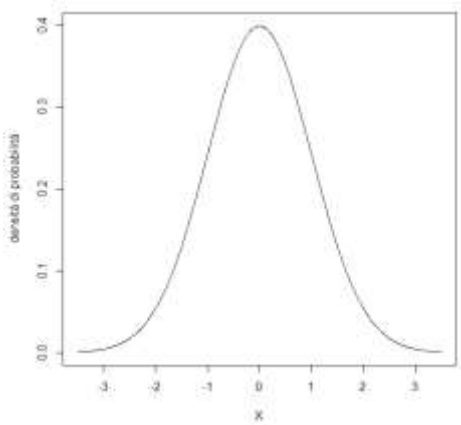
Possiamo facilmente plottare la **curva di densità** per variabili continue, come abbiamo plottato la massa di probabilità per variabili discrete: nei grafici precedenti abbiamo usato `RCommander`, ma con **`curve(funzione, da=, a=)`**, che fa parte delle statistiche di base, si fa molto rapidamente:

```
curve(dnorm,from = -4,to = 4, lwd=2, ylab="densità",
xlab="x")
```



Naturalmente, potremmo anche usare i comandi usati per plottare `dbinom` con `plot`, ma questa volta il grafico dovrà essere di **`type="l"`** (**lines**) per visualizzare la curva come linea continua invece che come distribuzione di punti separati (provate a eseguire la funzione senza l'argomento `type=` per vedere come si presenta)

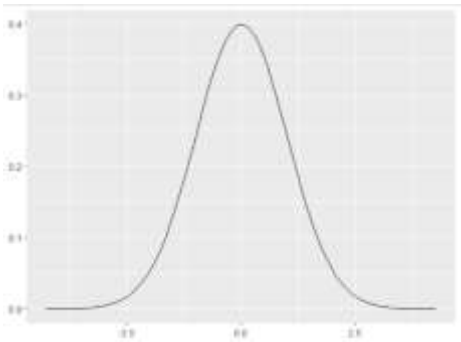
```
aleatoria<-seq(from = -3.5,to = 3.5, length.out = 1000)
plot(aleatoria, dnorm(aleatoria, 0,1))
```



Invece di `length.out` potete specificare la sequenza con `by=`, usando un passo molto ristretto:

```
aleatoria2<-seq(from = -3.5,to = 3.5, by=0.01)
```

Se siete di fretta, si traccia una curva normale specificando la sua media e la sua deviazione standard con `dist_norm(mean= , sd= )` di `sjPlot`; inseriamo `mean=0` e `sd=1` per tracciare una normale standardizzata:



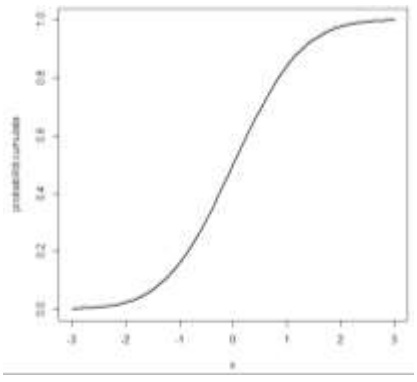
Con la funzione di ripartizione **`pnorm(q= quantile, μ, σ, lower.tail=TRUE/FALSE)`** torniamo a calcolare la probabilità cumulata: da $-\infty$ **fino a** x_i (se `lower.tail=TRUE`) o da x_i **a** $+\infty$ (se `lower.tail= FALSE`). Come esempio, usiamo uno "speciale" quantile della distribuzione standardizzata, cioè $z = 1.96$, che troveremo ben presto nel capitolo 6: vediamo qual è la probabilità cumulata di ottenere un valore $z \leq 1.96$:

```
pnorm(q = 1.96, mean = 0, sd = 1, lower.tail = TRUE)
[1] 0.9750021
```

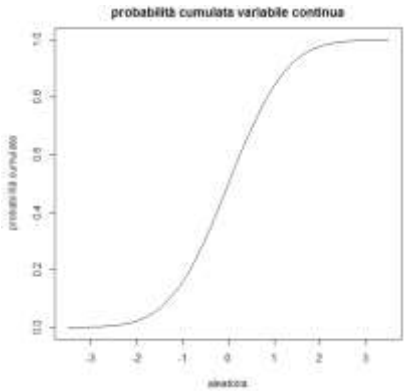
La probabilità cumulata di ottenere un valore $z > 1.96$ è :

```
pnorm(q = 1.96, mean = 0, sd = 1, lower.tail = FALSE)
[1] 0.0249979
```

Il grafico della probabilità cumulata per variabili continue è un'**ogiva**: possiamo usare ancora **`curve`**, applicata a `pnorm`, oppure il vettore `aleatoria` per plottare il grafico della probabilità cumulata, con `plot`:



```
curve(pnorm,from = -3,to = 3, lwd=2,
  ylab= "probabilità cumulata",
  xlab="x")
```




```
plot(aleatoria, pnorm(aleatoria, 0,1),
  main="probabilità cumulata variabile continua",
  type="l", ylab="probabilità cumulata")
```

L'inverso della funzione di ripartizione è **qnorm(p= probabilità cumulata, μ, σ, lower.tail= TRUE / FALSE)**, che, come nella distribuzione binomiale, restituisce il quantile della variabile aleatoria normale corrispondente alla probabilità cumulata indicata nella funzione. Il quantile che corrisponde a una probabilità cumulata $= .50$ è:

```
qnorm(p = .50, mean = 0, sd =1)
[1] 0
```

Naturalmente, il quantile restituito è la media della distribuzione, essendo moda, mediana e media coincidenti: in una distribuzione di probabilità normale, come in una distribuzione di frequenza normale, la media della distribuzione di probabilità divide in due metà esatte la distribuzione.

Scopriamo un altro quantile "magico", che lascia alla sua sinistra (fino a $-\infty$) il 95% della distribuzione e che ritroveremo nel capitolo 6:

```
qnorm(p = .95, mean = 0, sd =1)
[1] 1.644854
```

Per simmetria:

```
qnorm(p = .05, mean = 0, sd =1)
[1] -1.644854
```

Infine, per generare una variabile aleatoria di numeri casuali con distribuzione normale usiamo **rnorm(n=, μ, σ)**, in cui `n=` è il numero di elementi della distribuzione.

```
random_normale<-rnorm(n = 1000, mean = 10, sd = 5)
```

5.3.2 Variabili aleatorie continue diverse dalla normale

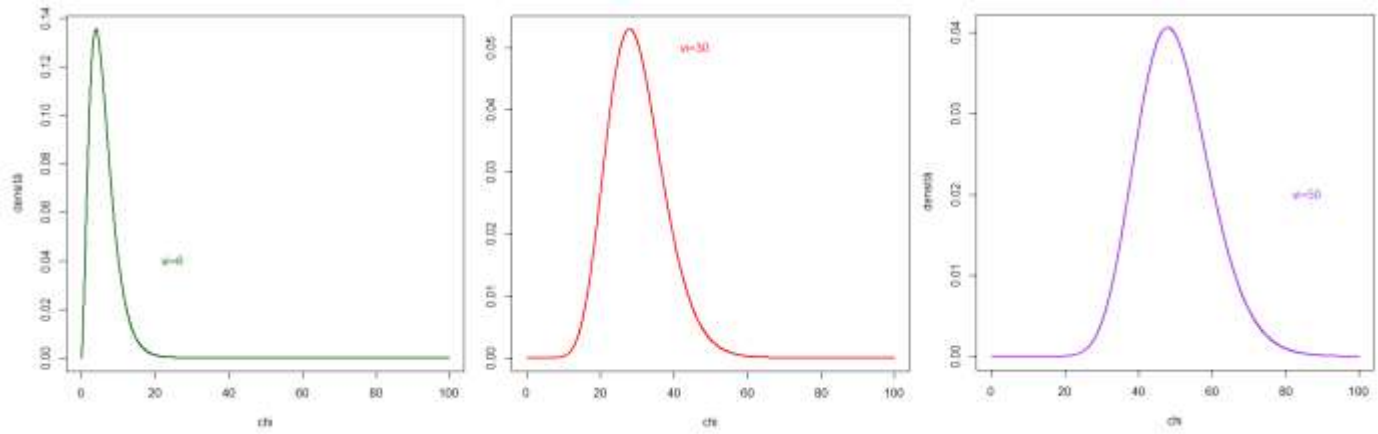
Quando passeremo alla verifica delle ipotesi, ci troveremo ad avere a che fare con fenomeni che si distribuiscono come variabili continue non solo normali (vedremo pure che, in alcune condizioni, alcune distribuzioni discrete si approssimano alla normale). Elenchiamole rapidamente:

Distribuzione chi quadrato χ^2

La distribuzione chi quadrato (χ^2), che dobbiamo a Karl Pearson, è una distribuzione di valori **standard al quadrato, indipendenti, estratti da una** variabile standardizzata distribuita normalmente, con μ e σ noti. Sono quindi definiti solo sull'asse positivo, da 0 a $+\infty$:

$$\chi^2_\nu = \sum_1^\nu \frac{(x_i - \mu)^2}{\sigma^2}$$

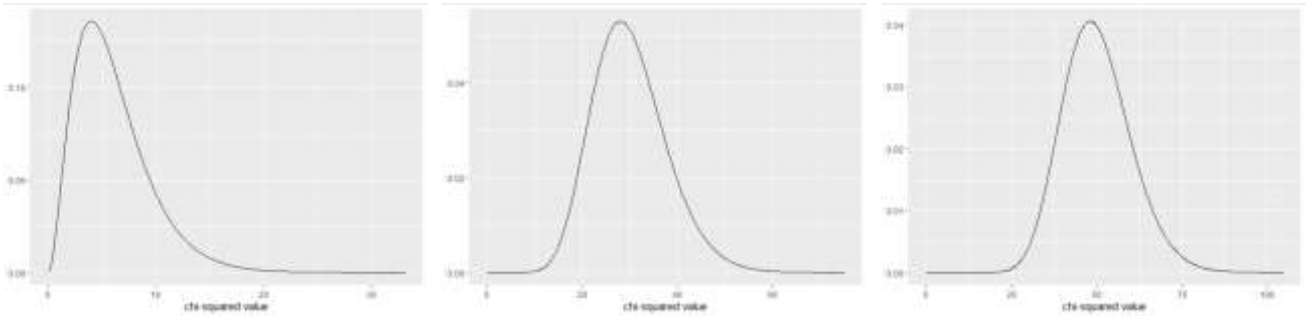
Notate che a pedice di χ^2 è indicato ν (**ni**): il **parametro** ν **corrisponde ai gradi di libertà** della distribuzione (gdl o df: li abbiamo già visti nel capitolo 3), è pari a $N - 1$ e **determina la forma di ogni distribuzione di probabilità** χ^2. Vediamo tre esempi di distribuzione χ^2 per diversi gradi di libertà; notate cosa succede alla curva man mano che i df aumentano:



Le tre distribuzioni sono state create plottando la variabile aleatoria `chi <- seq(0, 100, by=.05)` in ascissa e la funzione `dchisq(x, df)` in ordinata, variando l'argomento `df=` gradi di libertà.

Provate a creare i grafici delle tre distribuzioni, aggiungendo con `text` anche il testo nel grafico

Dopo l'esercizio, potete anche imparare a usare la più semplice funzione `dist_chisq(df=)` di `sjPlot`, specificando i diversi gradi di libertà:

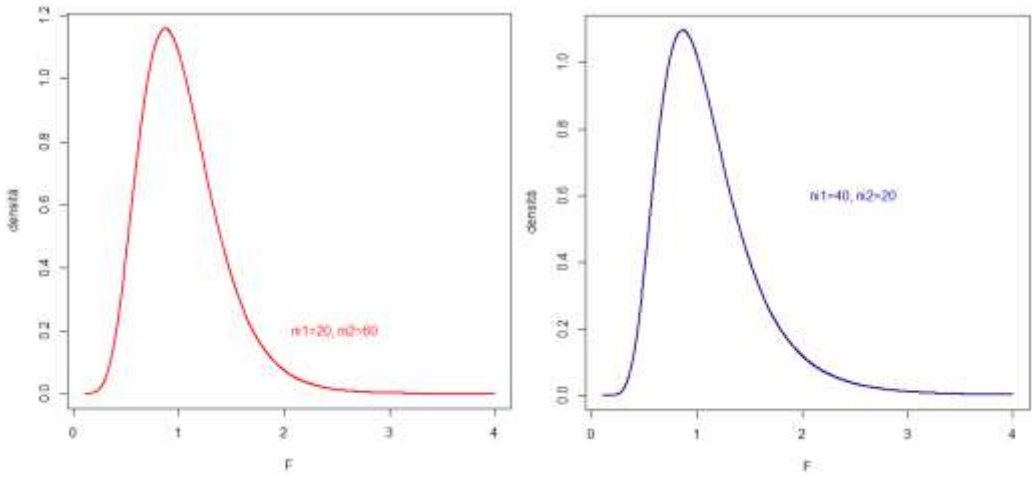


In analogia alle precedenti, si usa la funzione **`rchisq`**`(numero di osservazioni, df)` per creare una distribuzione di numeri casuali con distribuzione χ^2, **`pchisq`**`(quantile, df, lower.tail=TRUE/FALSE)` per ricavare la probabilità cumulata e **`qchisq`**`(probabilità cumulata, df, lower.tail=TRUE/FALSE)` come inverso della precedente.
Useremo la distribuzione χ^2 per stimare probabilità nei test di associazione fra variabili nominali, nella verifica della normalità multivariata e nella regressione logistica.

Distribuzione *F*

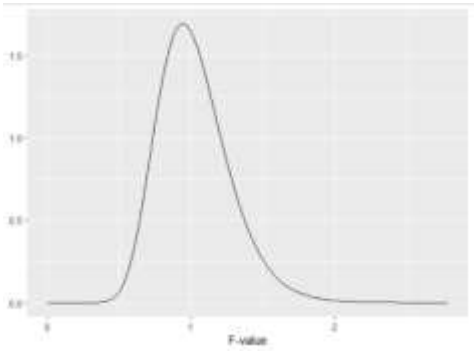
La distribuzione ***F* di Snedecor** è stata da lui così battezzata in onore di Robert Fisher: è data dal rapporto tra due distribuzioni χ^2 indipendenti; quindi, i suoi valori possono essere solo positivi. I gradi di libertà che definiscono la distribuzione F sono due, uno per la distribuzione al numeratore, l'altro per la distribuzione al denominatore.

$$F_{\nu 1,\nu 2} = \frac{\chi^2_{\nu 1}}{\chi^2_{\nu 2}} \times \frac{\nu 1}{\nu 2}$$



Provate a creare i due grafici della distribuzione della variabile `F<- seq(0.1, 4, length.out=1000)`

Esiste anche `dist_f(deg.f1 = , deg.f2 =)` di `sjPlot`;
inserendo come ν_1 `dfg.f1=60` e come ν_2 `deg.f2=80`,
otteniamo:



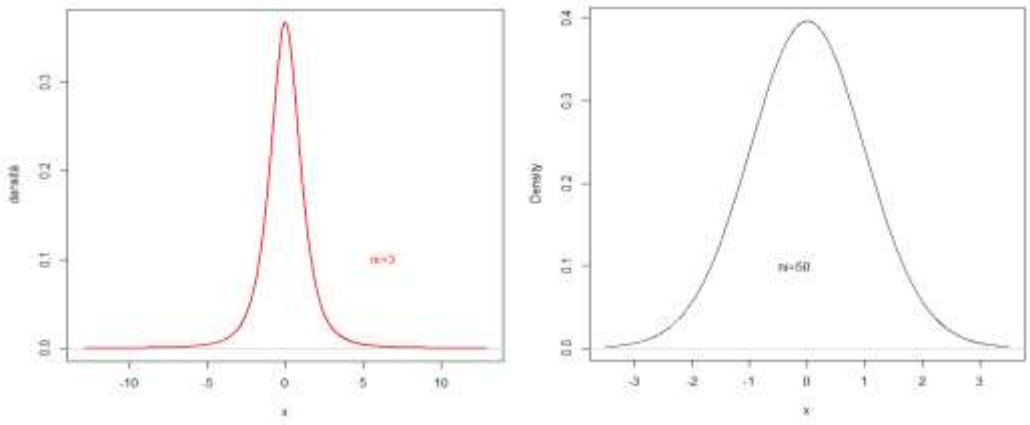
Abbiamo la funzione **`df`**`(x, df1, df2)` per stimare la densità, **`rf`**`(numero di osservazioni, df1, df2)` per creare una distribuzione di numeri casuali con distribuzione F, **`pf`**`(quantile, df1, df2 lower.tail=TRUE/FALSE)` per ricavare la probabilità cumulata e **`qf`**`(probabilità cumulata, df1, df2, lower.tail=TRUE/FALSE)` come inverso della precedente.
Useremo diffusamente la distribuzione F nei modelli lineari: regressione multipla e analisi della varianza.

Distribuzione *t*

La distribuzione ***t* di Student** (che è uno pseudonimo di **Gosset**) è soprattutto utile per campioni di n< 30. È data dalla **radice** quadrata della distribuzione F; anche in questo caso il parametro è ν.

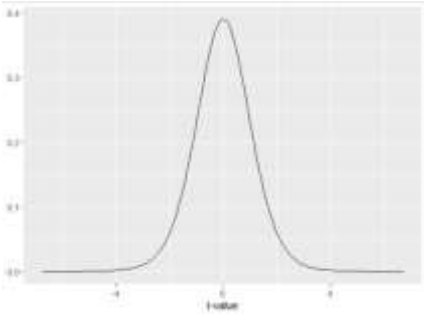
$$t = \sqrt{F_{1,\nu}}$$

La distribuzione t è **simmetrica intorno a *t*= 0**: assume quindi, simmetricamente, valori negativi e positivi. Per gradi di libertà tendenti a infinito (cioè sufficientemente grandi), si **approssima alla distribuzione normale**.



Sono disponibili la funzione `dt(x, df)` per stimare la densità, `rt(numero di osservazioni, df)` per creare una distribuzione di numeri casuali con distribuzione *t*, `pt(quantile, df, lower.tail=TRUE/FALSE)` per ricavare la probabilità cumulata e `qt(probabilità cumulata, df, lower.tail=TRUE/FALSE)` come inverso della precedente.

Avrete probabilmente intuito che per tracciarla esiste `dist_t(deg.f=)` di `sjPlot`, in cui basta inserire i df della distribuzione *t*.
Il grafico rappresenta una distribuzione *t con* `deg.f= 100`:



Troveremo la distribuzione *t* in molti test: test per un campione, correlazione, *t-test* (naturalmente), modelli lineari (regressione multipla e contrasti a priori e pianificati nell'analisi della varianza).

1. Il quoziente d'intelligenza è distribuito normalmente in popolazione, con μ= 100 e SE=15.

a) *Tra quali due valori di QI (eliminate tutti i decimali) si trova il 68.2% dei valori di QI della popolazione?*
b) *Quale proporzione della popolazione ha un QI inferiore a 80?*
c) *Quale proporzione della popolazione ha un QI inferiore a 110?*
d) *Quale proporzione della popolazione ha un QI compreso tra 95 e 115?*
e) *Quale proporzione della popolazione ha un QI compreso tra 70 e 85?*
f) *Quale valore di QI (eliminate tutti i decimali) ha il 10% della popolazione con il QI più alto?*
g) *Quale valore di QI (eliminate tutti i decimali) ha il 10% della popolazione con il QI più basso?*

2. È noto che il tempo medio impiegato dagli studenti per recarsi in Università è distribuito normalmente, con una media = 19 e SE= 10:

a) *Quale proporzione di studenti impiega meno di 22 minuti (due decimali)?*
b) *Quale proporzione della popolazione impiega più di 15 minuti(due decimali)?*
c) *Se tutti gli studenti partono da casa 20 minuti prima che inizi una lezione, quanti di loro saranno in ritardo (due decimali)?*
d) *Quale proporzione di studenti impiega tra 21 e 25 minuti per arrivare in università (due decimali)?*

Capitolo 6
L'inferenza statistica

Thus, a theory can very well be found to be incorrect if there is a logical error in its deduction or found to be off the mark if a fact is not in consonance with one of its conclusions. ***But the truth of a theory can never be proven****.*
Einstein, Collected papers, vol. 7, doc. 28

"Hallmark of good science is that it uses models and "theory", but never believes them"
Wilks, già cit.

Finora, abbiamo usato grafici e modelli statistici per **descrivere** al meglio una distribuzione univariata, e modelli **probabilistici** per associare alle modalità della distribuzione le probabilità, al fine di quantificare la verosimiglianza di un evento. D'ora in avanti, costruiremo modelli statistici per rappresentare **inferenze**.

6.1 Inferenze statistiche e metodo induttivo

Lo studio di una qualsiasi variabile su un'intera popolazione è perlopiù tecnicamente impossibile o quantomeno eccessivamente oneroso: è però possibile, con opportuni metodi statistici, individuare un **campione rappresentativo** della popolazione e realizzare un'indagine sul campione con lo scopo di estendere le conclusioni dell'indagine all'intera popolazione da cui il campione è stato estratto. **L'estensione dei risultati da un campione alla popolazione è detta inferenza**: una o più affermazioni esplicite su proprietà di un universo più ampio, basate su un insieme di osservazioni molto più ristretto. Le inferenze sono alla base del metodo **induttivo**, componente essenziale del metodo scientifico. A differenza delle ben solide certezze cui si giunge usando il metodo **deduttivo** (pensate alle dimostrazioni matematiche...), il metodo induttivo è per sua natura **incerto**[29]**.** Nel metodo induttivo, in seguito a ripetute e accurate descrizioni di un fenomeno, si possono rilevare regolarità, che contribuiscono a definire **leggi**, cioè asserzioni secondo cui alcuni fenomeni sono regolarmente associati in popolazione. Le regolarità non sono perfette, dato che diversi fattori, estranei alle variabili oggetto della legge, possono interferire nell'associazione riscontrata: **le leggi definiscono come (molto o poco) probabili le regolarità** riscontrate.

Le leggi consentono di formulare **teorie**, cioè insiemi di asserzioni che organizzano un vasto corpus di leggi in un singolo sistema di spiegazioni. Mentre le leggi descrivono le relazioni tra fenomeni osservati, le teorie introducono concetti (**costrutti latenti**) **non** direttamente osservati, necessari per spiegare e generalizzare tali relazioni. Quindi, nelle teorie sono inseriti concetti **non** presenti nella legge e **non** direttamente osservabili né direttamente misurabili, che possono essere confermati, smentiti o modificati solo da altri dati empirici. Oltre che per organizzare conoscenze e leggi, le teorie consentono di prevedere nuove leggi: più specifica è la spiegazione della teoria, tanto più precisa è la previsione su come dovrebbero apparire i dati in funzione di nuove leggi derivabili dalla teoria.

```
immediata<-c(9,8,6,7)
differita<-c(3,2,1,2)
mean(immediata)
[1] 7.5
mean(differita)
[1] 2
summary(immediata-differita)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    5.0     5.0     5.5     5.5     6.0     6.0
```

[29] Dice Bertrand Russell: un tacchino americano, usando il metodo induttivo, osserva di venire regolarmente nutrito tutti i giorni, e prevede perciò che anche il giorno successivo sarà nutrito. Ma se il giorno successivo è il Thanksgiving Day, allora sarà il contadino a mangiare, e non il tacchino...

Una teoria può essere definita **valida** solo quando resiste ai tentativi di **falsificarla**: deve poter essere messa alla prova e deve poter essere dimostrata come falsa, ma **non sarà mai possibile dimostrarla come vera**. Il procedimento di falsificazione è complesso da comprendere, poiché contro-intuitivo rispetto al modo usuale di ragionare degli esseri umani, che cercano perlopiù prove a favore di una teoria. Ne risulta che nessuna teoria è da considerarsi mai vera con certezza, ma solo **vera fino a prova contraria**: tanto più numerose sono le prove cui sopravvive una teoria, tanto più ci si potrà fidare della teoria stessa. Affinché una teoria possa dimostrare di resistere alla falsificazione, viene **operazionalizzata** in una (o più) **ipotesi**. Operazionalizzare significa che un costrutto teorico viene definito attraverso le operazioni con cui è misurato: per esempio, lo stress può essere definito tramite / come il livello di cortisolo salivare rilevato due minuti prima della prova scritta, oppure l'apprendimento può essere definito tramite / come il numero di risposte corrette ad una prova di rievocazione di un brano dopo 20 minuti dalla lettura, o la fame tramite / come deprivazione totale del cibo per le 24 ore precedenti. Se operazionalizziamo diversamente lo stress tramite / come conduttanza cutanea, l'apprendimento tramite / come numero di equazioni lineari correttamente risolte o la fame tramite / come somministrazione di cibo appena sufficiente a mantenere l'80% del peso corporeo, produrremo probabilmente risultati leggermente diversi, che contribuiranno a creare un corpus comune di conoscenze riguardanti i tre costrutti.
L'ipotesi è un'affermazione che si **può definire come probabilmente vera o probabilmente falsa** a seguito di una prova (esperimento, osservazione, ricerca, ecc.) empirica; segue la forma logica se $A \rightarrow B$, ovvero "se la tal teoria è vera, allora in queste condizioni il fenomeno si presenterà in tal modo".

Dovrebbe quindi essere chiaro come **l'inferenza integri la descrizione di un fenomeno con la probabilità**: per rilevare regolarità da verificare con ipotesi, è **essenziale descrivere il più correttamente possibile i fenomeni**; poiché le regolarità sono soggette a diverse fonti di errore e a variabilità "naturale", è **necessario definirne la probabilità di manifestarsi sotto determinate condizioni, oggetto di ipotesi**. **Se un fenomeno si manifesta in maniera diversa da quella attesa in base all'ipotesi, bisogna chiedersi quale sia la spiegazione più semplice**[30] **per la differenza riscontrata**. Ricordate che Buffon e Pearson sono arrivati a due stime diverse della probabilità di ottenere una faccia di una moneta, entrambe peraltro diverse dalla stima prevista dalla definizione formale di probabilità? Dovremmo forse pensare che uno o entrambi hanno truccato la loro moneta, o che aver tirato (forse) l'uno un franco e l'altro una sterlina hanno prodotto risultati diversi? Giammai: la **spiegazione più semplice**, che dovremo accettare, è che **la differenza tra le due probabilità sia compatibile con l'effetto del caso**, ovvero **con le fluttuazioni campionarie**: quindi, concluderemo che **possa non esserci alcuna reale differenza**.
Questo procedimento di **verifica** (ma sarebbe meglio dire falsificazione) delle **ipotesi** è stato messo a punto da Ronald **Fisher**, come vedremo diffusamente nel paragrafo 6.4.
L'altra faccia della medaglia dell'inferenza è la stima intervallare (§6.3): un'indagine su un campione può porsi come obiettivo quello stimare il valore sconosciuto di un parametro nella popolazione (una media, una varianza, una correlazione tra variabili, una differenza tra medie, ecc.). In questo modo, affianchiamo alla stima puntuale del dato nel campione la corrispettiva stima intervallare in popolazione: individuiamo un intervallo di valori all'interno dei quali il parametro sconosciuto deve trovarsi, con un grado di verosimiglianza elevato e predefinito. L'intervallo di valori è definito intervallo di fiducia (**confidence interval**, CI). L'ampiezza dell'intervallo sarà una misura preziosa della precisione della stima e dell'utilità della ricerca stessa: per quanto veridica, l'affermazione che la percentuale di promossi all'esame di Tecniche di analisi di dati I sta, con il 99% di probabilità, tra lo 0% e il 100%, non vi dovrebbe aiutare molto rispetto alla decisione sul tipo di atteggiamento che dovreste tenere rispetto al fare gli esercizi...

[30] Dice Guglielmo di Occam: "Entia non sunt multiplicanda preater necessitatem", o, secondo un'altra vulgata, "Pluralitas non è est ponendum sine necessitatem": non devono essere chiamate in causa più spiegazioni del necessario, ovvero, spiegazioni più semplici devono essere preferite a spiegazioni più complesse di uno stesso fenomeno. Galileo e Newton, separatamente, concordano.

6.2 Campionamento bernoulliano e distribuzione campionaria

Prima, un po' di definizioni.
Grandi insieme di eventi elementari sono comunemente chiamati **popolazioni** o **universi**, anche se il termine **spazio campionario** è forse più descrittivo. Il termine **distribuzione della popolazione** si riferisce alla distribuzione dei valori delle osservazioni possibili nello spazio campionario. Sebbene le caratteristiche o **parametri** della popolazione (per esempio la sua media, μ, o la sua deviazione standard, σ) siano di interesse pratico e teorico, sono raramente, se non mai, noti con esattezza. **Stime** dei loro valori sono ottenuti dai corrispondenti valori campionari, ovvero le **statistiche**. Chiaramente, per un campione di una data numerosità estratto casualmente da uno spazio campionario, esiste una distribuzione di valori di una particolare statistica riassuntiva: la **distribuzione campionaria** (**sampling distribution**). Nell'applicazione della statistica alla ricerca, sono le proprietà di queste distribuzioni che guidano le inferenze sulle proprietà delle popolazioni.

Perché le conclusioni dell'inferenza siano valide, è necessario che i campioni siano rappresentativi della popolazione: il metodo più usato per ottenere un campione rappresentativo è il campionamento **casuale o random o bernoulliano**, in cui ogni unità della popolazione ha la medesima probabilità di entrare a far parte del campione.
Quando si deduce un **parametro** di una popolazione (la sua media, la sua varianza, ecc.) sulla base delle corrispondenti statistiche osservate nel campione casuale, si sta effettuando una **stima** del parametro. La stima rappresenta un'approssimazione statistica ai risultati della ricerca sull'intera popolazione: si può considerare come una "**verità relativamente ottimale**", affetta da un errore accettabile e proporzionato al costo sostenibile in termini di tempo e denaro. Il campionamento random garantisce che la stima sia preservata da un **errore sistematico** di campionamento, e che l'errore riscontrabile nella stima sia solo **casuale**: l'errore casuale agisce in modo impredicibile su ogni misura, che viene sovrastimata e sottostimata un numero simile di volte. L'errore casuale fluttua, varia da prova a prova: i suoi effetti tendono a compensarsi quando le rilevazioni sono ripetute un gran numero di volte. Al contrario, un **errore sistematico** agisce in modo costante e sistematico per tutte le misurazioni, come una bilancia che mostra un peso inferiore di 2 kg per tutte le pesate; conduce sempre nella stessa direzione, verso una costante *sottostima* o *sovrastima.*
Se una misura è soggetta a molte piccole sorgenti di errori casuali e a trascurabili errori sistematici, allora i valori misurati saranno distribuiti su una curva a campana (l'abbiamo visto nel capitolo 4), e questa curva sarà centrata sul valore medio. **Operando infinite misurazioni, la media degli errori casuali tende a 0**.
In pratica, non ci sarà più sufficiente che il modello statistico abbia un buon fit nel campione: dovremo stabilire se ha un **buon fit rispetto alla popolazione da cui il campione è stato estratto**.

Possiamo estrarre più campioni, ciascuno dei quali con una prefissata numerosità, da **una popolazione**: per **ciascun campione**, possiamo **calcolare modelli / statistiche che lo descrivono -** la media del campione, la varianza del campione, la curtosi del campione, ecc. La **distribuzione delle statistiche relative a ciascun campione** costituisce una distribuzione campionaria (**sampling distribution**): avremo quindi la distribuzione campionaria delle medie dei campioni ($\boldsymbol{DCM}$), la distribuzione campionarie delle varianze dei campioni, la distribuzione campionaria delle curtosi dei campioni, e così via. Se i campioni sono stati estratti in maniera realmente casuale, **ciascuna delle statistiche varierà con una quota di errore casuale all'interno della distribuzione**. Per ciascun tipo di distribuzione campionaria è possibile calcolare descrittori: possiamo calcolare la media della distribuzione campionaria delle curtosi per sapere qual è la curtosi media dei campioni estratti, per esempio.
Delle varie distribuzioni campionarie possibili, il **nostro interesse si concentra su quella del modello statistico** descrittivo più efficace per variabili continue, cioè la **media**: la **distribuzione campionaria delle medie** o DCM.

Possiamo usare R per simulare efficacemente una distribuzione campionaria. Sappiamo che possiamo creare distribuzioni di numeri casuali da una popolazione con media e deviazione standard note: usiamo la distribuzione normale, e con **rnorm** estraiamo **mille distribuzioni**, cioè **mille campioni**, ciascuno dei quali composto da **100 numeri** casuali (**N=100**, campionamento **random**) da una **popolazione con $\mu = 50$ e $\sigma = 5$**. Le **mille medie delle mille distribuzioni così create costituiranno la nostra distribuzione campionaria delle medie**.

Certo, potremmo farlo distribuzione per distribuzione:

```
campione1<-rnorm(N=100, mean=50, sd=5)
campione2<-rnorm(N=100, mean=50, sd=5)
campione3<-rnorm(N=100, mean=50, sd=5)
```

fino al campione 1000, ma nella vita ci sono decisamente cose più interessanti cui dedicare tempo. Quindi, usiamo `replicate(n= numero di repliche, expr= espressione da replicare)`, con cui creiamo una matrice (**campioni**) composta da 1000 colonne (1000 campioni) di 100 righe ciascuno (100 soggetti). In ogni colonna sarà contenuta una distribuzione normale di 100 casi, casualmente estratta (`rnorm`) da una popolazione normale con $\mu = 50$ e $\sigma = 5$.

```
Campioni <-
matrix(replicate(n=1000,
expr=rnorm(n=100,
mean=50,sd=5)), nrow=100,
ncol=1000)
```

Ora dobbiamo calcolare le medie delle 1000 colonne, cioè le medie dei 1000 campioni che costituiscono la Distribuzione Campionaria delle Medie e che saranno contenute nell'oggetto ***DCM***. Usiamo una funzione simile a `tapply`: **`apply(X=matrice/dataframe, MARGIN=margine, FUN=funzione)`**. Come `tapply` ripete la funzione indicata da `FUN=` applicandola ai livelli di uno o più `factor`, così `apply` ripete la funzione dell'argomento `FUN=` applicandola alle righe (`MARGIN=1`) o alle colonne (`MARGIN=2`) di matrici o dataframe (`X`). Quindi, dato che noi dobbiamo calcolare la media (`FUN=mean`) delle 100 osservazioni in ciascuna nelle 1000 colonne / campioni della DCM (`MARGIN=2`) contenute nella matrice `campioni` (`X=campioni`) , scriveremo

```
DCM <- apply(X = campioni,MARGIN = 2, FUN = mean)
length(DCM)
[1] 1000
head(DCM); tail(DCM)
[1] 49.90577 50.10111 50.28119 49.63651 49.88786 50.25142
[1] 49.83510 50.34029 50.18313 49.94768 50.49777 49.55514
```

Le 1000 medie contenute nella DCM oscillano casualmente attorno alla media attesa della popolazione da cui abbiamo estratto ogni campione ($\mu = 50$), anche se nessuna tra esse è esattamente $= 50$:

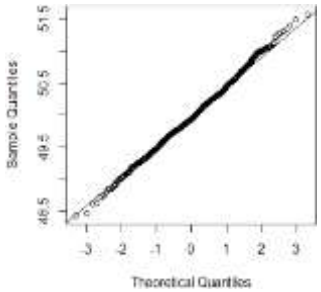
```
which(DCM==50.0)
integer(0)
```

La **media** della DCM è, approssimandola, pari a quella attesa in popolazione:

$$\mu = \mu_{\bar{x}}$$

La **forma** della DCM è assai simile alla forma normale della popolazione da cui abbiamo estratto i campioni:

```
mean(DCM)
[1] 49.96554
```



*Sembra scontato, ma esplicitiamolo: se provate a rifare la procedura con R, otterrete risultati simili a questi, ma leggermente **diversi**, dato che l'estrazione dei campioni è random!*

La deviazione standard della DCM ha un nome tutto suo: **errore standard o SE**

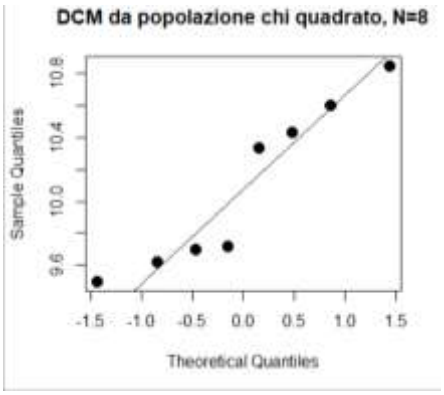
$$\sigma = SE_{\bar{x}}$$

Come la varianza e la deviazione standard sono indicatori della variabilità attorno al modello media nel campione, così **l'errore standard indica quanta variabilità ci sia tra le statistiche calcolate** (le medie, nel caso della DCM) dei differenti campioni: è quindi un **indicatore della variabilità del fenomeno indagato in popolazione**.
Naturalmente, nella realtà non conosciamo pressoché mai la vera varianza / deviazione standard della popolazione. Ci accontentiamo perciò di una **stima** dello SE, sfruttando le proprietà della distribuzione normale.

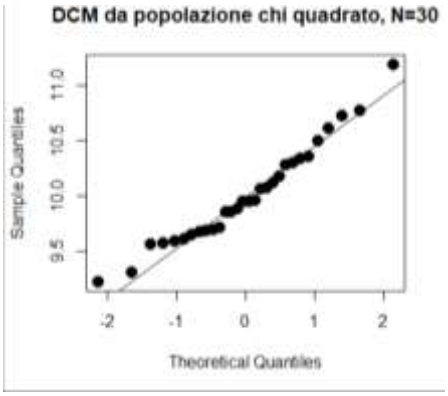
Sappiamo che tanto più il campione è grande ($N \geq 30$), tanto più la distribuzione campionaria tende a una forma normale, con **media uguale a μ** e deviazione standard uguale a:

$$\sigma_{\bar{x}} = \frac{s}{\sqrt{N}}$$

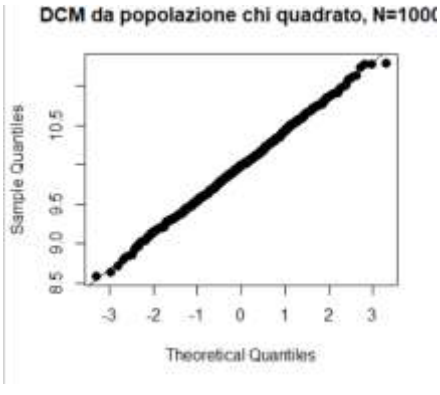
Il fatto che la DCM tenda alla distribuzione normale per 1000 campioni potrebbe essere ritenuto solo un effetto del fatto che la popolazione da cui abbiamo estratto i campioni è normale. In realtà, **indipendentemente da quale sia la forma della popolazione da cui li estraiamo, più il numero di campioni N aumenta, più la forma della DCM tende alla normale**. Vediamo tre esempi di estrazione casuale da una popolazione con distribuzione **chi quadrato**: in tutti e tre i casi i campioni sono composti da 100 osservazione e i gradi di libertà sono 10. La prima DCM è composta da 8 campioni, la seconda da 30, la terza da 1000. Osservando i Q-Q plot, si nota come la DCM tenda man mano ad adeguarsi a una forma normale.




```
chi_8<-matrix(replicate(n=8,
  expr=rchisq(n = 100, df = 10)),
  nrow=100, ncol=8)
DCM_8 <- apply(X = chi_8,MARGIN = 2,
  FUN = mean)
qqnorm(DCM_8, main="DCM da popolazione
  chi quadrato, N=8", pch=19, cex=1.5)
qqline(DCM_8)
```




```
chi_30<-matrix(replicate(n=30,
 expr=rchisq(n = 100, df = 10)),
 nrow=100, ncol=30)
DCM_30 <- apply(X = chi_30,MARGIN = 2,
 FUN = mean)
qqnorm(DCM_30, main="DCM da popolazione
 chi quadrato, N=30", pch=19, cex=1.5)
qqline(DCM_30)
```




```
chi_1000<-matrix(replicate(n=1000,
  expr=rchisq(n = 100, df = 10)),
  nrow=100, ncol=1000)
DCM_1000 <- apply(X = chi_1000,MARGIN =
  2, FUN = mean)
qqnorm(DCM_1000, main="DCM da
  popolazione chi quadrato, N=1000",
  pch=19, cex=1.2)
qqline(DCM_1000)
```

Questa osservazione costituisce il **teorema centrale del limite**[31]: **indipendentemente da quale sia la forma della popolazione da cui li estraiamo, più il numero di campioni N aumenta, più la forma della DCM tende alla normale**, **con** $\bar{x}_{DCM} = \mu$ **e** $SE = \frac{\sigma}{\sqrt{N}}$.

Più precisamente, dovremmo dire che **per** N **variabili aleatorie indipendenti, con uguale media e uguale** s^2**, indipendentemente dalla forma delle singole distribuzioni, la successione delle variabili aleatorie standardizzate tende ad avere una distribuzione normale, con** $\bar{x} = 0$ **e** $s = 1$

È grazie a questa legge che possiamo stimare la media di una popolazione con distribuzione non nota, usando la media campionaria, nonché il suo SE. In genere, con $N > 30$ si può essere abbastanza tranquilli nell'approssimare la distribuzione campionaria a quella normale.

6.3 Inferenza e intervallo di fiducia (*confidence interval*, CI)

Sono molti i metodi sviluppati per permettere inferenze dai dati: in questo paragrafo ci occupiamo di uno tra questi, le **stime intervallari** (*interval estimations*). Nonostante siano stati proposti più approcci alle stime intervallari, diversi per fondamenti e metodo di calcolo, in tutti si ottiene la **stima un parametro in popolazione che tiene conto dell'incertezza della misura / della variabilità campionaria**, fornendo un range di valori per il parametro invece di un singolo valore campionario. Tra questi approcci, i ***confidence* intervals o** CI, basati sulle stime campionarie, sono attualmente raccomandati come base per la migliore statistica inferenziale(ad esempio, Wilkinson e the Task Force in Statistical Inference, 1999; Cumming, 2008): li ritroveremo nel §6.6, con qualche limitato *caveat* all'entusiasmo.

Il CI di un parametro θ (**theta**), che può essere una qualunque quantità sconosciuta in popolazione (media, varianza, mediana, probabilità…), è un **intervallo** generato da una **procedura** (***confidence procedure***, CP) che, procedendo a **ripetuti campionamenti** (*sampling*), ha una probabilità prefissata di contenere il parametro θ.

Secondo la definizione originale di Neyman (1937), "un $x\%CI$ per un parametro θ è un intervallo, delimitato da un limite inferiore (**L: lower limit**) e un limite superiore (**U: upper limit**), generato da una procedura tale per cui, in ripetuti campionamenti, il CI ha una probabilità pari al $x\%$ di contenere il vero valore di θ, per tutti i possibili valori di θ". La definizione moderna afferma che il CI ha probabilità pari **ad almeno il** $x\%$ **di probabilità** di contenere θ, ma la differenza è piuttosto irrilevante ai fini del discorso. Il livello di probabilità si definisce ***confidence level*** (livello di fiducia), descrive **l'incertezza** associata a un metodo di campionamento ed è **arbitrario**: abitualmente, si costruiscono CI con il 95% (è l'opzione di default in R) o il 99% di probabilità di contenere il parametro in popolazione in campionamenti ripetuti, ma sono mere **convenzioni**, probabilmente ereditate dalla procedura di verifica delle ipotesi che vedremo nel §6.4.

Distinguiamo le caratteristiche della procedura che genera il CI dal suo prodotto: la *confidence procedure* è una qualsiasi procedura che genera CI che comprendono θ in una data percentuale x di campionamenti ripetuti, mentre il CI è uno specifico intervallo generato: la procedura è un processo casuale, ma il CI è osservato e fisso. Se ripetiamo la *confidence procedure* 100 volte, generando 100 CI, e l'abbiamo impostata in modo tale che ogni CI generato abbia il 95% di probabilità di contenere il valore atteso θ, avremo una distribuzione di $\simeq 95$ CI che contengono θ e $\simeq 5$ CI che non contengono θ: quindi, ognuno di questi 100 CI contiene o non contiene θ, dicotomicamente. Quello sviluppato da Neyman, quindi, è un metodo che propone una serie di attività per controllare i tassi di errore.

[31] O **teorema del limite centrale**; postulato da Laplace (1812), poi sistematizzato da Bernstein e Feller. Dovrebbe la sua denominazione di "centrale" proprio all'essere un elemento cardine nei metodi che si basano sull'inferenza statistica.

La **potenza** (*power*) delle CP è la frequenza con cui sono esclusi i valori **falsi** del parametro ($\boldsymbol{\theta'}$): diverse procedure di confidenza escluderanno questo valore falso con tassi differenti, per cui se la CP_A esclude θ' più spesso, in media, della procedura CP_B, allora CP_A è meglio di CP_B, per **quel parametro**. Talvolta, una CP esclude ogni falso valore θ' più spesso di qualsiasi altra: in questo caso, sarà uniformemente più potente, ma anche se una CP migliore non dovesse emergere, potremmo sempre comparare una procedura all'altra per determinare la migliore (Neyman, 1952).
Malgrado i primi scettici (tra cui, naturalmente, Fisher, 1935), i CI hanno guadagnato in popolarità a spese di altre procedure di stima intervallare, come gli intervalli fiduciali di Fisher - ***fiducial intervals*** (su cui il dibattito si sta forse riaprendo: Efron, 1998), o i ***credible intervals*** bayesiani (v. §6.6).

Quella che vediamo di seguito è il **più usuale metodo** per costruire un CI, ed è basato sulla conoscenza descrittiva di un dato campionario, ma **non è l'unico**: esistono altri metodi, caratterizzati dai diversi modi con cui si ricerca la stima puntuale da cui costruire il CI, tra cui la ***likelihood theory*** (ne parleremo nei capitoli 14 e 15) e il metodo ***bootstrapping*** (è un metodo di ricampionamento che non fa parte del programma, raccomandabile per CI in caso di distribuzioni fortemente non normali).

Il CI è centrato sulla statistica del campione, e quindi **contiene sempre la statistica campionaria**[32], che è una stima **puntuale**; **perlopiù**, la sua ampiezza è determinata **dall'errore standard**, moltiplicato per la **verosimiglianza prescelta** (α), o ***confidence level***: poiché dobbiamo individuare un limite superiore e un limite inferiore all'intervallo, la verosimiglianza **α è divisa a metà** ($\alpha/2$): $\boldsymbol{CI = statistica\ campionaria\ \pm SE \times {}^{\alpha}/_{2}}$.
L'errore standard è la stima della variabilità in popolazione, funzione della deviazione standard campionaria e di N: sappiamo che nel caso della media, per esempio, $SE = s/\sqrt{N}$ (se assumiamo una distribuzione normale).

Attenzione: la maggior parte dei CI con cui lavoreremo sono basati su questa formula, **ma non tutti.** Per esempio, vedremo i CI relativi a una proporzione (§6.7, §7.1), per il cui calcolo sono proposti diversi metodi, tra cui il metodo di **Wald**, basato su questa formula:

$$\hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{N}}$$

in cui z è il quantile della distribuzione nomale standardizzata corrispondente ad $\alpha/2$ e $\hat{p}$ ("p-hat") è la proporzione dell'evento atteso, cioè il numero di volte in cui si è verificato l'evento atteso sul totale. Non preoccupatevi di impararla a memoria, ma consideratela un esempio del fatto che **non esiste un solo modo** per ottenere un CI, anche se lavoreremo perlopiù con la formula più nota: $statistica\ \pm\ SE_x\ \times\ t(n-1)$.

In ogni caso, il CI si estende per una **distanza w** (***width***: **margine di errore**, *margin of error*): è delimitato dal limite inferiore (**lower limit – LL**) e da un limite superiore (**upper limit – UL**):

$$\boldsymbol{CI = LL_{\alpha/2} < parametro < UL_{\alpha/2}}$$

L'ampiezza del CI si definisce **precisione della stima**, e corrisponde a $2w$. Il *confidence level* α è **quantificato dal quantile corrispondente alla probabilità cumulata prefissata**, divisa a **metà**, della distribuzione **normale standardizzata** (grazie, teorema centrale del limite!). In realtà, è uso comune affidarsi ai **quantili della distribuzione t, per gradi di libertà $N-1$**, dato che esistono sempre buoni motivi per dubitare che una distribuzione campionaria sia realmente affine alla normale teorica – e comunque, sappiamo che al crescere di N i quantili t e z tendono a sovrapporsi.

[32] Mi raccomando: è un **errore empiricamente frequente** non ricordarsene e trarre conclusioni sballate sul fatto che la media (o la correlazione o quel che sia) del campione sia compresa nel CI in popolazione: è **sempre** così! **Deve** essere così!

A parità di SE, diversi livelli di fiducia creano diverse ampiezze, dato che al crescere di α, in valore assoluto, aumenta la dimensione dei quantili t:

```
t_05<-qt(p = .05,df = 50-1,lower.tail = FALSE)
[1] 1.676551
t_025<-qt(p = .025,df = 50-1,lower.tail = FALSE)  ← α= .05, 95%CI
[1] 2.009575
t_.005<-qt(p = .005,df = 50-1,lower.tail = FALSE) ← α= .01, 99%CI
[1] 2.679952
```

Quindi, **un CI al 99% dà una sicurezza maggiore, ma una precisione minore**, di trovare in ripetuti campionamenti il parametro atteso, rispetto a un CI al 95%.

```
CI_90<-c(20.5-(2.2*t_05),
20.5+(2.2*1.676551))
CI_90
[1] 16.81159 24.18841
CI_90[2]-CI_90[1]
[1] 7.376824
```

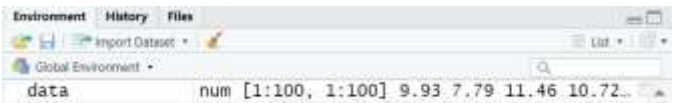
```
CI_95<-c(20.5-(2.2*t_025),
20.5+(2.2*2.009575))
CI_95
[1] 16.07894 24.92106
CI_95[2]-CI_95[1]
[1] 8.84213
```

```
CI_99<-c(20.5-(2.2*t_005),
20.5+(2.2*2.679952))
CI_99
[1] 14.60411 26.39589
CI_99[2]-CI_99[1]
[1] 11.79179
```

Attenzione: la maggior parte dei CI che tratteremo è **simmetrica** attorno alla statistica campionaria ($w_L = w_U$), ma **non è una regola generale**. Per esempio, i CI relativi a misure **nominali**, per cui si usa la distribuzione binomiale, non sono simmetrici, soprattutto se la proporzione campionaria è prossima a 0 o a 1; ne vedremo esempio, basato sui CI delle proporzioni, nel caso del test della binomiale (§6.7) e degli Odds Ratio – OR (capitolo 7). Altrettanto non simmetrici sono i CI delle **correlazioni** (capitolo 8), il cui calcolo richiede trasformazioni piuttosto complesse.

Nota bene: nei prossimi esempi, costruiremo empiricamente CI attorno alla **media**, ma nei capitoli successivi lo faremo per molte altre diverse statistiche campionarie: cambierà il modo **in cui si calcola l'errore standard dalla statistica**, ma, poiché lo faremo fare a R, potremo preoccuparcene poco (vedremo solo alcuni esempi di calcolo di SE diversi dallo SE della media, nessuno dei quali complesso).

Facciamo un esempio: usiamo la funzione `rnorm` per estrarre in maniera randomizzata 100 osservazioni ("soggetti") da una popolazione, normalmente distribuita, in cui il parametro $\mu = 10$ e $\sigma = 1.5$. Facciamo **cento repliche** di queste estrazioni, corrispondenti a 100 "esperimenti" fatti su campioni di uguale numerosità e provenienti dalla stessa popolazione: ci serviamo di `replicate(n= numero di repliche, expr= espressione da replicare)`, che abbiamo già usato per la DCM, con cui creiamo una matrice (**`data`**) composta da 100 colonne di 100 righe ciascuna.

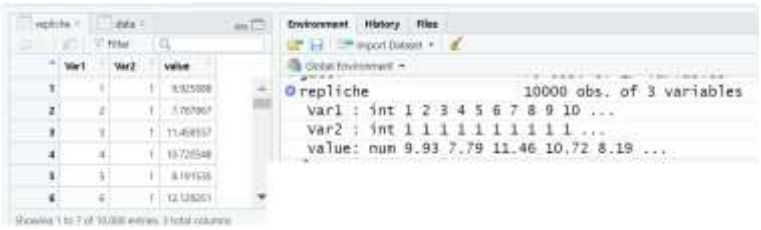



```
data<-matrix(replicate(n = 100, expr = rnorm(n
= 100, mean = 10, sd = 1.5)),nrow = 100, ncol =
100)
```

La media di **ciascuna** colonna rappresenta la media di una replica sperimentale. Però, per plottare rapidamente le 100 medie, disegnando attorno a ciascuna il rispettivo $95\%CI$, ci occorre **una** colonna, in cui siano contenute tutte le osservazioni, contraddistinte dal numero della replica in cui sono ricavate: questo corrisponde a **trasformare la**

struttura della matrice dall'attuale formato ***wide*** al formato ***long*** (ne abbiamo accennato nel §2.2 e ne riparleremo nel capitolo 10, per ora non fateci troppo caso).

Per questa trasformazione, usiamo `melt(data= dataframe / matrice da trasformare, measures.var= colonne da trasporre)` di `reshape2`, ottenendo il dataframe **`repliche`**:



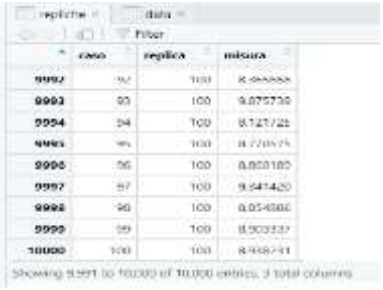

```
repliche<-melt(data, measure.vars = c("V1":"V100"))
```

`$var1` è il nome assegnato di default ai casi, `$var2` quello assegnato al numero delle repliche. Dobbiamo assegnare un nome alle colonne del dataframe , con `names`:

```
names(repliche)<-c("caso","replica","misura")
```

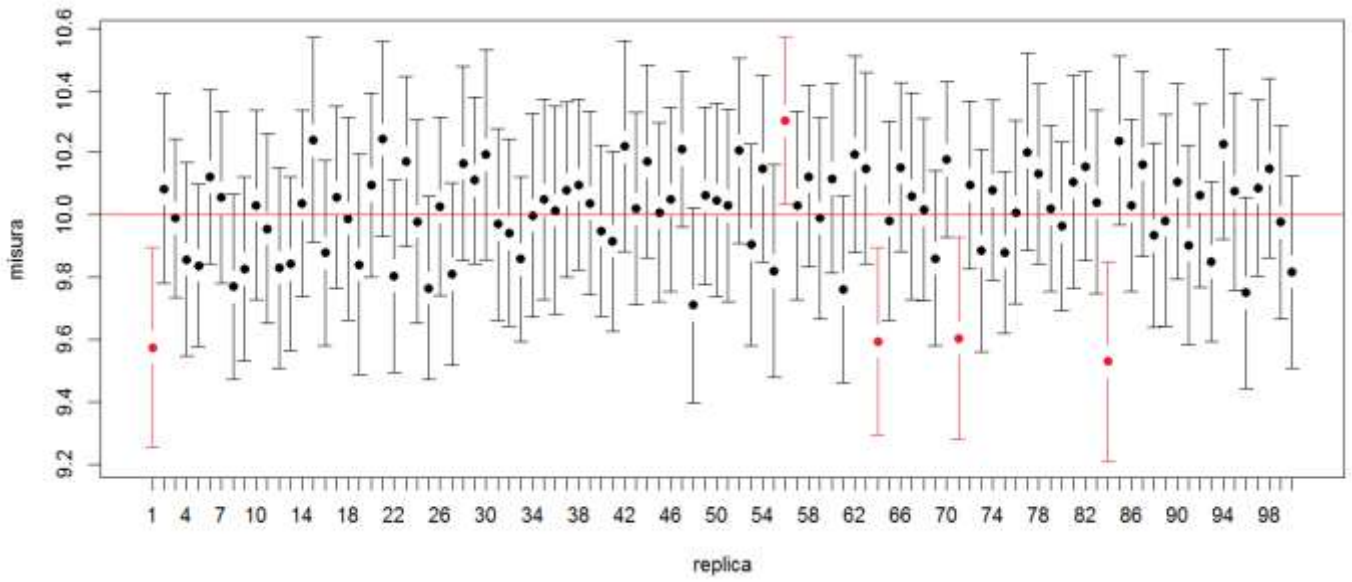
, poi cambiamo la classe della variabile `$replica`, che è stata considerata `integer`, ma in realtà è una variabile di raggruppamento `factor`:


```
repliche$replica <-
as.factor(repliche$replica)
```



Possiamo ora plottare medie e CI. Tra le varie possibilità, proviamo con `plotmeans(misura~fattore, bars= TRUE)`, del package `gplots`, che produce un grafico delle medie e barre di errore costituite dai CI; di default, il confidence level è al 95% (`p= .95`) ed è calcolato con i quantili t, ma è possibile usare i quantili z indicando `use.t= FALSE`. Tra le tante altre opzioni grafiche, eliminiamo le linee che connettono le medie (il grafico è già fitto così) con `connect= FALSE`, e decidiamo che il simbolo della media sia un pallino nero pieno (`col= "black", pch=19`); rendiamo nere anche le barre dei CI (di default `barcol="blue"`), perché useremo i colori per differenziare gli intervalli che comprendono il valore atteso $\mu = 50$ da quelli che non lo racchiudono; eliminiamo anche il numero della replica entro il grafico, perché sono sufficienti le etichette dell'asse x (`n.label=FALSE`). Infine, tracciamo la linea che identifica il parametro $\mu = 50$ con `abline(h= valore di μ)`.

```
plotmeans(repliche$misura~repliche$replica, connect = FALSE,bars = TRUE,xlab = "replica",ylab =
  "misura",barcol = "black",use.t = TRUE, pch=19, n.label = FALSE)
abline(h=10, col="red")
```



*Provate a fare altre simulazioni: naturalmente, dato che le **estrazioni sono casuali**, i risultati saranno leggermente diversi per ciascuno di voi:*

- *replicate le 100 simulazioni con N=100, μ= 10, σ= 1.5: cosa osservate?*
- *a parità di tutti gli altri parametri, cambiate il confidence level con α .99: cosa osservate?*
- *per α= .95 e a parità di μ e σ, **diminuite N a 50 e poi a 30: cosa osservate?***

Nota bene: per ottenere la **stessa sequenza di numeri casuali** (in realtà pseudocasuali!) in diverse estrazioni random, si può fissare il "seme" di partenza, con `set.seed(numero casuale)`.

Per esempio, **senza** `set.seed` avremo:

```
round(rnorm(n = 10, mean = 0,sd = 1),3)
 [1]  1.224  0.360  0.401  0.111 -0.556  1.787  0.498 -1.967  0.701 -0.473
round(rnorm(n = 10, mean = 0,sd = 1),3)
 [1] -1.068 -0.218 -1.026 -0.729 -0.625 -1.687  0.838  0.153 -1.138  1.254
```

eccetera, mentre fissando il "seme":

```
set.seed(seed = 123)
round(rnorm(n = 10, mean = 0,sd = 1),3)
 [1] -0.560 -0.230  1.559  0.071  0.129  1.715  0.461 -1.265 -0.687 -0.446
set.seed(seed = 123)
round(rnorm(n = 10, mean = 0,sd = 1),3)
 [1] -0.560 -0.230  1.559  0.071  0.129  1.715  0.461 -1.265 -0.687 -0.446
```

Se avete fatto l'esercizio precedente, avrete notato che, riducendo i df, il quantile t per la stessa probabilità cumulata aumenta, e, poiché viene moltiplicato per l'errore standard per costruire i CI, a **un minor numero di soggetti corrisponde sempre una maggior ampiezza del CI** – e quindi una minore precisione, a parità di probabilità, variabilità e media campionaria.

Vediamo ora qualche esempio di gestione dei CI di dati reali, applicandoli alla **media**.

*Da qui procediamo usando il dataframe **`adolescenti`**; scaricatelo da Elly, insieme alla descrizione delle variabili che lo compongono: leggete la descrizione e aprite il dataframe in R, prima di proseguire.*

Per accorciare la lunghezza del testo delle funzioni, creiamo ad. associandovi le caratteristiche di adolescenti; descriviamo le informazioni essenziali:

```
ad<-adolescenti
length(ad$soggetti)
[1] 1274
summary(ad$eta);sd(ad$eta)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  15.00   16.00   17.00   17.24   18.00   22.00
[1] 1.014791
round(prop.table(table(ad$genere))*100,1)
F    M
64.7 35.3
table(ad$istituto)
BOC BOD GIP   L  MA  ME ROM RON  SV  TO
 52  91 120  48 138 132 118  88 124 363
```

Abbiamo 1274 adolescenti, di età compresa tra 15 e 22 anni (quindi alcuni sono ben più che adolescenti...), con età media pari a 17.2 ± 1.1 anni; il problema dello sbilanciamento per genere è grave, dato che il 64.7% dei partecipanti è di genere femminile. Sono stati coinvolti 10 istituti scolastici, con una numerosità per istituto decisamente variabile: da un minimo di 52 a un massimo di 363.

*La ricerca era interessata alla gravità e alla **frequenza dei comportamenti a rischio**: in questi dati avete il **numero totale** **di comportamenti a rischio** per la salute (`a$comportamenti_rischio`) ammessi dai ragazzi. Prima di proseguire, descrivete i comportamenti a rischio dichiarati e interpretateli: anzitutto nel campione complessivo, poi separatamente per ragazzi e ragazze, infine separatamente per Istituto. Fate attenzione: è un campione molto **numeroso** (oltre 1000 casi), ma ci sono anche molti NA.*

Concentriamoci **sulla loro personalità**: hanno compilato il Temperament and Character Inventory (**TCI**, Cloninger, 1998), che valuta tratti temperamentali e caratteriali e che ritroveremo in altre ricerche. Ci interessano i tre tratti temperamentali fondamentali: **Novelty Seeking** – **NA** (tendenza a reagire con eccitazione agli stimoli o situazioni che comportano novità, necessità di alti livelli di stimolazione, tendenza all'esplorativà e all'entusiasmo, facilità ad annoiarsi, inclinazione all'impulsività), **Harm Avoidance** – **HA** (tendenza a rispondere intensamente agli stimoli negativi, preoccupazione per le possibili conseguenze delle proprie azioni, cautela, apprensività e sensibilità alle critiche ed alle punizioni), **Reward Dependence** – **RD** (tendenza a rispondere intensamente alle situazioni che comportano una ricompensa o gratificazione, soprattutto ai segnali di approvazione sociale e affettivi, o alle offerte di aiuto). Il manuale del test fornisce i **punteggi normativi attesi nella popolazione di ragazzi di pari età**: $NS: \mu = 20.2 \pm 6.6$, $HAS: \mu = 14.9 \pm 7.7$, $RD: \mu = 17.4 \pm 3.9$.

Come si pongono questi ragazzi rispetto alla popolazione di pari età? Rappresentano fedelmente le tre dimensioni dei coetanei? Sono un **modello attendibile della personalità degli adolescenti**? Per rispondere a questa domanda, stimiamo la media campionaria, tracciamo il CI al 95% e confrontiamola con la media attesa, dimensione per dimensione. Esercitiamoci nel calcolo passo passo con la dimensione NS; per le altre due, ricorreremo alle scorciatoie. Prendiamo solo la variabile NS, escludendo i dati mancanti (così non dobbiamo preoccuparci della loro gestione nelle funzioni successive):

```
ado_NS<-ad$NS_tot[is.na(ad$NS_tot)==FALSE]
```

Vediamo i tre elementi campionari del CI: N, media, deviazione standard:

```
length(ado_NS);mean(ado_NS);sd(ado_NS)
[1] 1269
[1] 17.46493
[1] 4.679953
```

Da sd e N ricaviamo l'errore standard; salviamolo in un oggetto:

```
SE_NS<- sd(ado_NS)/sqrt(length(ado_NS))
[1] 0.1313744
```

Lo SE della media si può ottenere anche con `MeanSE(distribuzione)` di `DescTools`:

```
MeanSE(ado_NS)
[1] 0.1313744
```

Scegliamo un confidence level $\alpha = .95$; il quantile t per $df = 1269 - 1 = 1268$ e $p = .025$ è:

```
(t_95<-qt(p = 0.025,df = 1269-1,lower.tail = FALSE))
[1] 1.961837
```

Calcoliamo il lower limit (LL) e l'upper limit (UL):

```
LL<-mean(ado_NS)-(SE_NS*t_95)
UL<-mean(ado_NS)+(SE_NS*t_95)
LL;mean(ado_NS);UL
[1] 17.2072
[1] 17.46493
[1] 17.72267
```

Il tratto di personalità NS nel campione è pari a 17.465 ± 4.679; il parametro NS atteso nella popolazione da cui abbiamo estratto questi ragazzi è compreso, con una verosimiglianza del 95% tra 17.207 e 17.723. Il CI è simmetrico:

```
mean(ado_NS)-LL; UL-mean(ado_NS)
[1] 0.2577351
[1] 0.2577351
```

L'ampiezza w_2 del CI è ristretta, pari a circa mezzo punto: la precisione della stima è piuttosto buona:

```
print(w2<-(mean(ado_NS)-LL)*2)
[1] 0.5154702
```

Ora che sappiamo come si fa, nessuno ci biasimerà se useremo funzioni preimpostate per calcolare un CI: non ce ne sono tra le funzioni di base, ma già conosciamo `Desc(Y~X)` di `DescTools`, che tra molte informazioni dà anche il CI delle medie (con quantili t). Più sintetica è **`MeanCI(distribuzione)`**, dello stesso package, che riporta esclusivamente quanto promette (default `conf.level=.95`), ma con una buona flessibilità: oltre ad α, si può stabilire anche il metodo con cui è calcolato (quello che usiamo qui è `method= "classic"`, ma c'è anche il *bootstrapping*: `method= "boot"`), l'uso di una media *trimmed* invece della media aritmetica, utile in presenza di outlier: `trim= proporzione da troncare`. Se non viene specificato l'argomento `sd=`, lo SE sarà stimato dalla deviazione standard della distribuzione e saranno usati i quantili t; se invece la σ è nota, si inserisce in `sd=` e saranno usati i quantili z. Infine, se ci sono dati mancanti, è indispensabile specificare `na.rm= TRUE`. Avremo quindi:

```
MeanCI(ado_NS)
    mean   lwr.ci   upr.ci
17.46493 17.20720 17.72267
```

```
MeanCI(a$NS_tot, na.rm=TRUE)
    mean   lwr.ci   upr.ci
17.46493 17.20720 17.72267
```

Il manuale del TCI fornisce la sd della popolazione normativa: possiamo inserire quest'informazione nella funzione e calcolare il $95\%CI$ usando la distribuzione di probabilità normale standardizzata:

```
MeanCI(ado_NS,sd = 6.6)
    mean   lwr.ci   upr.ci
17.46493 17.10180 17.82806
```

Non stupisce che i due CI siano estremamente simili, dato che il quantile t di una distribuzione di probabilità t con df= 1268 è praticamente identico al quantile z di una distribuzione di probabilità normale standardizzata corrispondente alla stessa funzione di ripartizione:

```
qt(p = .95,df = 1268,lower.tail = FALSE); qnorm(p = .95,0,1,lower.tail = FALSE)
[1] -1.646056
[1] -1.644854
```

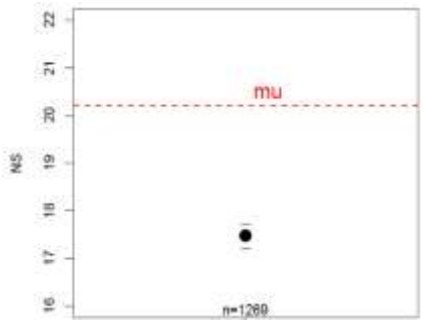
Confermato che non abbiamo sbagliato i calcoli, passiamo alla domanda fondamentale: questo campione di adolescenti è **rappresentativo** della popolazione di coetanei, rispetto al parametro Novelty Seeking?

- **Se nel CI (per una verosimiglianza di almeno .95) della statistica campionaria è compreso il parametro atteso in popolazione, il campione è probabilmente estratto dalla popolazione**, e la sua media è probabilmente solo una **fluttuazione casuale di μ,** con una verosimiglianza del 95%.
- **Se nel CI (per una verosimiglianza di almeno .95) della statistica campionaria non è compreso il parametro atteso in popolazione, il campione è probabilmente estratto da un'altra popolazione**.

Verifichiamo quindi la **significatività della differenza tra un campione e una popolazione**, che riprenderemo da una diversa – ma relativamente coerente - ottica nel §6.7. Dal manuale del TCI, sappiamo che in popolazione $\mu_{NS} = \mathbf{20.2}$: poiché il $95\%CI = 17.21 - 12.72$ del campione non comprende il valore atteso, i nostri adolescenti **non sembrano appartenere alla popolazione normativa**; si può dire che il campione sembra significativamente differente dalla popolazione, ovvero appartenente a una diversa popolazione, meno amante del rischio, di quella considerata nella taratura del test.

Plottiamo. `plotmeans` di `gplots` si aspetta un fattore nella formula `misura~fattore`, che noi non abbiamo: creiamone uno falso, ripetendo 1269 volte "fake":

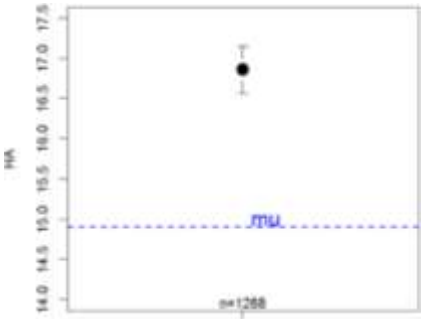
```
fattore<-as.factor(rep("fake",1269))
plotmeans(ado_NS~fattore, pch=19, ylim=c(16, 22), lwd=2, barcol =
"black", ylab="NS", xlab="campione", cex=2)
abline(h=20.2, lty=2, lwd=2, col="red")
text(labels = "mu",x = 1,y=20.5, pos = 4, col="red", cex=1.5)
```



È decisamente evidente che μ_{NS} non è compresa nel CI campionario.

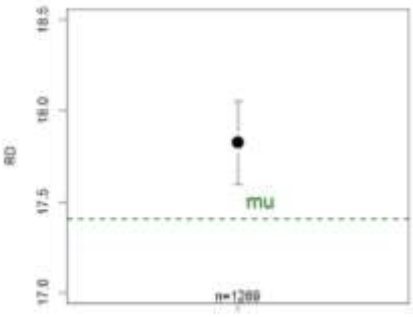
Vediamo ora le altre due dimensioni:

```
ado_HA<-a$HA_tot[is.na(a$HA_tot)==FALSE]
MeanCI(ado_HA)
    mean   lwr.ci   upr.ci
16.86356 16.57363 17.15350
```



Poiché in popolazione $\mu_{HA} = \mathbf{14.9}$, il campione sembra significativamente differente dalla popolazione anche per il parametro HA. La precisione della stima è buona.

```
ado_RD<-a$RD_tot[is.na(a$RD_tot)==FALSE]
MeanCI(ado_RD)
    mean   lwr.ci   upr.ci
17.82506 17.59627 18.05384
```



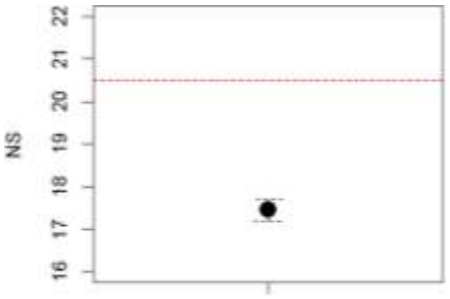
Poiché in popolazione $\mu_{RD} = \mathbf{17.4}$, il campione sembra significativamente differente dalla popolazione anche per il parametro RD. Anche in questo caso, la precisione è buona.

Forse, è il momento di rivelare che i dati normativi si riferiscono a una popolazione di studenti **statunitensi** ☺

Esistono altre funzioni per rappresentare i CI; abbiamo usato **`plotMeans`** di `RcmdrMisc`, raffigurando le deviazioni standard come barre di errore, ma possiamo usarla anche per i CI: `plotMeans(response, factor1, error.bars=`

"conf.int", level=.95, connect= TRUE/FALSE). L'argomento response= richiede la misura di cui si calcola la media, factor1= i livelli del fattore per ciascuno dei quali si calcola la media (si può aggiungere anche un factor2, motivo per cui useremo questa funzione nell'ANOVA con due predittori, capitolo 13); error.bars= "conf.int" indica di tracciare l'intervallo di confidenza attorno alla media, per una verosimiglianza specificata da level (default =.95); connect=TRUE/FALSE indica se connettere le medie con una linea oppure no (di default, TRUE).

```
plotMeans(response = ado_NS, factor1 = fattore, error.bars =
 "conf.int", level = .95, connect=FALSE,pch = 19,
 xlab="",ylab="NS", ylim=c(16,22))

abline(h=20.5, lty=3, col="red", lwd=2)
```



Un altro modo per conoscere il CI della media di una distribuzione è usare la funzione di base **t.test(distribuzione, conf.level= .95)**: la useremo ripetutamente, per falsificare ipotesi che richiedono test basati sulla distribuzione t. Il suo scopo non è fornire (solo) il CI, ma per ora potete ignorare tutto il resto dell'output e concentrarvi su media e CI. Per esempio:

```
t.test(a$HA)
        One Sample t-test
data:  a$HA
t = 114.11, df = 1267, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 16.57363 17.15350
sample estimates:
mean of x
 16.86356
```

Ne discuteremo ampiamente nel §6.6.1, quando affronteremo tutte le altre informazioni presenti.
Riprenderemo l'uso dei CI in aggiunta – o in sostituzione – dei $p - value$ associati alle statistiche campionarie, nel §6.6, e torneremo sulla rappresentazione grafica nella verifica delle ipotesi nei §10.1.2 – 10.2.2 (t-test per dati indipendenti e dati appaiati).

6.4 Inferenza e verifica delle ipotesi: Null Hypothesis Significance Test (NHST)

Sic enim se profecto res habet, ut numquam perfectam veritatem casus imitetur.
Cicerone, De Divinatione, Libro I

*Statistical rituals largely eliminate statistical thinking in the social sciences […] What I call the "**null ritual**" consists of three steps: (1) set up a statistical null hypothesis, but do not specify your own hypothesis nor any alternative hypothesis, (2) use the 5% significance level for rejecting the null and accepting your hypothesis, and (3) always perform this procedure.*
Gigerenzer, Mindless statistics, 2004

L'approccio alla verifica delle ipotesi attualmente più utilizzato (ma anche più criticato) nelle scienze sociali è l'approccio **Null Hypothesis significance Test** (**NHST**). È un metodo ibrido, prodotto dalla fusione piuttosto infelice di due approcci diversi, facenti capo a tre statistici con rapporti piuttosto tesi: da una parte (sir) **Ronald Fisher**, che propone per primo (1925) il $p - value$ **approach (PVA)**, dall'altra **Egon Pearson** (figlio di Karl Pearson, che troveremo ripetutamente nelle nostre statistiche) e Jerzy **Neyman**, che avanzano una integrazione all'approccio di Fisher definito **Fixed alpha**

approach (1928 e successivi). Il conflitto di lunga data tra Fisher e Pearson padre si è trasferito alla generazione successiva, dato che Fisher aveva memoria lunga e un carattere poco amabile: le difficili relazioni umane non hanno giovato all'integrazione statistica dei due approcci.

PVA e FAA sono abbastanza superficialmente simili da essere, purtroppo, facilmente confusi, nonostante gli autori (Fisher, soprattutto) fossero perfettamente consapevoli delle differenze teoriche sottostanti e pronti a dichiararle senza remore. Ciò non ha impedito la nascita dell'approccio **NHST**, filosoficamente confuso, essendo l'ibridazione di costrutti in buona parte differenti, ma tecnicamente *appealing*, proprio perché facilita un'inappropriata decisione tutto–o–nulla sulla propria ipotesi alla luce dei risultati ottenuti, nonostante PVA e FAA insistano sulla necessità di prendere decisioni e fare scelte ragionate in tutti i passi del processo di verifica delle ipotesi.

Accenniamo qui a una breve esplorazione dei due approcci, e a come siano stati incautamente mescolati nell'approccio NHST; nel paragrafo §6.7 approfondiremo le critiche a NHST e le opportune integrazioni (o alternative) proposte. Per chi voglia approfondire, si suggerisce l'ottimo articolo di Perezgonzalez[33] (2015), da cui soprattutto è stato preso lo specchietto di confronto a fine paragrafo.

6.4.1 P-Value Approach

Fisher trasferisce la **dimostrazione per assurdo** (*reductio ad absurdum*) dal mondo logico e deterministico del ragionamento **deduttivo** al mondo (empirico e quindi al più probabilistico) del ragionamento **induttivo**:

> Se, data una premessa [ipotesi] come vera, ne discende una conclusione logica contraddittoria, allora la premessa [ipotesi] è falsa.

Questa dimostrazione viene tradotta in:

> Se, dato un modello come vero [**l'ipotesi nulla**], i risultati empirici sono **in evidente contraddizione** con le sue previsioni [sono **troppo poco probabili secondo le previsioni del modello**], allora il modello [ipotesi nulla] è respinto - ***disproved***.

Con "**ipotesi nulla**" intendiamo, nel senso più generale, una **serie di affermazioni su come funzionano le cose in popolazione**, in particolare su un determinato valore di un parametro della popolazione. Fisher ne conclude quindi che: "note le caratteristiche di un **universo [popolazione]**, se, tratto da esso un **campione**, **questi viola le nostre aspettative**, possiamo inferire che è stato tratto da una **diversa** popolazione"[34].

Il metodo di *significance testing* o **test di significatività** punta a calcolare il **valore di probabilità del risultato empiricamente osservato** [o **uno ancora più estremo,** quindi con una **probabilità ancora minore**], **sotto condizione di ipotesi nulla**, ovvero **ponendo come assunto che l'ipotesi nulla sia vera**: questo valore di probabilità viene definito $p - value$. Se il $p - value$ associato al risultato (o a uno più estremo) è **grande**, allora è probabile che il modello (l'ipotesi nulla) sia adeguato a spiegare la realtà; se invece il $p - value$ associato al risultato (o a uno più estremo) è **piccolo**, allora è probabile che il modello (l'ipotesi nulla) non sia adeguato alla realtà, e pertanto **non viene confermato**.

Dice Fisher: "Every experiment may be said to exist **only in order to give the facts a chance of disproving the null hypothesis**".

Il procedimento può essere riassunto in cinque passi:

---

[33] Perezgonzalez, J.D. (2015). Fisher, Neyman-Pearson or NHST? A tutorial for teaching data testing. *Frontiers in Psychology*, doi: 10.3389/fpsyg.2015.00223

[34] *Statistical methods for research workers (1925): http://psychclassics.yorku.ca/Fisher/Methods/*

1. **Scegliere il test adeguato**, a seconda degli scopi della ricerca e del modo in cui sono state misurate le variabili d'interesse: come vedremo, se l'obiettivo è stimare relazioni simmetriche tra due variabili potremo scegliere tra: test χ^2 a due vie se le variabili sono nominali, coefficienti di cograduazione se sono ordinali, coefficienti di correlazione se sono metriche. Se l'obiettivo è stimare una differenza fra due gruppi indipendenti potremo scegliere tra t-*test* per campioni indipendenti o analisi della varianza o molto altro... e così via, anche alla luce di ulteriori restrizioni d'uso per ogni test.
2. **Fissare l'ipotesi nulla H_0** in base al test scelto: per esempio, se intendiamo valutare la differenza tra le medie di due gruppi, l'ipotesi nulla affermerà che la differenza tra le due medie è uguale a zero, ovvero che i due gruppi non sono differenti: $H_0: M_1 - M_2 = 0$; più genericamente, il **parametro** oggetto dell'ipotesi nulla è definito come **theta**, $\boldsymbol{\theta}$. Alcuni dei parametri di H_0 sono derivati dai dati (la varianza, i gradi di libertà...), altri sono assunti teoricamente (come la forma della distribuzione delle frequenze). Attenzione: come ribadiremo, non è sempre vero che H_0 debba prevedere uno "0", anche se può essere effettivamente il caso più frequente. Sempre riferendoci all'esempio relativo alle medie di due gruppi, possiamo porre come ipotesi nulla che la differenza tra loro, espressa in deviazioni standard (standardizzata), **non è superiore a 1**: $H_0: z_{M_1} - z_{M_2} \leq |1|$. L'ipotesi nulla può quindi essere espressa come:
   - **Bidirezionale / a due code:** $\boldsymbol{H_0: \theta_1 = \theta_2}$
   - **Monodirezionale destra / a una coda, destra:** $\boldsymbol{H_0: \theta_1 \leq \theta_2}$
   - **Monodirezionale sinistra / a una coda, sinistra:** $\boldsymbol{H_0: \theta_1 \geq \theta_2}$
3. **Calcolare la probabilità teorica del risultato ottenuto sotto condizione di ipotesi nulla**: assumendo che H_0 sia vera, si ricava il $p - value$ del dato ottenuto **e anche** di risultati più estremi: nell'esempio delle due medie, stimiamo la probabilità che si verifichi la differenza verificata nei dati, o una maggiore, assunta l'ipotesi nulla. Ricordate la funzione di ripartizione? Il $\boldsymbol{p - value}$ **NON è una probabilità puntuale, ma una probabilità cumulata**, che si estende dal risultato ottenuto alla coda della distribuzione.
4. **Stabilire la significatività statistica del risultato**: Fisher identifica i risultati **interessanti** come quelli con una bassa probabilità di verificarsi come semplici fluttuazioni casuali di un'ipotesi nulla. Giudica **conveniente** un $p - value = .05$ come limite per stabilire se l'evento sia davvero eccezionale: "it's usual and **convenient** for experimenters to take this point as a limit in judging whether a deviation is to be considered significant or not […], **as a standard level of significance**, in the sense that **they are prepared to ignore all results which fail to reach this standard** and [….] to eliminate from further discussion the greater part of the fluctuations which chance causes have introduced in their experimental results. We shall not **often be astray** if we draw a **conventional** line at .05"[35]. Però, afferma anche che "if one in twenty [.05] does not seem high enough odds, we may […] draw the line at one in fifty (.02), or one in a hundred (.01)". Questo comporta che i limiti convenzionali non devono **mai essere usati acriticamente per prendere decisioni dicotomiche** sull'accettare o rifiutare H_0: lo rivedremo nel §6.6. La conclusione sulla significatività / interesse del risultato può anche **essere lasciata al lettore**: per questo motivo, nei risultati di una ricerca andrebbero riportati i $p - value$ esatti (Gigerenzer, 2004).
5. **Interpretare la significatività statistica del risultato**: di fronte a un $p - value$ inferiore allo "standard level of significance" precedentemente stabilito, Fisher direbbe che: "either an **exceptionally rare** chance has occurred, or the **theory is not true**". D'altronde, Cicerone aveva posto un problema simile, ma senza trarne diagrammi e

[35] *Ibidem,* pp. 47,79.

formule, un paio di millenni prima[36]: “Davvero può accadere per caso ciò che ha i caratteri della verità? Quattro dadi danno il colpo di Venere [quattro facce tutte diverse] per caso: ma **se lanci quattrocento dadi, e il colpo di Venere appare cento volte, potresti ancora dire che è un caso?** [...] Una scrofa con il suo grugno avrà tracciato sul terreno la lettera A: per questo la crederai capace di scrivere l’Andromaca di Ennio? [...] Le cose, non c’è dubbio, stanno esattamente così: **il caso non può mai imitare perfettamente la verità**.” Questa potrebbe essere una delle prime circostanze in cui avremmo potuto accettare o respingere l’ipotesi nulla.

Attenzione: Anche nel caso in cui i dati **non siano in contraddizione** con il modello**,** ovvero in presenza di un *p*-value superiore al livello di significatività e che non disconferma H_0, **non è comunque possibile affermare di aver dimostrato che il modello è vero!** Si può solo dire che i dati campionari non offrono sufficiente evidenza per rifiutare l’ipotesi nulla / il modello: ***Absence of evidence is not evidence of absence***. D’altro canto, di fronte a un $p - value$ inferiore al livello significatività, che non porta sufficiente evidenza a H_0 e quindi la disconferma, non si può semplicisticamente affermare che sia stata dimostrata proprio l’ipotesi **alternativa** a quella disconfermata (§6.4.2).
Quindi, si potrebbe obiettare, a che serve realmente un $p - value$? In effetti, a poco o nulla, replicano diversi autori che vedremo nel §6.7. D’altronde, lo stesso Fisher afferma che l’unico modo per fare inferenze sensate basate sui risultati di un test di significatività è il **controllo sul disegno di ricerca**, in particolar modo la randomizzazione dei casi (1955), e insiste sia sul fatto che un solo risultato significativo deve essere considerato esclusivamente come un punto di partenza, da **replicare** in ulteriori ricerche (1954), sia sulla necessità di usare **meta-analisi** (§6.5) che combinino risultati a sostegno o a disconferma di H_0, provenienti da ricerche correlate (1960).

Facciamo un esempio terra terra basato sulla **rappresentatività di un campione rispetto alla popolazione** da cui è estratto, come nel paragrafo precedente: se i risultati del campione saranno in contraddizione con quelli attesi in base alla popolazione, ovvero saranno troppo poco probabili per essere solo variazioni casuali rispetto a quanto previsto, allora si **dovrà respingere l’ipotesi che il campione sia rappresentativo della popolazione**.
Per puro amore della scienza, compriamo 36 barattoli magnum di Nutella **scelti a caso** per verificare se loro peso corrisponda a quello dichiarato in etichetta per il loro formato (**3Kg**→ peso della popolazione cui appartengono i barattoli): **l'ipotesi nulla** afferma che il peso dei 36 barattoli è uguale a quello della popolazione cui appartengono, ovvero =3Kg. Però, quando pesiamo i 36 barattoli, scopriamo che il loro peso medio è = 2.92Kg (**peso medio della distribuzione campionaria**). È evidente che $\mathbf{2.92 \neq 3}$ (infatti, nella **realtà empirica**, l'ipotesi nulla, in un'accezione restrittiva di perfetta identità, è **sempre falsa**); dobbiamo però tener conto della **variabilità campionaria** (e degli errori di misura), per cui riformuliamo la domanda cui rispondere:

La **differenza** tra la **statistica** campionaria osservata (2.92 Kg) e il corrispondente **parametro** nella popolazione (3Kg) è tanto **piccola** da poter essere considerata una **fluttuazione casuale del parametro?**

Ovvero:

Quanto è probabile che si possa osservare **solo per caso** una **differenza** $|\bar{x}_{DCM} - \mu|$ **pari o superiore** a quella osservata, se i campioni appartengono effettivamente alla popolazione (ovvero sotto condizione di ipotesi nulla)?

[36] De Divinatione, libro I, XIII: “Quicquam postest esse casu esse factum, quod omnes habet in se nomeros veritatis? quattuor tali iacti casu Venerium efficiunt; num etiam centum Venerios, si quadrigentos talos ieceris, casu futurus putas? […] Sus rostro si humi A litteram impresserit, mum propterea suspicari poteri Andromacham Enni ab ea posse decribi? [...] Sic enim se profecto res habet, ut numquam perfectam veritatem casus imitetur”.

Tanto più grande è questa probabilità, tanto più probabilmente il campione è effettivamente tratto dalla popolazione; tanto più piccola è questa probabilità, tanto più probabilmente il campione non apparterrà alla popolazione.
Per attribuire una probabilità alla $\bar{x}_{DCM} = 2.92$ di verificarsi sotto condizione d'ipotesi nulla, ricorriamo alle note proprietà della distribuzione campionaria delle medie (DCM); diciamo che **la deviazione standard è $s = .18$**. Quindi, la prima cosa da fare è **calcolare l'errore standard**:

```
print(SE<-.18/sqrt(36))
[1] 0.03
```

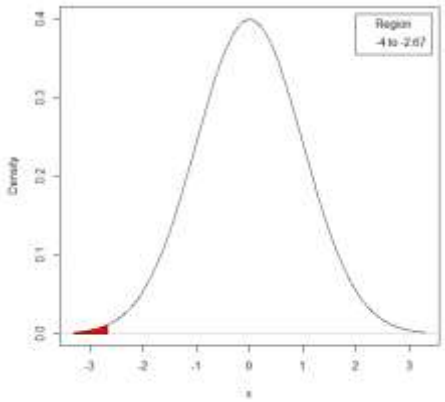
Ora possiamo **conoscere la probabilità cumulata** pari al **quantile 2.92 o a un peso inferiore**, in una distribuzione normalmente distribuita con $\mu = 3$ e $\sigma = .03$, grazie alla **funzione di ripartizione**; applicandola alla distribuzione normale, e badando di usare l'argomento `lower.tail= TRUE` avremo:

```
pnorm(q = 2.92, mean = 3, sd = .03, lower.tail = TRUE)
[1] 0.003830381
```

Quindi, la probabilità che da una popolazione con peso medio $\mu = 3$ siano tratti barattoli la cui media è pari o inferiore a 2.92, tenendo conto della variabilità campionaria, è **molto bassa**, inferiore al 4 per mille. Diversamente detto, la probabilità che un peso medio $\bar{x} = 2.92$ sia solo una **fluttuazione casuale** della media $\mu = 3$ è **molto bassa**.

Se standardizzassimo il quantile 2.92 per visualizzarlo in una distribuzione normale standardizzata, avremmo:

```
z_peso<-(2.92-3)/SE
z_peso
[1] -2.666667
pnorm(-2.667, 0,1, lower.tail = TRUE)
[1] 0.003826584
```



Dobbiamo ora decidere se il risultato è *"noteworthy"*, interessante, significativo, oppure una mera fluttuazione casuale della differenza prevista da H_0: $\bar{x} - \mu = 0$. Anche accettando le proposte di eccezionalità più prudenti, sembra che il nostro risultato sia davvero eccezionale. L'ipotesi che i barattoli di Nutella pesino solo casualmente meno di quanto previsto in popolazione non gode di molto appoggio sulla base di questo risultato; però, prima di fare causa all'azienda produttrice, dovremmo seguire il suggerimento di Fisher e replicare il risultato, oppure controllare in letteratura se e quante altre verifiche del genere siano state fatte, e quali siano stati i loro risultati, combinandoli in una meta-analisi.

6.4.2 Fixed Alpha Approach Hypothesis Testing

Una delle più importanti critiche all'approccio di Fisher riguarda la mancanza di un'esplicita ipotesi alternativa, dato che sembra inutile rifiutare un'ipotesi nulla se non è disponibile una spiegazione alternativa (Gigerenzer, 2004). In realtà, Fisher considera **implicitamente** le ipotesi alternative, come negazioni di H_0 ($\neg H_0$), dato che per lui il più importante compito del ricercatore è quello di respingere sistematicamente, con sufficiente evidenza, le corrispondenti H_0.
Il Fixed Alpha Approach (FFA, 1928) è stato proposto da Egon Pearson e Jerzy Neyman a integrazione del $p - value$ approach, anche se si è successivamente trasformato in una e propria alternativa al primo. Il suo apporto più innovativo è proprio l'introduzione di una **esplicita ipotesi alternativa** H_A:

"**Assunto che un dato campione sia tratto da una popolazione**, il fenomeno che stiamo considerando non sarà significativamente differente dal corrispondente parametro della popolazione nel caso sia vera H_M (**main hypothesis** o ipotesi principale), mentre **differirà nel caso sia vera H_A".**

Il procedimento di verifica è diviso in cose da stabilire a priori, prima di raccogliere i dati, e (in numero molto minore) a posteriori.

A PRIORI:

1. Si **dichiara l'effect size atteso in popolazione.** Nella formulazione più semplice, H_A rappresenta una seconda popolazione (ad esempio, soggetti con ritardi nello sviluppo cognitivo) che si dispone a fianco della popolazione oggetto dell'ipotesi principale (ad esempio, soggetto con sviluppo cognitivo tipico), sullo stesso continuum di valori (i punteggi QI a un test di intelligenza). Queste due popolazioni **differiscono sul continuum di una certa quantità**: la **differenza è l'effect size o *ES*** (il concetto di ES è stato ripreso e diffuso da Cohen: lo ritroveremo nel §6.7): tanto minore è l'ES, cioè la differenza nel continuum tra le due popolazioni, tanto più sarà difficile rilevarlo. Spostandoci dalla differenza tra popolazioni alle differenze tra i campioni che dovrebbero rappresentarle, le distribuzioni campionarie hanno errori standard minori: ritroviamo l'ES, con lo stesso significato interpretativo, dato che le medie in popolazione non sono toccate, ma le distribuzioni campionarie appariranno più separate che sovrapposte, con un minore errore standard. È l'ipotesi alternativa H_A a fornire informazioni sull'effect size atteso: però, in questo approccio è l'ipotesi principale H_M che viene messa alla prova, non H_A: quindi, il metodo FAA ignora in gran parte la distribuzione H_A, tranne una piccola percentuale della sua area, definita **beta** o β. Beta è il **minimum effect size** (**MES**), cioè quella parte di H_M che non sarà respinta dal test
2. Si **seleziona il test ottimale**: al di là dei vincoli legati a obiettivi e livelli di misura già indicati per PVA, in questo approccio si **preferisce scegliere il test più potente**, cioè quello che **ha maggiori probabilità di rilevare l'effect size** (riprenderemo la potenza nel punto 7: i test parametrici sono più potenti dei non parametrici; le ipotesi unidirezionali sono più potenti di quelle bidirezionali; alcune condizioni sperimentali), e usare **condizioni sperimentali che incrementino la potenza** (ad esempio, aumentando la numerosità del campione).
3. Si **definisce l'ipotesi principale H_M** in questo approccio si hanno almeno due ipotesi in competizione, ma ne viene verificata solo una, quella **più importante** (Neyman, 1942). H_M incorpora il MES:

$$H_M: M_1 - M_2 = 0 \pm MES$$

per esplicitare che i valori entro $0 \pm MES$ sono considerati ragionevolmente probabili secondo l'ipotesi principale, mentre i valori esterni a $0 \pm MES$ sono ritenuti più probabili sotto condizione di H_A. È facile notare che H_M è molto simile all'H_0 di Fisher: in effetti, Pearson e Neyman l'hanno chiamata "*null hypothesis*" e l'hanno spesso definita in modo simile, ma non vanno confuse:

   - H_M deve essere esplicitata nel momento in cui si pianifica la ricerca, mentre H_0 è perlopiù implicita;
   - H_M è definita in modo da comprendere qualsiasi valore inferiore al MES, mentre gli ES non fanno parte dell'approccio di Fisher;
   - è solo una di due ipotesi in competizione per emergere come esplicative dei risultati della ricerca, mentre H_0 è l'unica protagonista nel PVA.

   L'aspetto più importante da considerare durante la definizione di H_M è il tipo di **controllo dell'errore di tipo I** (***Type I error***) che si vuole mantenere nella ricerca: questo errore viene compiuto ogni volta che **H_M è erroneamente respinta**, ovvero ogni volta che H_A è erroneamente accettata. **Alfa** rappresenta **la soglia di assunzione del rischio di commettere l'errore di accettare H_A quando avremmo dovuto confermare H_M**. Dato che è H_M a essere

verificata, questo è l'errore che si cercherà sempre di minimizzare: **alfa** (α) è la soglia massima di **probabilità di commettere un errore di I tipo a lungo termine**. Convenzionalmente, sono ritenuti adeguati livelli α pari al 5% o all'1% (.05 - cinque errori su 100 decisioni, o .01 - un errore su 100 decisioni), ma possono esserne scelti altri: tra i più frequenti, .001 (un errore su 1000 decisioni).

Scriveremo H_M, quindi, in modo da incorporare anche il livello alfa:

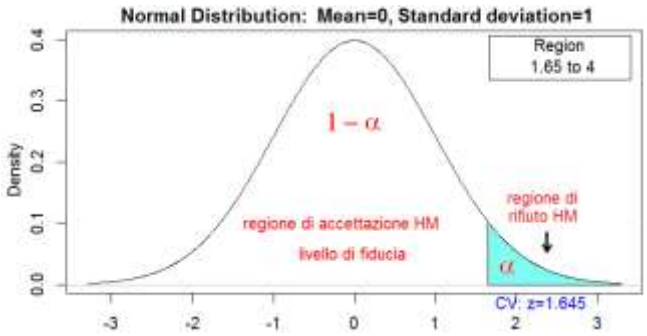
$$H_M: M_1 - M_2 = 0 \pm MES, \alpha = .05$$

Per confondere le cose, alfa è molto simile al livello di significatività del risultato proposto da Fisher, tanto che Pearson e Neyman l'hanno chiamata "livello di significatività del test" e hanno adottato gli stessi valori convenzionali suggeriti da Fisher. Però:

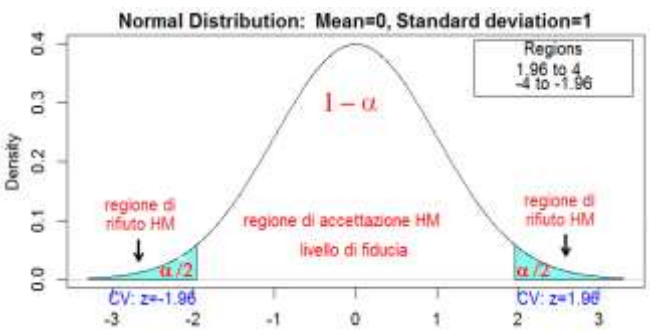
- l'alfa di Pearson-Neyman deve essere stabilita ancor prima di raccogliere il dato - non per caso si definisce **fixed alpha** -, mentre il livello di significatività di Fisher è definibile a risultato ottenuto – e addirittura può essere lasciato al lettore, come abbiamo visto;
- l'approccio FAA non è un test di significatività, perché gli autori non sono interessati alla forza dell'evidenza **contro** H_M, ma un **test di accettazione**: si tratta di decidere se accettare H_A invece di H_M;
- alfa non ammette una gradazione: per uno stesso test si sceglie un livello alfa O un altro, ma non entrambi. In pratica, dire che "il risultato è altamente significativo" perché ha un $p - value = 0.00001$ ha senso nell'approccio di Fisher, che ammette vari livelli di eccezionalità, ma non in quello di Pearson – Neyman.

Questo titolo è uno storico, in tutti i sensi, errore di I tipo☺ Potete leggerne la storia e trarne l'opportuna lezione qui: "How sampling bias may ruin your model", https://medium.com/@ODSC/dewey-defeats-truman-how-sampling-bias-can-ruin-your-model-f4f67989709e

Il livello alfa traccia una **regione critica**, o regione di **rifiuto**, sulla distribuzione di probabilità di H_M: qualsiasi statistica che cada **fuori** da questa regione critica sarà ritenuta come **ragionevolmente probabile sotto** H_M; qualsiasi statistica che **cada entro questa regione** critica sarà ritenuta come **ragionevolmente probabile sotto** H_A.



Ipotesi monodirezionale



Ipotesi bidirezionale

Alfa, quindi, consente di **identificare il valore critico** del test (critical value $\boldsymbol{CV}$, o test criterion T_{crit}), cioè il confine tra le due ipotesi, ancor prima di condurre la ricerca. Una volta noto il valore critico, completiamo H_M inserendolo:

$$H_M: M_1 - M_2 = 0 \pm MES, \alpha = .05, CV_z = |1.96|$$

Di nuovo, la regione critica è molto simile a quella stabilita dal livello di significatività di Fisher come punto di cut off tra eccezionale e non eccezionale. Però:

- mentre Fisher è più interessato al $p - value$ del risultato ottenuto, la regione critica è basata su un valore critico indipendente dal valore del test effettivamente osservato;
- è fissato a priori, quindi immobile;
- non è graduabile, mentre nell'approccio di Fisher si possono delimitare diverse regioni critiche più estreme come aree di maggior evidenza.

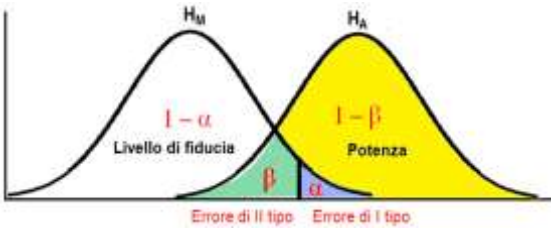
4. **Definizione dell'ipotesi alternativa** H_A. Nonostante sia uno degli aspetti più innovativi di questo approccio, H_A è spesso postulata in maniera aspecifica, anche dagli autori stessi → $H_A: M_1 - M_2 \neq 0$. È però più corretto inserirvi anche il MES:

$$H_A: M_1 - M_2 \neq 0 \pm MES$$

Oltre al già discusso MES, nel definire H_A dobbiamo anche stabilire quale sia **l'errore di II tipo** (**Type II**[37]) che ci si consente di commettere. L'errore di II tipo è commesso quando **erroneamente confermiamo** H_M, ovvero quando **respingiamo scorrettamente** H_A. Poiché fare un errore di II tipo ha conseguenze meno gravi di un errore di I tipo, la probabilità di questo errore viene stabilita **dopo aver scelto alfa**: **beta** (β) è la soglia massima di **probabilità di commettere un errore di II tipo a lungo termine**. Per minimizzarlo, Cohen (1969) propone, per motivazioni di buon senso, che il rischio di commettere a errore di primo tipo sia valutato al massimo quattro volte maggiore di uno di II tipo, per cui: $\boldsymbol{\beta = 4 \times \alpha}$. Quindi, se $\alpha = .05$:

$$H_A: M_1 - M_2 \neq 0 \pm MES, \alpha = .05,\ \beta = .20$$

Naturalmente, è impossibile pensare di **eliminare** gli errori di I e II tipo: si può ridurre al massimo la loro frequenza, però, e, conoscendo la probabilità con cui si verificano, si può prendere la decisione su H_M con la consapevolezza della probabilità di stare facendo un errore. Perciò, se visualizziamo le distribuzioni di probabilità di H_M e H_A una a fianco dell'altra, vedremo:



Le ipotesi alternative possono essere bidirezionali o monodirezionali, coerentemente con l'ipotesi principale:

- **Bidirezionale / a due code**: $H_M: \theta_1 = \theta_2$ *versus* $H_A: \theta_1 \neq \theta_2$
- **Monodirezionale destra / a una coda, destra**: $H_M: \theta_1 \leq \theta_2$ *versus* : $H_A: \theta_1 > \theta_2$
- **Monodirezionale sinistra / a una coda, sinistra**: $H_M: \theta_1 \geq \theta_2$ *versus* : $H_A: \theta_1 < \theta_2$

Se l'ipotesi alternativa è bidirezionale, H_A non sarà confermata **sia** se θ_1 risultasse **non casualmente maggiore** di θ_2, sia nel caso in cui θ_1 risultasse **non casualmente minore** di θ_2. Invece, nel caso dell'ipotesi monodirezionale, H_M non sarà confermata **solo** se la **direzione della differenza tra le medie sarà esattamente quella prevista**: se $H_A: \theta_1 > \theta_2$, dovremo accettare H_M sia se $\theta_1 = \vartheta_2$, sia se $\vartheta_1 < \theta_2$. Le ipotesi monodirezionali possono essere le uniche teoricamente accettabili: per esempio, nel caso dei barattoli di Nutella dovremmo formulare un'ipotesi alternativa monodirezionale, dato che non sarebbe per noi un problema se i barattoli dovessero **pesare più** del peso dichiarato in etichetta.

Nel processo decisionale su H_M e H_A abbiamo quindi quattro possibili esiti (purché la potenza sia buona: vedi sotto):

[37] L'opinione di Fisher sul concetto di errore di II tipo, e in generale sulla proposta di E. Pearson e Neyman, è inequivocabile: "The phrase 'Errors of the second kind,' although apparently only a harmless piece of technical jargon, is useful as indicating the **type of mental confusion in which it was coined**" (Fisher, 1955 p. 73)

5. Si calcola la **numerosità campionaria richiesta per una buona potenza**: la **potenza** è la preoccupazione principale di questo approccio: è la probabilità di respingere correttamente H_M in favore di H_A, ovvero di accettare correttamente H_A, e quindi è il **complemento di beta**: $\mathbf{1 - \boldsymbol{beta}}$. La potenza dipende dal tipo di test e dalla numerosità campionaria (l'abbiamo già visto), come anche dall'entità dell'effect size atteso (intuitivamente, la potenza è funzione **crescente** della differenza riscontrata, in valore assoluto), la variabilità dei dati (la potenza è funzione **decrescente** della varianza), $\boldsymbol{\alpha}$ (un alfa maggiore aumenta la potenza) e, ovviamente, β stesso (un beta minore aumenta la potenza), direzionalità dell'ipotesi (a parità di tutti gli altri fattori, l'ipotesi a una coda è sempre più potente della corrispettiva a due code, perché il valore critico corrispondente, espresso in termini assoluti, è sempre minore). In genere, si ritiene adeguata una potenza pari ad almeno $1 - .20 = .80$. N, α, β ed effect size sono quindi legati tra loro, come approfondiremo nel §6.7.
   La principale differenze tra le procedure FAA e PVA è proprio che H_A dà informazioni esplicite al test su ES e beta: se ignoriamo ES e beta, stiamo usando, più o meno consapevolmente, l'approccio di Fisher.
6. Si calcola il **valore critico del test** (**CV**), che sarà usato come cut off per decidere tra H_M e H_A, in base al test identificato come ottimale, a N (in molti test partecipa al calcolo dei gradi di libertà) e alfa.

A POSTERIORI:

7. Raccolti i dati, si calcola il **valore del test** da loro derivante (**valore ottenuto** o **research value**, **RV**): questo valore è tanto più prossimo a zero quanto più il dato campionario è prossimo alla media prevista da H_M. I $p - value$ possono essere usati per verificare i dati anche in questo approccio, dato che la verifica dei dati sotto condizione di H_M è simile a quella sotto condizione di H_0: si calcola la probabilità teorica del risultato campionario assunta per vera H_M: $P(D|H_M)$. Da questa prospettiva, il $p - value$ è tanto più grande quanto più il dato è prossimo al valore atteso da H_M, all'opposto dei RV.
8. **Si decide a favore di H_M o H_A**: una volta che i passi precedenti sono stati soddisfatti, questa decisione è piuttosto meccanica:
   a. Se i risultati osservati cadono **nella regione critica**, si **respinge** H_M e si accetta H_A
   b. Se i risultati osservati cadono **fuori dalla regione critica e** il test ha **una buona potenza**, si accetta H_M;
   c. Se i risultati osservati cadono **fuori dalla regione critica e** il test ha una **scarsa potenza**, non si **conclude nulla**... non si dovrebbero eseguire ricerche con potenza insufficiente.

L'approccio FAA è più potente del PVA per testare i dati sul lungo periodo, anche se il campionamento ripetuto è una procedura abbastanza rara in ricerca: è quindi particolarmente adatto a ricerche che campionano ripetutamente dalla stessa popolazione, con gli stessi test (per esempio, nei controlli di qualità industriali, o nei test diagnostici su larga scala). L'approccio è deduttivo, piuttosto meccanico una volta prese le decisioni a priori, e di conseguenza meno flessibile di quello di Fisher, a cui è superficialmente simile.

Vediamone un esempio: siccome la replicazione è l'anima della scienza, ricompriamo altri barattoli magnum di Nutella scelti a caso per verificare se il peso dichiarato in etichetta ($\boldsymbol{\theta = 3Kg}$ → peso della popolazione) corrisponde a quello atteso.

1. Siamo interessati a una sensibile differenza tra il valore atteso e quello riscontrato nei barattoli campionari: pochi grammi non sarebbero interpretativamente significativi, anche se dovessero risultare statisticamente significativi. Diciamo allora che ci aspettiamo **almeno 70 grammi di differenza**, un po' più di quattro cucchiai, secondo H_A: questo sarà il **minimum effect size**. Se altri ricercatori, prima di noi, avessero fatto questa stessa ricerca, potremmo basarci sui loro risultati per determinare il MES.
2. Scegliamo il test: torniamo ad utilizzare la **distribuzione di probabilità normale**.
3. Definiamo l'ipotesi principale H_M **e alfa**: la differenza tra la media dei barattoli e il valore atteso è pari a 0 ± il minimum effect size. Scegliamo di controllare il rischio di respingere erroneamente questa H_M a vantaggio di H_A (errore di I tipo) con un margine di errore pari al 5%: $\boldsymbol{\alpha = 0.05}$
4. Come ipotesi alternativa, scegliamo una H_A **monodirezionale**: ci preoccupa che il peso campionario sia **inferiore** a quello previsto, non superiore, quindi avremo un'ipotesi monodirezionale **sinistra**: $\boldsymbol{H_A: \bar{x} < \mu}$. Scegliamo di controllare il rischio di respingere erroneamente questa H_A a vantaggio di H_M (errore di II tipo) accogliendo il suggerimento di Cohen e ponendo la probabilità di commettere un errore di II tipo (**beta**) come quattro volte maggiore rispetto ad alfa: $\boldsymbol{\beta = 0.05 \times 4 = .20}$. Di conseguenza, la **potenza** del nostro test sarà pari a $\boldsymbol{1 - .20 = .80}$.
5. Calcoliamo la numerosità campionaria richiesta, alla luce del test prescelto, di alfa, beta ed effect size. La tecnica delle procedure di **power analysis** (v. anche §6.5) non è molto complessa, ma è fuori programma: gli interessati possono approfondirla, con alcune funzioni di R, nell'Appendice 2. Qui usiamo rapidamente la funzione `power.norm.test` di **`pwr`**, che effettua analisi di potenza quando si confronta un valore campionario con uno atteso e la distribuzione di probabilità attesa è la normale standardizzata: i suoi argomenti sono **d=** $(\theta_A - \theta_M)/\sigma$, ovvero il minimum effect size ponderato per la variabilità attesa (in altre parole, il minimum effect size standardizzato), `sig.level= alfa`, `power= 1-beta`, `alternative = "less/greater/two.sided"`, `n= numerosità`. **Inserendo** solo **quattro di questi argomenti, sarà restituito il quinto**. Recuperando la **s= .18** del precedente esperimento, scriveremo allora:

```
pwr.norm.test(d = (2.93-3)/.18,sig.level = .05,alternative = "less", power = .80)
```

```
     Mean power calculation for normal distribution with known variance
              d = -0.3888889
              n = 40.88058
      sig.level = 0.05
          power = 0.8
    alternative = less
```

Dovremo compare **41 barattoli**, questa volta.

Verificate come cambia la numerosità attesa, a parità di tutti gli altri parametri e per lo stesso test, quando il minimum effect size atteso è pari a 20 grammi e a 100 grammi. Cosa ne deduciamo? E se mantenessimo fissa la numerosità della precedente ricerca ($N = 36$), a parità di tutti gli altri fattori, quale potenza otterremmo? Provate con i tre MES (20, 70 e 100 grammi): sarebbe soddisfacente in tutti i casi? Perché?

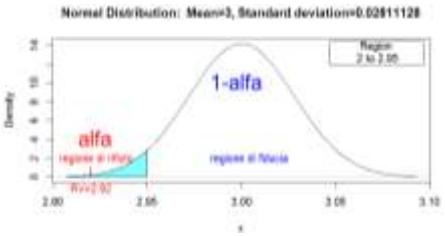
6. Calcoliamo il **valore critico del test** ($\boldsymbol{CV}$): per questa analisi, dovremo calcolare il quantile z di una distribuzione normale con $\mu = 3$ e $\sigma = .18/\sqrt{41}$, corrispondente ad $\alpha = .05$ nella coda **sinistra** della distribuzione, perché $H_A: M_1 < \theta$. Usiamo l'inverso della nota funzione di ripartizione `pnorm`, specificando `lowert.tail= TRUE`:

```
qnorm(p = 0.05, mean = 3, sd = .18/sqrt(41), lower.tail = TRUE)
[1] 2.953761
```

Qualsiasi peso medio della nuova distribuzione di barattoli uguale o inferiore a 2.954 Kg cadrà nella regione di rifiuto di H_M, portandoci ad accettare H_A.

7. Ora procediamo a comprare i 41 barattoli, nel modo più randomizzato possibile, e a pesarli nel modo più accurato possibile, per evitare l'errore sistematico nella misura e garantici il più possibile da quello casuale: il peso medio di questa seconda ricerca è ancora $\boldsymbol{RV = 2.92\ Kg}$,

8. Il valore ottenuto **cade nella regione di rifiuto di** H_M. Dovremmo concludere che il peso medio dei barattoli non è una fluttuazione casuale della media della popolazione, ovvero che il campione non è rappresentativo della popolazione.



Errori di III e di IV tipo

A seguito della definizione degli errori di I e II tipo, diversi autori hanno proposto, più o meno provocatoriamente, l'esistenza di altri "errori" tipici nella verifica di ipotesi.

Sono diversi i cosiddetti errori di III tipo: Mosteller (1948) lo definisce come l'errore di respingere correttamente l'ipotesi nulla, ma per la ragione sbagliata; Kaiser (1966) lo attribuisce a un errore di direzione quando si accetta un'ipotesi alternativa bidirezionale: si afferma che la differenza favorisce una condizione sperimentale, ma in popolazione la differenza nei parametri va nella direzione opposta. Secondo Kimball (1975) è l'errore di dare la risposta corretta al problema sbagliato, che si verifica quando lo statistico dà una risposta corretta a una domanda diversa da quella posta dal committente della ricerca. Per ridurne la probabilità di comparsa, può essere una buona idea usare test **a due vie**, invece di test a una via, nonostante questi siano più potenti.

Marascuilo e Levin (1970) aggiungono anche un errore di IV tipo, da loro riscontrato in molte ricerche: la scorretta interpretazione di un'ipotesi correttamente respinta, analoga alla corretta diagnosi di un medico seguita dalla prescrizione di una medicina sbagliata. Può dipendere da difetti del modello statistico (per esempio la multicollinearità tra i predittori, che troveremo nella regressione multipla, capitolo 11), o dall'esecuzione di un test inadatto ai dati (per esempio, applicare un test parametrico quando non sono rispettati i suoi prerequisiti: ne parleremo diffusamente).

6.4.2 Null Hypothesis Significance Testing

Il Null Hypothesis Significance Testing (NHST) è attualmente la procedura più comune di verifica delle ipotesi, nonostante le pesanti critiche che vedremo in dettaglio nel paragrafo successivo. È in effetti un miscuglio di PVA e FAA, ma non è chiaramente definito, e, a seconda di chi lo propone, può pendere di più verso l'uno o l'altro (APA, 2010; Kline, 2004): perlopiù, segue proceduralmente il FAA e filosoficamente il PVA, nonostante i due approcci siano incompatibili in più punti. (Hubbard, 2004). FAA e PVA lavorano con gli stessi strumenti statistici e producono gi stessi risultati (le differenze sono soprattutto relative alla filosofia della ricerca e a come interpretare i risultati), quindi, per estensione, NHST fa altrettanto nella pratica.

Nello specchietto seguente (modificato da Perezgonzales, 2015) sono evidenziate le affinità o differenze tra FAA e PVA, e le caratteristiche del NHST. In questo approccio, lo scopo del test di verifica è quello di attribuire una probabilità al dato sotto condizione di ipotesi nulla (anche se il suo risultato è spesso equivocato come attribuzione di una probabilità

all'ipotesi nulla alla luce di quanto verificato: vedi oltre); sono prese in considerazione contemporaneamente, in competizione tra loro, due ipotesi: **l'ipotesi nulla** (perlopiù indicata come H_0) e **l'ipotesi alternativa** (perlopiù indicata come H_1). L'ipotesi alternativa è concepita come "non H_0", è a questa subordinata (non è l'oggetto diretto della verifica), e, a differenza del FAA, solo saltuariamente si vede associati a priori l'effect size atteso e beta. La statistica d'interesse è il $p - value$, che viene confrontato con un cut off identificato dalla soglia alfa decisa a priori; questo punto è tuttavia particolarmente confuso, dato che in molti testi si interpreta il $p - value$ **graduandolo** ("molto significativo") come nel PVA. Viene data priorità all'errore di I tipo, ma è consuetudine considerare anche quello di II tipo: ne consegue che la valutazione della potenza è abbastanza frequente, e il calcolo della numerosità campionaria a priori, necessaria a sostenere la potenza del test, è spesso richiesto. Se il risultato del test cade al di fuori della regione critica, si conferma H_0; se cade entro la ragione critica, H_0 viene considerata disconfermata e si accetta H_1.

	Obiettivo del test	**Approccio**	**Scopo della ricerca**	**Ipotesi testata**	**Ipotesi alternativa**
Fisher	Dati: $P(D\|H0)$	A posteriori	Significatività statistica dei risultati	H_0, da falsificare con l'evidenza	Non necessaria (implicitamente, $\neg H_0$)
↓ ↑	=	≠	≠	≈	≠
Pearson-Neyman	Dati: $P(D\|HM)$	A priori (perlopiù)	Significatività statistica, ma anche usato per decidere tra ipotesi	H_M, da favorire rispetto ad H_A	Necessaria; fornisce ES e β
NHST	Dati: $P(D\|H0)$ – ma spesso equivocato come: $P(H0\|D)$	A posteriori, talvolta entrambi	Decidere tra ipotesi in competizione	Entrambe: $H_0 = H_M$; $H_1 = H_A$	H_A posta come $\neg H_0$; ES e β talvolta considerati

	Distribuzione di probabilità del test	**Cut off**	**Calcolo della numerosità campionaria**	**Statistica d'interesse**	**Errore**
Fisher	Appropriata per H_0	La significatività identifica i risultati notevoli; può essere graduata e corretta a posteriori	No	$p - value$, come evidenza contro H_0	Possibile, ma irrilevante entro i singoli studi
↓ ↑	=	≠	≠	≠	≠
Pearson-Neyman	Appropriata per H_M	Comune a CV, α, β e MES; non graduata, non corretta a posteriori	Basato su test, ES, α e potenza	CV (il $p - value$ può essere usato come sostituto)	α= tipo I, β= tipo II
NHST	Appropriata per H_0	Significatività = α; graduata, corretta a posteriori	Facoltativo (ma consigliato)	$p - value$, come evidenza contro H_0 e sostituto per accettare H_A	$p - value = \alpha$ = errore di tipo I (talvolta è considerato anche β)

	Ipotesi alternativa	**Risultato esterno alla regione critica**	**Risultato interno alla regione critica**	**Interpretazione risultato in regione critica**	**Passi successivi**
Fisher	Non necessaria (implicitamente, $\neg H_0$)	Si ignora il risultato, non significativo	Si respinge H_0	O è un evento eccezionale, o H_0 è inadeguata	Necessarie repliche del risultato, utili le meta-analisi
↓ ↑	≠	≠	≠	≠	≠
Pearson-Neyman	Necessaria; fornisce ES e β	Si accetta H_M, se la potenza è buona, altrimenti nessuna conclusione	Si accetta H_A = si respinge H_M in favore di H_A	H_M spiega il risultato meglio di H_A, per un dato α	Impossibile sapere se è stato commesso un errore. necessari ricampionamenti dalla stessa popolazione
NHST	H_1 posta come $\neg H0$; ES e β talvolta considerati	Si ignora il risultato come non significativo, o si accetta H_0, o non si conclude nulla	Entrambi	H_0 è stata disconfermata	Ulteriori studi possono essere raccomandati soprattutto se i risultati non sono significativi

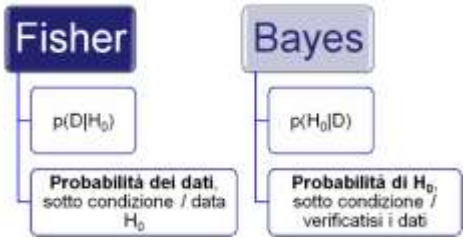
Riassumendo, il percorso logico di un'inferenza statistica secondo NHST è:

- Sottoponiamo a falsificazione H_0, che si riferisce sempre a un dato **parametro della popolazione** (media, proporzione, varianza, ecc.) e si esprime solo con il segno **=** (per esempio, $H_0: \mu = 4$);

- formuliamo H_1, che si esprime con il segno ≠, oppure > o < (per esempio, H_0: $\mu \neq 4$), ma ricordiamoci che **non** è questa che viene effettivamente sottoposta a verifica;
- H_0 e H_1 non hanno pari rango: solo se l'evidenza empirica contro H_0 è realmente forte, essa viene rifiutata (meglio dire, forse, "accantonata" fino a prova contraria);
- La conclusione cui si giunge al termine del percorso di verifica, che si attua adattando un **modello statistico ai dati**, consente di accettare o rifiutare H_0 se il test è rispettivamente non significativo (la probabilità dei datti sotto condizione di H_0 è sufficientemente alta) o significativo (la probabilità dei dati sotto condizione di ipotesi nulla è sufficientemente bassa).
- Ricordiamo che un **risultato non significativo non attesta che H_0 è vera**, ma solo che i dati non portano sufficiente evidenza del suo contrario;
- Se impieghiamo ripetutamente il procedimento di verifica nel caso in cui H_0 sia vera, sarà possibile ottenere, anche se molto raramente, **casi estremi che ci portano a rifiutare H_0,** sbagliando.

Attenzione: una affermazione come: "**una misura statistica** [ad esempio, una differenza tra medie] **è significativa per $p < .05$"**, che si legge correntemente nei report di ricerca, vuol significare che la **probabilità dei dati disponibili, dato che l'ipotesi nulla sia vera, è inferiore al 5%**. **Non è vero, quindi, che stabilire il livello di significatività vuol dire valutare la probabilità dell'ipotesi nulla**, che è un errore comune, ma da evitare accuratamente.

È **l'approccio di Bayes**[38], assai usato in altre discipline, ma ancora poco in quelle umanistiche, che permette effettivamente di **stimare la probabilità dell'ipotesi nulla**, che nell'approccio NHST è data, come assunto, pari a 1 (evento certo).



[38]La formula di Bayes sarebbe in realtà: $p(H|D) = \frac{(D|H)\times p(H)}{p(D|H)\times p(H)+p(D|non\ H)\times p(non\ H)}$

Prima di proseguire: distribuzioni di probabilità centrali e non centrali

Abbiamo introdotto H_0 e H_1, e stiamo per introdurci decisamente, senza guardarci indietro, nella verifica delle ipotesi, in cui useremo distribuzioni di probabilità, diverse a seconda della statistica usata dal test che applicheremo ai dati: possiamo complicare un po' le distribuzioni di probabilità che abbiamo affrontato nel capitolo 5, distinguendo tra **distribuzioni di probabilità centrali e non centrali**. In effetti, quando si parla di **distribuzione di probabilità senza ulteriori specificazioni** (distribuzione t, F, χ^2, eccetera) , intendiamo distribuzione di probabilità **centrale**.

La distribuzione di probabilità centrale rappresenta il modo in cui è distribuita la statistica di un test sotto condizione di ipotesi nulla; la distribuzione di probabilità **non centrale** rappresenta come si distribuisce la statistica di un test quando H_1 è vera in popolazione.

Per esempio, vedremo che la distribuzione (centrale) t di Student è usata per stabilire quanto è verosimile una data differenza tra due medie campionarie, se le medie attese delle popolazioni da cui provengono i gruppi sono le stesse: quanto è probabile che la differenza nel punteggio di ansia tra un gruppo di donne ($\bar{x}_D = 40.5$) e un gruppo di uomini ($\bar{x}_U = 38.5$) risulti $d = 2$, se uomini e donne provengono dalla medesima popolazione ($\Delta = 0$)? Con la distribuzione di probabilità non centrale, invece, ci chiediamo quanto è verosimile una data differenza tra due medie campionarie, se le medie delle popolazioni da cui provengono i gruppi sono diverse: quanto è probabile che la differenza nel punteggio di ansia tra un gruppo di donne ($\bar{x}_D = 40.5$) e un gruppo di uomini ($x_U = 38.5$) risulti $d = 2$, se uomini e donne provenissero da una popolazione di donne con media attesa $\mu_D = 45$ e una popolazione di uomini con $\mu_U = 35$ ($\Delta = 10$)?

La **forma** della distribuzione di probabilità **centrale varia solo in funzione dei gradi di libertà** del test (per la distribuzione t applicata a due gruppi indipendenti: $N - 2$), mentre la **forma** della distribuzione **non centrale anche è una funzione del parametro di non centralità** (**NCP**), a sua volta derivato dalla numerosità campionaria N e dall'effect size atteso; per la distribuzione t: $ncp_t = \sqrt{\frac{N}{2}} \times effect\ size$. Intuitivamente, infatti, se vogliamo sapere quanto è verosimile una data differenza tra medie osservate, se le medie delle popolazioni differiscono per una certa quantità Δ, dobbiamo considerare l'entità di questa differenza e la grandezza del campione.

Il parametro di non centralità è l'argomento opzionale **ncp**= `valore` delle funzioni `pt`, `pf`, `pchisq`: di default è `ncp=0`, esprimendo che la differenza attesa tra le medie in popolazione è pari a zero – cioè esprimendo una distribuzione di probabilità centrale, sotto condizione di ipotesi nulla. Dato che lavoreremo con H_0 - e così fanno le funzioni che calcoleranno per noi le statistiche dei test e il loro $p - value$, potremo ignorare questo argomento. Ma ricordiamo che le domande cui rispondono `ncp=0` e `ncp≠0` sono sottilmente diverse.

*Quanto è verosimile trovare una **differenza** tra due gruppi ($n_1 = 19$, $n_2 = 18$: $N = 37$) **pari o superiore a** $d = 13$, se i due gruppi **provengono da due popolazione con la stessa media**, e quindi in cui Δ=0?*

```
pt(q = 13,df = 35,lower.tail = FALSE)
[1] 2.930281e-15
```

```
pt(q = 13,df = 35,lower.tail = FALSE, ncp=0)
[1] 2.930281e-15
```

È moooolto poco verosimile: non confermiamo H_0, i due gruppi appartengono a popolazioni con medie differenti

*Quanto è verosimile trovare una differenza tra due gruppi ($n_1 = 19$, $n_2 = 18$: $N = 37$) pari o superiore a $d = 13$, se i due gruppi provengono da **due popolazioni con medie diverse**, tali per cui il parametro di non centralità, alla luce di N, è:*

... uguale a ncp= 5

```
pt(q = 13,df = 35,lower.tail = FALSE, ncp=5)
[1] 2.717007e-06
```

È moooolto poco verosimile: non confermiamo H_0, i due gruppi appartengono a popolazioni con medie differenti

... uguale a ncp= 12

```
pt(q = 13,df = 35,lower.tail = FALSE, ncp=12)
[1] 0.3146047
```

... uguale a ncp= 20

```
pt(q = 13,df = 35,lower.tail = FALSE, ncp=20)
[1] 0.9998951
```

È sufficientemente verosimile: confermiamo H_0, i due gruppi appartengono a popolazioni con uguale media / a una medesima popolazione.

6.5 Critiche – non troppo velate - all'approccio NHST

"...Sir Ronald [Fisher] has befuddled us, mesmerized us, and led us down the primrose path. I believe that the almost universal reliance on merely refuting the null hypothesis as the standard method for corroborating substantive theories […] ***is a terrible mistake****, is* ***basically unsound****,* ***poor scientific strategy****, and* ***one of the worst things*** *that ever happened in the history of psychology"*
Meehl, 1978, pag.817

L'opinione di Meehl, un eccellente statistico, riportata nella citazione a inizio paragrafo, è riferita *sensu strictu* al PVA, ma si estende al NHST da esso derivato: può sembrare impietosa, ma è tutt'altro che isolata[39] e, a ben vedere, è necessaria; in effetti, una delle criticità maggiori del NHST è anche un probabile motivo della sua attuale diffusione, ovvero il fatto che spinga a vedere il mondo in bianco e nero, a formulare le ipotesi e a trarre le conclusioni in termini dicotomici (un effetto esiste o non esiste, una relazione è significativa o non significativa), quando la realtà è ben diversa (Kirk, 2003).
Proviamo a separare le critiche per punti:

1. Problemi logici derivanti dalla natura probabilistica dell'approccio NHST: tra (molti) altri, Cohen[40], un fondamentale statistico che abbiamo già intravisto nel §6.4 e di cui riparleremo a breve per ringraziarlo di aver diffuso nella pratica il concetto di **effect size**, critica le basi logiche del ragionamento di Fisher, che, come ricorderete, si basa sul sillogismo. Il *modus tollens* (che nega le premesse negando le conseguenze) nella logica formale è un ragionamento perfettamente valido → da due premesse discende **una** sola conclusione:

Se H_0 è vera, allora questo dato non può verificarsi.	$A \rightarrow \neg B$
Tuttavia, questo dato si è verificato.	B
Quindi, H_0 è falsa	$\neg A$

Però, il ragionamento alla base di NHST è probabilistico, non deterministico, e nel ragionamento probabilistico il *modus tollens* perde efficacia:

Se l'ipotesi nulla è corretta, allora questo dato è **altamente improbabile**	***Se uomo suona la chitarra, probabilmente non fa parte degli AC/DC***
Tuttavia, questo dato si è verificato.	***Tuttavia, Angus Young fa parte degli AC/DC***
Quindi, l'ipotesi nulla è altamente improbabile.	***Quindi, probabilmente Angus Young non suona la chitarra***[41]

Questa fallacia è stata definita "*the illusion of attaining improbability*" (Falk e Greenbaum, 1995) o "*the odds-against-chance fantasy*" (Carver, 1978).

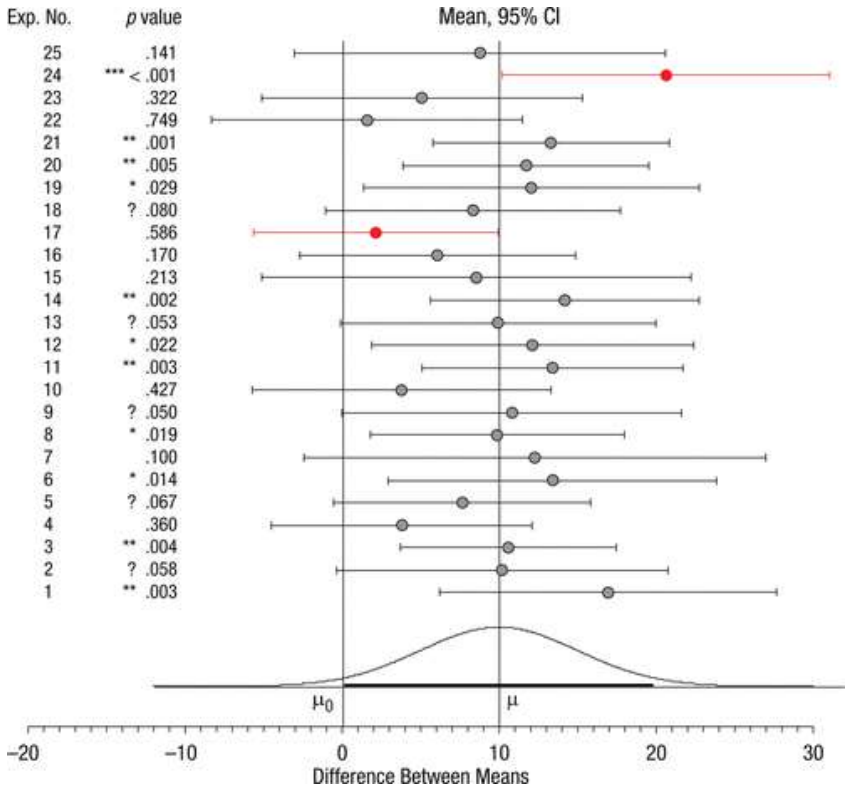
[39] **Alcuni** esempi di voci critiche riportati da Balluerka, Gómez & Hidalgo (2005): anni '60: Bakan, 1966; Cohen, 1962; Grant, 1962; Lykken, 1968; Meehl, 1967; Rozeboom, 1960. Anni '70: Carver, 1978; Cronbach, 1975; Greenwald, 1975; Meehl, 1978; Morrison & Henkel, 1970; Tversky & Kahneman, 1971; Anni '80: Brewer, 1985; Cohen, 1988; Dar, 1987; Falk, 1986; Gigerenzer & Murray, 1987; Gigerenzer et al., 1989; Guttman, 1985; Huberty, 1987; Kupfersmid, 1988; Oakes, 1986; Rosnow & Rosenthal, 1989; Sedlmeier & Gigerenzer, 1989; anni '90: Carver, 1993; Cohen, 1990, 1994; Dar, Serlin, & Omer, 1994; Falk & Greenbaum, 1995; Finch, Cumming, & Thomason, 2001; Gigerenzer, 1993; Harris, 1991; Hubbard, 1995; Hunter, 1997; Hunter & Schmidt, 1990; Kirk, 1996, 2001; Loftus, 1991, 1995, 1996; Meehl, 1990a, 1990b; Rossi, 1990, 1997; Shaver, 1993; Schmidt, 1992, 1996; Thompson, 1993, 1994, 1996, 1997; Tukey, 1991... eccetera.
[40] Leggete "**The earth is round (p <.05)**" di Cohen, disponibile su Elly, e quasi tutto quello che vorrete sapere sull'inferenza statistica vi sarà spiegato, in maniera realmente spassosa.
[41] Non è necessario specificare che **Angus Young suona effettivamente la chitarra**, vero? Vero?!

2.L'approccio NHST non offre le informazioni che il ricercatore vuole ottenere. È una delle critiche più forti: diversi autori[42] hanno affermato che NHST e inferenza statistica hanno obiettivi diversi. L'obiettivo dell'inferenza statistica è conoscere la probabilità che H_0 sia vera alla luce dei risultati ottenuti nel campione, cioè $P(H0|D)$. Invece, l'approccio NHST ci dice solo quale sia la probabilità di ottenere dati che sono ugualmente o più discrepanti da quelli effettivamente ottenuti, assunto che H_0 sia vera, cioè $P(D|H0)$. Lindley ha dimostrato (1957) che in alcune condizioni la $p(H0|D)$ può avvicinarsi a 1, mentre la $p(D|H0)$ si approssima 0 (**paradosso di Lindley**): il $p-value$, dunque, non riflette la probabilità che H_0 sia scorretta. Come abbiamo già notato, la stima di $p(H0|D)$ può essere ottenuta solo tramite tecniche di tipo bayesiano.

3. L'approccio NHST non permette di verificare le teorie psicologiche: un'erronea concettualizzazione, affine alla precedente, è che l'NHST possa essere usato per determinare la probabilità che un'ipotesi di ricerca sia corretta, e, di conseguenza, che la teoria alle sue spalle ne venga confermata. In realtà, anche quando una H_0 è respinta dai dati, è sempre necessario escludere un'altra serie d'ipotesi concorrenti, prima di verificare la validità dell'ipotesi di ricerca. L'aumentata veridicità di questa ipotesi può essere supportata solo da una solida base teorica, da un disegno di ricerca (sperimentale) appropriato e da ripetute **repliche** dello studio in differenti condizioni. Inoltre, molti autori[43] affermano che l'approccio NHST non riesce neppure a garantire informazioni sull'importanza pratica dei risultati e sulla grandezza degli effetti osservati. Come ha puntualizzato Tukey (1969, 1991), l'avanzamento della conoscenza richiede informazioni tanto sulla direzione della differenza quanto sulla sua grandezza, e NHST dice qualcosa solo sulla sua direzione.

4. La fallacia della replicabilità: diversi autori[44] affermano che un'altra cattiva interpretazione, collegata all'illusione di raggiungere l'improbabilità, è che il valore complementare di p, cioè $1-p$, esprima la probabilità che i risultati siano replicabili. Se il $p-value$ calcolato ci dicesse qual è la probabilità che H_0 sia vera in popolazione, allora costituirebbe un indice della replicabilità dei risultati, ma, come abbiamo già detto, il $p-value$ non fa nulla di tutto ciò. In effetti, quella che Gigerenzer (1993) chiama "*the fallacy of the replication*" confonde il livello di significatività con la potenza statistica. Se il $p-value$ rivelasse la verità, e si replicasse l'esperimento con condizioni identiche tranne un diverso campione casuale, il $p-value$ della replica dovrebbe confermare la stessa verità: in realtà, in studi basati su **simulazioni** di repliche di esperimenti si trova che il $p-value$ cambia drammaticamente da simulazione a simulazione, tanto da rendere impossibile predire se la replica successiva avrà un $p-value$ molto, poco o per nulla inferiore alla soglia alfa. Cumming (2008; 2014) ha efficacemente definito il fenomeno "***$p-value$* dance**": investite 10 minuti di tempo per vedere il video: https://www.youtube.com/watch?v=5OL1RqHrZQ8, non li rimpiangerete (anche: https://www.youtube.com/watch?v=OcJImS16jR4, ma questo non è sottotitolato). Per ragionare sulla stabilità del risultato sperimentale, l'autore propone di interpretare i CI ignorando completamente i $p-value$. La figura sottostante (Cumming, 2008) illustra la "danza" dei $p-value$ e dei CI, fornendo affiancati i CI delle medie e i $p-value$ riferiti a repliche di un confronto tra le medie di due campioni, di uguale numerosità, con un effect size moderato ($d = .50$) e una potenza altrettanto moderata ($1-\beta = .50$).

[42]per esempio, Berger & Sellke, 1987; Carver, 1978; Cohen, 1990, 1994; Cronbach & Snow, 1977; Falk & Greenbaum, 1995; Gigerenzer & Murray, 1987; Kirk, 1996; Oakes, 1986; Rozeboom, 1960.
[43]Bakan, 1996; Bracey, 1991; Cohen, 1994; Meehl, 1967; Nickerson, 2000; Rosenthal, 1983, 1993; Rosnow & Rosenthal, 1989; Shaver, 1985, 1993; Thompson, 1996; Thompson & Snyder, 1998; Wilson, Miller, & Lower, 1967.
[44]Bakan, 1996; Carver, 1978, 1993; Cohen, 1994; Falk & Greenbaum, 1995; Gigerenzer, 1993; Gigerenzer & Murray, 1987; Lykken, 1968; Oakes, 1986; Shaver, 1993; Thompson, 1996.



Anche I CI "danzano" avanti e indietro, ma molto più macroscopica è la grandissima varianza dei $p - value$, che comprende praticamente qualsiasi valore intermedio tra 0 e 1. Notate che, comunque, questo costituisce un problema soprattutto per l'approccio di Fisher, che ammette diverse gradazioni di eccezionalità e quindi si confronta con risultati altamente incoerenti, mentre per quello di Pearson e Neyman non è così grave, finché i $p - value$ portano alla stessa decisione su H_0, dato che la loro soglia di riferimento è fissa è prestabilita.

5. **Rigidità della decisione**. Altre critiche riguardano più il metodo di applicazione dell'approccio che la sua logica. Una delle obiezioni più pertinenti è la rigidità della decisione: quando i ricercatori adottano un livello di significatività fisso, convertono un continuum di incertezza, che varia in un range di probabilità da 0 a 1, in una decisione dicotomica su respingere / non respingere H_0. Quindi, un'intera ipotesi può essere disconosciuta anche se il $p - value$ del risultato è di poco superiore alla **convenzionale** soglia $\alpha = .05$: come dicono Rosnow e Rosenthal: "***surely, God loves the .06 nearly as much as the .05 level of significance***" (1989, p. 1277) e, infatti, ricordiamo che Fisher ha solo **suggerito** come conveniente un criterio di eccezionalità pari al 5%. Le motivazioni alla base di questo suggerimento non sono affatto basate su ragioni matematiche, ma sono decisamente arbitrarie, e probabilmente editoriali. All'inizio del Ventesimo secolo, il ricercatore che voleva associare una probabilità a una statistica (χ^2, r, z), poteva consultare il testo di Pearson ("*Tables for staticians and biometricians"*, 1914), che riportava lunghe e dettagliate tabelle con i valori di probabilità. Il testo di Fisher ("*Statistical methods for research workers*"), uscito nel 1925 e ripubblicato più volte fino al 1970, ha sostituito nella pratica quello di Pearson, aggiornandolo con i valori critici dei test successivamente elaborati, ma selezionando sempre più strettamente le tavole di probabilità: questo era dovuto da un lato al bisogno dell'editore di mantenere una foliazione simile tra le edizioni, ma dall'altro al fatto che Fisher aveva bisogno del permesso di Pearson per riprodurre le tabelle... e da tempo il conflitto tra i due si era inasprito ("*Kendall mentioned that Fisher produced the tables of significance levels to save space and to avoid copyright problems with Karl Pearson, whom he disliked"*, Good, 1971[45]).

[45] Cit. in di V. P. Godambe & Sprott (eds.), Foundations of Statistical Inference, pag. 513.

D'altronde, Fisher stesso è stato esplicito sull'insensatezza di un uso rigido di un livello di significatività prefissato (come d'altronde è stato esplicito nei commenti negativi sull'ipotesi alternativa): "*no scientific worker has a fixed level of significance at which, from year in year and in all circumstances, he*[46] *rejects hyptheses: he rather gives his mind to each particular case in the light of the evidence and his ideas"* (1956).

Dall'altra parte della barricata, un buon numero di pubblicazioni altrettanto pregevoli ha difeso la validità e l'utilità dell'approccio NHST[47]: molte strategie, proposte come completamente sostitutive all'approccio NHST dalle voci più radicali, sono state suggerite come complementari alla verifica della significatività. Vediamo (e useremo!) quelle il cui uso è stato proposto dalla Task Force on Statistical Inference dell'American Psychological Society (APA; Wilkinson and TFSI, 1999).

6.6 Strategie complementari o alternative all'approccio NHST

6.6.1 Calcolare e interpretare gli intervalli di fiducia

Sono molti gli autori[48] che ritengono che calcolare i CI attorno alle stime campionarie, come abbiamo fatto nel §6.3, costituisca un'eccellente integrazione, se non un sostituto, della ricerca del $p - value$; è stato anche sperimentalmente dimostrato che interpretare **solo** i CI, ignorando il $p - value$, porta a fare meno errori sulla corretta decisione da prendere, rispetto alla valutazione combinata di CI e $p - value$ (Coulson, Healey, Fidler e Cummings, 2010; Fidler e Lotus, 2009). In effetti:

quando il CI (almeno al 95%) contiene il valore previsto da H_0, qualunque esso sia, accettiamo H_0: il campione appartiene alla popolazione ($\bar{x} - \mu = 0$); due campioni appartengono alla stessa popolazione ($\mu_1 - \mu_2 = 0$); due distribuzioni sono linearmente indipendenti ($\rho_{x_1 x_2} = 0$; capitoli 8 e 9), e così via.

Anche se capita spesso, come negli esempi qui sopra, il valore previsto da H_0 **non è sempre 0**: attenzione, quindi, **non memorizzate** l'informazione "il valore previsto da H_0 è 0", perché **non è vera**. Per esempio, nel caso delle proporzioni H_0 prevede che la proporzione con cui si verifica l'evento atteso p non è significativamente diversa dalla proporzione con cui si verifica l'evento non atteso q, quindi $H_0: \frac{p}{q} = 1$.

Inoltre, abbiamo detto che i CI danno **anche informazioni sulla precisione della stima** dei parametri in popolazione: ampi CI riflettono stime meno accurate. I CI relativi alla differenza tra parametri (ad esempio tra medie di due campioni li vedremo nel t-test per campioni indipendenti e nell'analisi della varianza), oltre a contenere / non contenere il valore previsto da H_0 (per esempio, $\mu_1 - \mu_2 = 0$) , indicano anche la **direzione e la grandezza della differenza** tra parametri. La stima puntuale nel campione e la stima intervallare nella popolazione usano la stessa unità di misura, rendendo **facile l'interpretazione** del risultato. Insomma, i CI evitano molti dei problemi intrinseci dei classici test di significatività, non richiedono che le ipotesi siano formulate a priori, danno molte informazioni e sono più facili da interpretare. Con tanti pregi, sembra difficilmente spiegabile il fatto che i CI siano riportati piuttosto raramente negli articoli di molte

[46] Notate che ai tempi di Fisher il *politically correct* in fatto di genere non era ancora di uso corrente: un ricercatore poteva essere, implicitamente, solo maschio. Oggi scriveremmo: "he/she rejects hypothesis"...
[47] Per esempio, Abelson, 1995, 1997; Chow, 1987, 1988, 1989, 1991, 1996, 1998a,b; Cortina & Dunlap, 1997; Cox, 1977; Dixon, 1998; Frick, 1996; Hagen, 1997).
[48] Per esempio: Allison, Brown, George e Kaiser, 2016; 1966; Cohen, 1990, 1994; Kline, 2004; Goodman, 2008; Kirk, 1996, 2001; Loftus, 1991, 1995, 1996; Loftus&Masson, 1994; Trafimow e Marks, 2015; Meehl, 1997; Nuzzo, 2014; Schmidt & Hunter, 1997; Steiger & Fouladi, 1997

branche delle discipline psicologiche: alcuni (per esempio Cohen, 1994) affermano malignamente che la rarità dei CI negli articoli sarebbe dovuta proprio alla loro imbarazzante ampiezza[49].

Nonostante questa sia al momento la prospettiva ampiamente prevalente in letteratura, segnaliamo per amore di completezza che una parte di statistici (perlopiù di approccio bayesiano), al momento minoritaria, contesta l'uso a fini inferenziali dei CI, che sarebbe **basato su una triplice fallacia** (Morey**,** Hoekstra, Rouder, Lee e Wagenmakers, 2016), basata su un ragionamento a posteriori (dal dato campionario al parametro in popolazione):

1. ***Fundamental confidence fallacy***: l'affermazione "Se la probabilità che un intervallo casuale contiene il vero valore è pari al $X\%$, allora la plausibilità o probabilità che un particolare intervallo osservato contenga il vero valore è anch'essa pari al $X\%$", anche se sembra plausibile e suggerita dal termine "*confidence*" (semanticamente legato ai concetti di plausibilità e credibilità), non regge. La confusione fondamentale è tra quello che si sa **prima** di osservare i dati, cioè che il CI, qualunque sarà, avrà una probabilità prefissata di contenere θ, con quello che si sa **dopo** aver osservato i dati. La teoria del CI non dice nulla sulla probabilità che un dato CI campionario contenga θ: questa può essere solo = 0 (il CI non contiene θ) o 1 (il CI contiene θ). Anche un tenace sostenitore dell'uso dei CI come Cumming mette in guardia contro gli abusi interpretativi: "è essenziale essere estremamente attenti quando si menziona la probabilità in relazione a un CI. È corretto affermare che la probabilità che la media attesa μ sia compresa tra $\bar{x} - w$ e $\bar{x} + w$ è pari al 95%, ma questa è un'affermazione sulla probabilità dei limiti inferiori e superiori, che varia da campione a campione. Sarebbe scorretto affermare che il CI ha una probabilità pari al 95% di includere μ, perché questo suggerisce che μ possa variare, quando invece μ è fissa, anche se sconosciuta" (Finch e Cumming, 2005, pag. 171). Secondo Mayo (1981), l'incomprensione sembra radicata nel desiderio che i CI forniscano "qualcosa che non possono legittimamente dare: una misura del grado di probabilità, credenza o verosimiglianza che il valore di un parametro sconosciuto stia in uno specifico intervallo".
2. ***Precision fallacy***: l'affermazione: "la grandezza di un CI indica la precisione della nostra conoscenza sul parametro: piccoli CI forniscono una conoscenza precisa, mentre larghi CI danno una conoscenza imperfetta sul fenomeno in popolazione" è quantomeno parziale. Secondo Morey et al., non ci sarebbe una connessione logicamente necessaria tra la precisione di una stima e l'ampiezza del CI, il che rende l'affermazione vera solo in alcuni [molti] casi, ma non in tutti.
3. ***Likelihood fallacy***, o fallacia della verosimiglianza: è relativa alla credenza che un CI racchiuda i valori probabili per il parametro, per cui i valori entro il CI sono più probabili di quelli esterni. Anche in questo caso, quest'affermazione non sarebbe vera in tutti i casi, e alcune simulazioni hanno dimostrato che anche "buoni" CI possono escludere quasi tutti i valori ragionevoli o essere vuoti o infinitamente piccoli, escludendo tutti i possibili valori (ad esempio, Steiger, 2004).

Morey et al. ricordano che i CI basati sulla stima campionaria non sono l'unica stima intervallare possibile, e altre stime, come i ***credible intervals*** dell'approccio bayesiano non soffrono dei limiti rimproverati ai CI e danno, in piena legittimità, esattamente le informazioni erroneamente cercate nelle tre fallacie.

[49] "[…] Yet they [CI] are rarely to be found in the literature. I suspect that the main reason they are not reported is that they are so embarrassingly large!"

Non preoccupatevi di dover giustificare l'uno o l'altro approccio: limitiamoci a seguire il *mainstream* attuale nelle scienze psicologiche... ma sappiate che quando i vostri figli vi mostreranno il loro testo di Statistica, le cose potrebbero essere diverse da come le avete imparate ☺

6.6.2 Indici di intensità dell'effetto: Effect sizes

La Task Force on Statistical Inference sollecita i ricercatori a presentare sempre indicatori di grandezza dell'effetto trovato, e a interpretarli nel contesto della pratica e della teoria. Cohen (1988) definisce l'**effect size** (ES) come il **grado in cui il fenomeno è presente nella popolazione**, o, nel campo della verifica della significatività, come **il grado in cui H_0 è falsa**. In un'altra accezione, Snyder e Lawson (993) definiscono l'ES come il **grado in cui la variabile dipendente è controllata, predetta o spiegata dalla variabile** indipendente / dalle variabili indipendenti (lo vedremo nella regressione lineare, nella regressione logistica e nell'analisi della varianza). Oltre a fornire queste informazioni, gli indici di ES **permettono di confrontare direttamente i risultati ottenuti da diversi studi, anche se usano analisi diverse**, dato che questi indici sono trasferibili su una scala comune. Sono essenziali per fare analisi di potenza (**power analysis)** e per gli studi di **meta-analisi**.

Una definizione più pignola distingue:

- **indici di ES** → coefficienti che quantificano la differenza standardizzata tra medie dei gruppi, che secondo H_0 è $= 0$ (ad esempio, le medie dei risultati a un test di psicopatia di un gruppo clinico e di un gruppo di controllo): tra i più usati, d ed f di Cohen, g di Hedges, g di Glass. Sono comunque numerosi anche gli indici di ES che non si riferiscono a medie, ma a indicatori robusti (mediane, medie trimmed...): ne vedremo esempi nell'ANOVA robusta;
- **misure di associazione** → coefficienti che quantificano la proporzione di varianza della variabile dipendente associata / spiegata dalla varianza di una o più variabili indipendenti (ad esempio, quanta variabilità del peso corporeo dipende dall'altezza, dall'attività fisica, dalle calorie ingerite): tra quelli che impareremo troviamo r, R^2 (semplice e multiplo), Odds Ratio (OR), η^2, ω^2, w^2.

Ai fini pratici, tuttavia, potremo indifferentemente usare la denominazione "indice di ES" senza tema di sbagliare.
Naturalmente, anche questi indici non sono esenti da difetti: tra i più diffusi, R^2, d di Cohen e η^2 possono dare stime *biased*, inaffidabili, quando nella distribuzione ci sono outliers o i requisiti di applicabilità dei test parametrici, che vedremo nei prossimi capitoli, sono violati (Kline, 2013).

Approfondiamone uno, per ora, probabilmente il più importante per anzianità, semplicità, comprensibilità e **coerenza** con la logica stessa dell'ES: il coefficiente **d di Cohen** (1954), che esprime la differenza tra due medie standardizzata per la variabilità intragruppo; la gran parte degli altri coefficienti di ES può essere facilmente convertita in d.

$$d = \frac{|\bar{x}_s - \bar{x}_c|}{s}$$

Quella nella formula è la **forma base** del coefficiente, riferito alla differenza in valore assoluto tra **due gruppi indipendenti**, che in questo caso sono sperimentale (*s*) e controllo (*c*). Al denominatore è inserita la **deviazione standard comune**, cioè la **media** (aritmetica se i due gruppi sono ugualmente numerosi, ponderata se i gruppi hanno diversa numerosità) d**elle sd** dei due gruppi. In pratica, il coefficiente **d standardizza la differenza tra le medie di due campioni, esprimendola in unità di deviazioni standard** (avreste dovuto cogliere una certa affinità con i punti z, effettivamente...). Il segno del coefficiente d indica solo la direzione della differenza (quale gruppo è più grande), quindi è arbitrario, mentre la grandezza dell'effetto si legge in valore assoluto.

In effetti, in effetti, tutti i coefficienti di ES valutano in sostanza questo rapporto:

$$ES = \frac{segnale\ o\ intensità\ della\ differenza}{rumore\ o\ variabilità}$$

Questa formulazione si rifà al rapporto segnale – rumore (*signal to noise ratio*), nato nel campo delle comunicazioni radio, ma trasferito in numerosi campi e affine alla teoria della detezione del segnale, che quantifica la capacità di discriminare il segnale vero e proprio, dotato di significato, in mezzo al rumore di fondo, privo di significato e confondente.

Per ciascuno dei coefficienti di ES esistono dei **criteri convenzionalmente stabiliti** che definiscono un effetto, o una relazione, come **trascurabile**, **debole**, **moderato** o **forte**. Per esempio, le soglie per il d di Cohen, secondo quanto indicato dall'autore stesso, sono:

|.0 - .20|	|.20- .50|	|.50- .80|	|> .80|
Effetto trascurabile	**Effetto debole**	**Effetto moderato**	**Effetto forte**

Tuttavia, interpretare rigidamente e acriticamente questi indici porta a commettere lo stesso errore imputato all'approccio NHST: è meglio leggerli sempre **comparativamente**, ad esempio rispetto a risultati di ricerche precedenti, oppure rispetto al disegno di ricerca. Per esempio, in un disegno realmente sperimentale, con il massimo controllo di soggetti e covariate, è lecito aspettarsi poco rumore di fondo e un segnale forte. In una ricerca sul campo, in cui il rumore di fondo è forte e non completamente controllabile, è lecito attendersi che il segnale sia "naturalmente" più debole. Il senso da attribuire a un $d = .30$ (effetto "debole") ottenuto in un disegno sperimentale e quello da attribuire allo stesso $d = .30$ in una ricerca sul campo, quindi, è ben diverso.

Anche per gli indici di ES dovrebbero essere calcolati i $\boldsymbol{CI}$ (R lo fa volentieri, come vedremo).

6.6.3 Analisi di potenza (power analysis)

Dobbiamo la diffusione (non la scoperta) della *power analysis* nelle scienze sociali a Cohen (1969).

L'analisi di potenza o **power analysis** nasce dalle argomentazioni sulla presa di decisione tra H_M e H_A di E. Pearson e Neyman (1928): i due autori hanno dimostrato che, data la grandezza della differenza tra H_M e (ovvero, l'effect size in popolazione per il parametro considerato), e fissando i valori delle probabilità associate agli errori di tipo I (α) e di tipo II (1-potenza, $1 - \beta$, è possibile **determinare a priori la numerosità campionaria $\boldsymbol{N}$** necessaria per rilevare l'effetto nel campione, se esso davvero esiste in popolazione. Per esempio, postuliamo che la differenza tra popolazione clinica e popolazione normativa rispetto ai punteggi di un test sui pensieri intrusivi sia forte (effect size $d \geq .80$), dichiariamo una soglia di rischio per l'errore di I tipo pari a $\alpha = .05$ e una soglia di rischio per l'errore di II tipo pari $\beta = .20$, quindi una potenza pari a .80. Con l'analisi di potenza possiamo stimare quale debba essere la numerosità **minima** dei campioni clinico e normativo che dovremo reclutare per ottenere una differenza non casuale nel punteggio sui pensieri intrusivi tra i due campioni, **ammesso** che nelle popolazioni clinica e normativa tale differenza esista realmente.

Alternativamente, fissati l'effect size, N e α, è possibile **determinare $\boldsymbol{\beta}$** o $1 - \beta$; fissati l'effect size, N e β, è possibile **determinare $\boldsymbol{\alpha}$**; infine, fissati N, α e β, è possibile **stimare l'effect size** che dovrebbe essere ottenuto nella ricerca. Non vi sarà difficile concludere che, quindi, **fissati tre dei parametri tra α, β, effect size e N, è possibile stimare il quarto**.

L'analisi di potenza è particolarmente rilevante quando i risultati di uno studio hanno portato a non respingere H_0, per determinare se davvero l'effetto non è esistente in popolazione o la sua grandezza è irrilevante. I risultati di Cohen (1962) rispetto alla scarsa potenza di molti studi nelle discipline psicologiche sono stati confermati successivamente

(Kazdin e Bass, 1989; Rosnow e Rosenthal, 1989; Sedlmeier e Gigerenzer, 1989): questo costituisce un problema tutt'altro che trascurabile rispetto all'avanzamento della conoscenza su base empirica.
La tecnica della power analysis non rientra nel nostro programma: tuttavia, poiché è piuttosto semplice da eseguire con R, soprattutto per i test statistici più semplici, e poiché potrebbe esservi necessaria quando affronterete il progetto di tesi, nell'Appendice II trovate alcune indicazioni per l'utilizzo del package **pwr** (per statistiche più complesse, considerate il package **pwr2**).

6.6.4 Replica e meta-analisi

La conoscenza scientifica si sviluppa attraverso la replica degli studi: i risultati di uno studio non replicato, indipendentemente dalla significatività statistica raggiunta, sono solo speculativi (Hubbard e Armstrong, 1994) e privi di significato intrinseco (Lindsay e Ehrenberg, 1993). Ciononostante, e anche se molti autori[50] hanno chiaramente delineato il ruolo fondamentale giocato della replicazione nella costruzione di conoscenze, la percentuale di studi pubblicati che contengono la replica di uno studio precedente è scarsa, nella psicologia (Hubbard e Ryan, 2000). I tentativi di riprodurre i risultati ottenuti da altri sono procedure essenziali per impedire che la letteratura di ricerca sia inondata da risultati spuri, dato che il metodo più obiettivo per verificare se il risultato di un singolo esperimento sia attendibile è la sua replica, sia esterna (facendo un nuovo esperimento) sia interna (usando metodi come la validazione incrociata – *cross validation* – o procedure di ricampionamento). Ben raramente, o mai, un singolo studio è così privo di errore di misura che i suoi risultati possono essere conclusivi: sono sempre necessarie ulteriori evidenze empiriche, ottenute con repliche esatte dello studio, che possono fornire stime più precise dell'effetto rilevato, e/o con repliche che prevedono modifiche allo studio originale, per verificare anche la generalizzabilità del risultato. Serve un numero minimo di repliche (quattro) per accertare la potenza dello studio, ovvero un minimo di quattro studi su cinque che ottengono l'esito atteso per arrivare all'80% di potenza: un'unica replica, per quanto significativa, non può essere considerata una base sufficiente per supportare o contraddire il risultato di uno studio precedente (PerezGonzalez, 2015).

Le **meta-analisi** (Glass, 1976: "***an analysis of analyses***"), come suggerisce il nome, intendono andare **oltre** (dal greco μετα) i **singoli studi**, combinando i dati provenienti da diverse ricerche, anche se ciascuna di esse non è conclusiva o è in contraddizione con il resto del corpus: lo **scopo è aumentare la potenza rispetto a singoli studi e ottenere la stima migliore dell'effetto atteso (*true effect*)** di uno specifico intervento, o dell'esposizione a un particolare fattore. Una meta-analisi è un vero e proprio articolo di ricerca, che adotta una precisa e rigorosa metodologia e tecniche statistiche che consentono di interpretare sinteticamente i suoi risultati. R, naturalmente, offre un'efficiente serie di *package* per condurre meta-analisi con relativo agio.
Tracciamo qui un veloce ritratto dei passaggi e delle particolari caratteristiche delle meta-analisi, rimandando ad altri testi e altri programmi una trattazione più estesa. Gli interessati possono fare riferimento all'ottimo manuale online del Cochrane Group, specializzato nella produzione di meta-analisi e revisioni sistematiche: https://training.cochrane.org/handbook/current

1. Si formula una precisa domanda di ricerca (ad esempio: "*L'approccio psicoterapeutico TIZIO ai sintomi del DOC è realmente più efficace dell'approccio CAIO applicato alla stessa sintomatologia?*"): fare la domanda giusta porterà alla corretta esecuzione dei passi successivi. Tra i diversi suggerimenti per formulare correttamente una domanda di ricerca è diffusa la mnemotecnica **PICO[T]**, il cui acronimo sta per :

   P – **Patient** o **Problem** o **Popolazione**: a quali pazienti / problemi / popolazioni si applica la domanda;

[50] Per esempio: Carver, 1978, 1993; Cohen, 1990, 1994; Falk & Greenbaum, 1995; Hubbard, 1995; Levin, 1998; Lykken, 1968; Robinson & Wainer, 2001; Rosnow & Rosenthal, 1989; Shaver, 1993; Thompson, 1993, 1994, 1996, 1997

I – Intervention: a quale trattamento / intervento si riferisce la domanda;

C – Control: con quale condizione o gruppo di controllo si confronta l'intervento;

O – Outcome: qual è o quale dovrebbe essere l'impatto del trattamento

[T] - Time (opzionale): se e quanti follow up sono eseguiti.

Nel nostro esempio: P–*Per i pazienti con diagnosi di DOC* I–*un intervento basato sulla mindfulness* C–*è più efficace di un trattamento di ristrutturazione cognitiva* O–*nella riduzione del rimuginio* T–*a sei mesi, uno e due anni dal trattamento?* Tra altre mnemotecniche, citiamo SPIDER (Sample-Phenomenon-Design-Evaluation-Research type, per ricerche qualitative o miste) o SPICE (Setting-Perspective-Intervention-Comparison-Evaluation, nei contesti delle scienze sociali).

2) Si **identificano gli studi rilevanti** per la domanda di ricerca, stilando a priori **criteri di inclusione ed esclusione rigorosi** – ma non eccessivamente restrittivi. Perlopiù, le meta-analisi includono ***randomized clinical trials*** (**RCT**), che offrono le migliori garanzie di controllo e randomizzazione, ma possono essere inclusi anche altri diversi disegni, purché ben fatti. Naturalmente, una scelta errata delle pubblicazioni porterà la meta-analisi a conclusioni distorte o errate, proprio come un errato campionamento di soggetti da una popolazione porterà a errori quando si vorrà generalizzare il risultato alla popolazione. Dovrebbero essere inclusi tutti i possibili **database** delle pubblicazioni *peer reviewed* relativi a un dato settore scientifico (per la psicologia, PubMed, PsychInfo, Medline sono i più utilizzati), usando accortamente i loro filtri (anno di pubblicazione, categoria di soggetti, tipo di disegno...) e inserendo nei loro motori di ricerca le corrette parole chiave, combinandole variamente con AND e OR. Dovranno essere ricercate anche le cosiddette fonti di *gray literature*: abstract, atti di convegni, capitoli di libri, tesi, registri e banche dati specifici. Lo scopo di questo processo sarà massimizzare in primis la sensibilità della ricerca (identificare più articoli potenzialmente rilevanti possibile), per dedicarsi successivamente all'ottimizzazione della sua specificità, cioè all'inclusione dei soli articoli davvero rilevanti, arrivando al **set definitivo di studi** da sottoporre alla meta-analisi.

3) **Estrazione dei dati**: dagli studi selezionati si estraggono le informazioni rilevanti, che saranno diverse, naturalmente, a seconda della domanda di ricerca. Tra le più frequenti, troviamo le variabili di **moderazione dell'effetto** (genere, età, professione, esposizione a fattori di rischio...), la misura o le misure che **operazionalizzano l'esito** (punteggi a test, mortalità o guarigione, ecc.), il **tipo di disegno**, i *follow up*, et cetera.

4) Naturalmente, i dati si presenteranno perlopiù eterogenei: le fonti di variabilità possono essere relative al tipo di partecipanti, di interventi e di outcome (**eterogeneità clinica**), o relative al tipo di disegno e agli strumenti di misura (**eterogeneità metodologica**). Da queste fonti di variabilità discende naturalmente la **variabilità negli effetti riscontrati**, ovvero l'eterogeneità statistica, o, convenzionalmente, **eterogeneità** *tout court*, che si manifesta quando gli effetti degli interventi sono più diversi l'uno dall'altro di quanto atteso in base al solo caso (errore casuale), e quindi quando bisogna sospettare l'esistenza di fonti di errore sistematico. L'eterogeneità statistica derivante dall'eterogeneità metodologica indica che gli studi soffrono di differenti gradi di ***bias***; evidenze empiriche (studi meta-epidemiologici) suggeriscono che alcuni aspetti del disegno di ricerca possono distorcerne in molti casi il risultato: valutazioni effettuate non in cieco (il ricercatore e/o il soggetto stesso sanno a quale condizione sperimentale è stato assegnato), soprattutto se sono utilizzate misure *self report*, mancata randomizzazione degli stimoli o errata sequenza degli stessi. La **valutazione della qualità metodologica** degli studi è importante, dato che un esito derivante da uno studio malfatto inserisce nella meta-analisi più distorsione di uno studio del tutto escluso (Berger e Alperson, 2009), ed è un processo complesso; per esempio, si può usare lo schema **RoB2** (Higgins et al., 2011) del Cochrane Systematic Review Groups,

che stima i bias di ogni singolo studio RCT (errata randomizzazione, casi mancanti, mortalità, misure di esito, selezione di risultati riferiti), secondo tre livelli di gravità: basso–probabile – alto rischio di bias.

5) **Analisi statistica e produzione del forest plot**: l'unità di misura della meta-analisi è l'effect size – su cui ci siamo abbondantemente soffermati - indipendentemente dalla metrica specifica di ciascun studio individuale. Può essere espresso come coefficiente r, Odds Ratio, d di Cohen, omega o eta quadrato (li vedremo tutti), e molti altri: nella meta-analisi i diversi coefficienti vengono trasformati in un unico indicatore di *effect size*, perlopiù il **coefficiente g di Hedges** (lo ritroveremo nel §11.1.2). Se uno studio non riporta espressamente una misura di ES, questa viene ricavata dalle misure campionarie riportate nell'articolo.

Da questi singoli ES si calcola una **stima sintetica dell'effetto** dell'intervento, come **media ponderata degli *effect sizes* rilevati nei singoli studi**: i **pesi**, come la numerosità campionaria o l'errore della stima, riflettono l'importanza relativa di ogni studio.

$$media_w = \frac{\sum(ES \times peso)}{somma\ dei\ pesi} = \frac{\sum Y_i W_i}{\sum W_i}$$

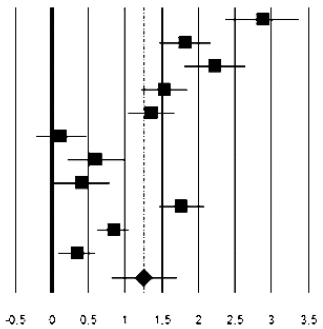
Naturalmente, se i pesi W_i assegnati a tutti gli studi sono gli stessi, ovvero se gli studi condividono la medesima importanza e qualità, la media pesata sarà uguale alla media degli effect size.

Oltre a questa forma – base, esistono vari metodi alternativi per produrre una media pesata. In generale, si possono scegliere due approcci: a **effetti fissi** (*fixed effects models*), se si ritiene che le ricerche siano molteplici repliche di uno stesso studio, in cui la misura dell'effetto atteso stimato in ogni studio è lo stessa, o a **effetti variabili** (*random – effects models*), se si ritiene che non tutti gli studi stiano stimando lo stesso effetto, ma che le stime degli effetti seguano una distribuzione tra gli studi. Ritroveremo i *fixed* e *random effects* nel capitolo 15.

In ciascun approccio si possono adottare statistiche specifiche per particolari tipologie di outcome: dicotomici (ad esempio il metodo di Mantel – Haensnzel), continui con distribuzioni fortemente asimmetriche, outcome che esprimono cambiamenti dalla baseline (*change scores*), combinazioni di outcome dicotomici e continui (ad esempio, punteggi continui a un test sulla depressione in alcuni studi e percentuali di pazienti categorizzati come depressi / non depressi in altri; Anzures – Cabrera et al., 2011), per eventi rari, i cui studi singoli soffrono generalmente di scarsa potenza, a causa della ridotta numerosità, per frequenze e proporzioni, et cetera.

La rappresentazione grafica delle meta-analisi è il **forest plot**, che mostra le stime degli effetti e relativi CI per i singoli studi e la media pesata della meta-analisi (Lewis e Clark, 2001). Ogni studio è rappresentato da un quadrato, la cui area esprime il peso assegnato allo studio, e una barra che indica l'ampiezza del CI dell'effect size. Per esempio, nella figura sottostante è rappresentato il *forest plot* di una meta-analisi relativa agli studi di efficacia nel miglioramento di atteggiamenti e credenze dei corsi di formazione alla sicurezza nel lavoro (Ricci, Chiesi, Bisio, Panari e Pelosi, 2016). Gli ES sono espressi come coefficienti g di Hedges (0= nessun effetto): la media pesata *overall* indica un effect size complessivo $g = 1.26$, $CI = .81 - 1.71$, e uno solo degli studi in analisi, peraltro piuttosto eterogenei, non ha rilevato alcun effetto significativo del corso di formazione (Stave et al., 2007a: notate il suo CI, che include 0).

	ES	95% CI	W	V	SE	Sig.
Lesch 2008a/W1	2,88	2,38 , 3,38	8,62%	0,06	0,25	0,000
Lesch 2008a/W2	1,82	1,47 , 2,18	9,09%	0,03	0,18	0,000
Lesch 2008b/W1	2,24	1,82 , 2,65	8,89%	0,05	0,21	0,000
Lesch 2008b/W2	1,54	1,23 , 1,86	9,19%	0,03	0,16	0,000
Sam et al. 2008/W1	1,37	1,06 , 1,69	9,20%	0,03	0,16	0,000
Stave et al. 2007a/W1	0,14	-0,21 , 0,49	9,10%	0,03	0,18	0,448
Stave et al. 2007b/W1	0,61	0,22 , 1,00	8,97%	0,04	0,20	0,002
Stave et al. 2007c/W1	0,42	0,03 , 0,81	8,99%	0,04	0,20	0,036
Stephenson et al. 2011/W1	1,78	1,47 , 2,09	9,21%	0,02	0,16	0,000
Stephenson et al. 2011/W2	0,85	0,62 , 1,07	9,41%	0,01	0,11	0,000
Williams et al. 2010/W1	0,35	0,09 , 0,61	9,33%	0,02	0,13	0,008
Overall (random-effects model)	1,26	0,81 , 1,71	100,00%	0,05	0,23	0,000



Per quanto accurata sia la selezione degli studi, una **meta-analisi è sempre sottoposta a *bias***: per esempio, il *publication bias* descrive la situazione in cui risultati che respingono H_0 sono pubblicati più facilmente di risultati in cui viene accettata H_1, sia per una scelta editoriale della rivista, sia per una scelta del ricercatore stesso, che non osa neppure sottoporre un risultato "negativo" alla revisione. Ne consegue che i risultati negativi possono essere sottoposti con più ritardo ed essere accettati più lentamente, con il risultato che, per esempio, le prime evidenze di mancata efficacia di un trattamento arrivano molto dopo le evidenze della sua efficacia (*time lag bias*). Da un altro punto di vista, i risultati in cui si accetta H_1 possono più facilmente produrre il *duplicate* o *multiple publication bias*, in cui uno stesso studio origina più pubblicazioni relative ai medesimi dati (Gøtzsche, 1989), portando a una sovrastima delle evidenze a favore.

Come condurre una ricerca "onesta" usando i suggerimenti dei paragrafi precedenti?

Operazionalizziamo le indicazioni dei paragrafi precedenti, usando il suggerimento di **Cumming** (2013) per fare una ricerca che verifichi ipotesi usando accortamente la statistica:

1) **Formulate le domande di ricerca in termini di stima**: chiedetevi "Quanto grande sarà l'effetto..." o "in che misura...", invece di usare espressioni che prevedono risposte dicotomiche, come "verificare l'ipotesi che non ci sia alcuna differenza tra..." o "verificare se questo trattamento sia migliore di..."
2) **Identificate l'effect size** che rappresenta la migliore risposta alle domande della ricerca: se vi interrogate sulla differenza tra due medie, allora vi serve un coefficiente come il d di Cohen; se vi chiedete quanto bene un modello rappresenti i dati, allora vi servirà una misura di bontà di adattamento o (goodness of fit: le vedremo dal capitolo 10 in poi).
3) **Esplicitate tutti i dettagli della procedura e dell'analisi** dei dati, **prima** di eseguire lo studio; usate la power *analysis* per definire la numerosità campionaria.
4) Dopo aver eseguito lo studio, **calcolate le stime puntuali e i CI dei coefficienti di effect size** prescelti.
5) **Rappresentateli graficamente**, includendo le barre d'errore dei CI
6) **Interpretateli**: nello scrivere i risultati, discutete l'intensità degli ES, che rappresenta il principale *outcome* della vostra ricerca, e l'ampiezza dei loro CI, che indica la precisione della stima. Discutete le implicazioni teoriche e pratiche dei risultati, secondo lo scopo dello studio.
7) **Pensate sempre in un'ottica meta-analitica**: raffiguratevi ogni singolo studio come fondato sulle ricerche passate quanto come un mattone con cui costruire quelle future. Presentate i vostri risultati in modo da facilitarne l'inclusione in future meta-analisi; quando necessario, usate le meta-analisi per integrare i risultati.
8) Quando **presentate i risultati**, fate una descrizione completa della ricerca e (se possibile) mettete a disposizione anche i dati grezzi. Siate completamente trasparenti su ogni passaggio, compresa l'analisi statistica, soprattutto se avete eseguito qualche forma di esplorazione o selezione dei dati.

6.6 La verifica delle ipotesi su un solo campione

Cominciamo con questo paragrafo a entrare nel cuore della verifica delle ipotesi, partendo dal caso più semplice, che vede in gioco una sola distribuzione campionaria: prima affrontiamo il confronto tra una distribuzione campionaria continua e il parametro atteso in popolazione, poi confrontiamo una distribuzione categoriale osservata con la distribuzione attesa dall'ipotesi nulla, e infine compariamo la forma della distribuzione campionaria con una distribuzione attesa (normale).

6.6.1 Un solo campione, variabile continua

Nel paragrafo 6.3 avevamo costruito i CI delle medie di NS, HA e RD nel campione di adolescenti, usandoli per stimare la differenza tra campione e popolazione normativa: nelle sue ultime righe, abbiamo menzionato la funzione `t.test`: è una funzione molto versatile, usata per:

- ✓ **confrontare una media campionaria con la media attesa in popolazione**, cioè per fare quello che abbiamo appena fatto usando i CI con i nostri adolescenti; il test prende il nome di ***t-test* per campione unico**;
- ✓ confrontare le medie di due livelli indipendenti di una variabile factor: ***t-test*** per **campioni indipendenti**;
- ✓ per confrontare due medie a misure ripetute, cioè prese sullo stesso soggetto in due condizioni diverse, per esempio prima e dopo un trattamento: ***t-test* per dati appaiati**.

Queste tre modalità richiedono argomenti diversi, che vedremo a suo tempo per il t-*test* per campioni indipendenti e dati appaiati; possiamo invece occuparci subito del t-*test* per campione unico.

La formula del t-*test* per una media è molto semplice: il **rapporto tra la differenza esistente tra media campionaria e media della popolazione** e la variabilità stimata in popolazione, cioè **l'errore standard** stimato dalla deviazione standard campionaria, si distribuisce come un quantile di una distribuzione t per $df = N - 1$.

$$t_{df=N-1} = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{N}}}$$

In pratica, possiamo generalizzare la formula come:

$$t_{N-1} = \frac{\bar{x} - modello}{variabilità\ nel\ modello}$$

Gli argomenti di **`t.test` per campione unico** sono: `x=` una distribuzione numerica, `mu=` la media attesa in popolazione; di default, H_1 è bidirezionale (`alternative= "two.sided"`; può essere cambiata con `"greater"` o `"less"` per H_1 monodirezionali in cui il valore campionario è maggiore o minore del parametro in popolazione) ed è calcolato un CI al 95%: si può cambiarlo con l'argomento opzionale **`conf.level=`** .

Per verificare se il nostro campione di adolescenti è stato estratto dalla popolazione attesa per un dato tratto di personalità, ovvero se la sua media è un'oscillazione casuale del tratto negli adolescenti (ipotesi nulla), oppure se la sua media è non casualmente differente da quella attesa, perché sarebbe un evento troppo raro se H_0 fosse vera, dovremo inserire nella funzione la distribuzione di personalità e la media in popolazione μ. Formalizzando:

- $H_0: \bar{x} = \mu$
- $H_1: \bar{x} \neq \mu$

Usiamo il tratto NS; dal paragrafo 6.3 sappiamo che la media attesa è $\mu = 20.2$, perciò:

```
t.test(ado_NS, mu = 20.2)
        One Sample t-test
data:  ado_NS
t = -20.819, df = 1268, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 20.2
95 percent confidence interval:
 17.20720 17.72267
sample estimates: → media del campione
mean of x
 17.46493
```

Cominciamo a leggere l'output partendo dal CI: la media della popolazione non è compresa nell'intervallo di confidenza: con il 95% di verosimiglianza, il campione non è rappresentativo della popolazione.

Vediamo ora la prima riga: `t=-20.819` è il quantile risultante dal rapporto tra differenza tra media empirica e media in popolazione e variabilità (stimata come deviazione standard su radice quadrata della numerosità campionaria). Se volete verificare, il calcolo è:

```
numeratore <- mean(ado_NS, na.rm = TRUE)-20.2
denominatore<-sd(ado_NS, na.rm=TRUE)/sqrt(1269)
round(t<-numeratore/denominatore,3)
[1] -20.819
```

Il $p - value$ associato a t è la **probabilità di ottenere quel risultato o uno più estremo** sotto condizione di ipotesi nulla, ovvero nell'ipotesi che il campione sia rappresentativo della popolazione. Come direbbe Fisher, nel nostro caso sembrerebbe che: "an **exceptionally rare** chance has occured", ma è più verosimile che "the **theory is not true**": il campione non è rappresentativo della popolazione. È la stessa conclusione cui eravamo giunti osservando che nel CI della media del campione non è compreso il valore della popolazione.

Notate che, per risparmiare spazio ed evitare troppi zeri, R usa la notazione scientifica per esprimere il $p - value$, e lo fa spesso: in questa modalità, un numero viene **espresso come potenza di 10**, **positiva** (ad esempio, $\mathbf{2}.190127e + 04$ equivale a scrivere $2.190127 \times \mathbf{10^4}$) o **negativa**, come in questo caso: $2.2e - 16$ equivale a $\mathbf{2.2 \times 10^{-16}}$. L'interpretazione della notazione scientifica è comunque facile: basta **spostare la virgola per un numero di cifre equivalenti alla potenza**: a **destra della prima cifra** nel caso di una potenza **positiva**, a **sinistra** nel caso di una notazione **negativa.** Quindi:

```
2.190127*10^4
[1] 21901.27
2.190127*10^-4
[1] 0.0002190127[51]
```

Si può obbligare R a **non usare la notazione scientifica** con `options(scipen= numero di zeri da stampare prima di passare alla notazione scientifica)`: si utilizza prima di fare l'analisi, in modo che R eviti la notazione scientifica per tutta la sessione di lavoro, inserendo nell'argomento `scipen` un numero molto alto, convenzionalmente = 999). Per tornare alla notazione scientifica, si ripete `options(scipen= 0)`.

Non abbiamo finito: dobbiamo sempre associare, se non sostituire, al $p - value$ un indicatore di effect size. Possiamo usare il **coefficiente d di Cohen per una media**, che al numeratore vede la differenza tra media campionaria e μ, e al denominatore la deviazione standard della distribuzione campionaria (notate l'affinità con il t-test...).

$$d_{una\ media} = \frac{|\bar{x}_Y - \mu|}{s_Y}$$

[51]Immaginate di mettere davanti alla prima cifra del risultato un numero di zeri pari alla potenza indicata e poi mettete la virgola dopo il primo zero, se vi risulta più facile pensarla così

In R esistono *package* appositi per ottenere il d per una media, anche se il calcolo è talmente facile che è più veloce impostare la formula così:

```
(mean(ado_NS, na.rm=T)-20.2)/sd(ado_NS, na.rm=T)
[1] -0.5844218
```

Dato che il segno indica solo la differenza della direzione e non la sua intensità, d si interpreta in valore assoluto; quindi, potete anche chiedere:

```
abs((mean(ado_NS, na.rm=T)-20.2)/sd(ado_NS, na.rm=T))
[1] 0.5844218
```

Ricordando le soglie convenzionali, diremmo che l'entità della differenza tra campione e popolazione è al più moderata, nonostante l'enorme numero di zeri nel $p - value$.

Per avere anche il ***CI*** di questo ES, potete usare il package `effsize` e la funzione `cohen.d(d=distribuzione, f=NA, mu=)`, che ritroveremo nei t-test per campioni indipendenti e per dati appaiati. L'argomento `f=` richiede l'inserimento di una variabile `factor`, che non partecipa al calcolo nel caso di una sola media: va quindi indicato `NA`. Ricordate di inserire `na.rm=TRUE`, se, come nel nostro caso, la distribuzione vede dati mancanti:

```
cohen.d(d=ado_NS, f = NA, mu = 20.2, na.rm = T)
      Cohen's d (single sample)
d estimate: -0.5844218 (medium)
Reference mu: 20.2
95 percent confidence interval:
     lower      upper
-0.6968928 -0.4719509
```

Intermezzo

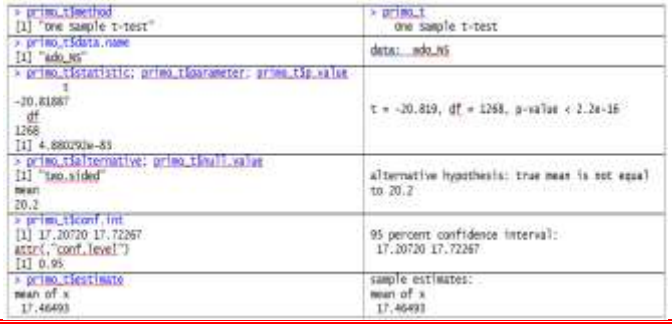
Abbiamo usato il primo (di una lunga serie di) test per verificare ipotesi che R sintetizza in una funzione. Abbiamo già visto nel capitolo 2 che queste **funzioni-test possono essere salvate come oggetti:** `oggetto_modello ← funzione.test`:

```
primo_t<-t.test(ado_NS)
class(primo_t)
[1] "htest"
```

Gli `oggetti` creati dai test che useremo noi saranno fondamentalmente di tre classi: **`htest`**`Errore. Il segnalibro non è definito.` [hypotesis test], **`lm`** [linear model], **`glm`** [generalized linear model]. Gli oggetti delle tre classi sono una lista (`list`) di elementi (`oggetto$values`): alcuni di questi sono stampati nell'output quando si richiede la funzione-test; altri, invece, possono essere selezionati nella lista degli `$values` e visualizzati a parte. Ogni funzione-test produce oggetti con elementi diversi, che vedremo quando avremo occasione di usare i test di verifica delle ipotesi.

Per il *t-test* che abbiamo appena usato, gli elementi dell'`oggetto_modello` creato da `t.test` sono:

Gli elementi che sono stati inseriti nell'output del `t.test` sono tutti quelli disponibili nella lista: `$statistic` (quantile t), `$parameter` (df), `$p.value`, `$conf.int` (CI), `$estimate` (media nel campione), `$alternative` (coda di H_1), `$null.value` (valore previsto da H_0), `$method` (tipo di *t-test*); `$data.name` (variabile).



Se volete usare `RCommander` per fare il *t-test* per un campione, scegliete Statistiche → Medie → *t-test* per un campione: dovete indicare variabile, mu, direzione dell'ipotesi alternativa e livello di verosimiglianza del CI:

Se la nostra **variabile continua avesse una distribuzione normale**, potremmo usare i quantili z di una distribuzione di probabilità normale standardizzata, invece dei quantili t, per la verifica delle ipotesi, ma il corrispondente ***z* test** non è implementato nelle funzioni di base.

È peraltro molto semplice da calcolare, dato che dobbiamo solo calcolare il quantile z dato dalla differenza tra media campionaria e μ, rapportato alla stima dello SE, per poi attribuirgli il corretto $p - value$:

$$z = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{N}}}$$

Con oltre 1000 adolescenti, è scontato che t e z saranno sostanzialmente identici, ma tanto per vedere come si fa, proviamo. Prima verifichiamo che la distribuzione sia adeguatamente normale:

```
Skew(ado_NS); Kurt(ado_NS)
[1] 0.06462953
[1] -0.5340214
```

Un po' platicurtica, ma i parametri sono entrambi sufficienti per una ragionevole normalità. Calcoliamo il quantile z e poi usiamo `pnorm` per associargli un $p - value$; serviamoci di `MeanSE` di `DescTools` per calcolare senza fatica lo SE:

```
z<-(mean(ado_NS)-20.2)/MeanSE(ado_NS)
z
[1] -20.81887
```

Il quantile z è praticamente identico al quantile t. Poiché z è negativo, in `pnorm` indichiamo `lower.tail=TRUE`, oppure usiamo il valore **assoluto** di z e indichiamo `lower.tail= FALSE`:

```
pnorm(q = z, mean = 0,sd = 1, lower.tail = TRUE)
[1] 1.459804e-96
pnorm(q = abs(z), mean = 0,sd = 1, lower.tail = FALSE)
[1] 1.459804e-96
```

Come previsto, il risultato del t-test è replicato.

1. Un campione casuale è formato da lavoratori che sono pagati con un salario medio di 4.60 euro all'ora, con uno scarto quadratico medio = .40 euro. I salari sono distribuiti in modo normale.

a) Qual è la probabilità che un lavoratore guadagni 4.50 euro o meno all'ora?
b) Qual è la probabilità di avere un campione di 20 lavoratori con un salario medio di 4.50 euro all'ora o meno?
c) Qual è la probabilità di avere un campione di 50 lavoratori con un salario medio di 4.50 euro all'ora o meno?
d) Perché le risposte alle domande a(, b) e c) sono così diverse?

2. Un campione casuale di lavoratori in un altro settore è costituito da dipendenti pagati in media 5.10 euro all'ora, con uno scarto quadratico medio di .50 euro. I lavoratori di una ditta del settore ritengono di essere sottopagati: un campione casuale di 30 lavoratori in questa ditta ha un salario medio di 4.50 euro. I dipendenti hanno ragione?

3. Verificate anche per le dimensioni HA e RD se il campione di adolescenti è rappresentativo della popolazione; non trascurate l'intensità dell'effetto.

6.6.2 Un solo campione, variabile discreta

Invece di usare una distribuzione continua come z o t, possiamo **usare anche distribuzioni non continue per verificare ipotesi** che riguardano un campione e una popolazione. Per esempio, nel file `attaccamento` avevamo notato che il **genere** dei caregiver sembra decisamente sbilanciato:

```
table(attaccamento$genere)
 F  M
35  5

prop.table(table(attaccamento$genere))
    F     M
0.875 0.125
```

Ci sono 35 donne (87.5%) e 5 uomini: $\boldsymbol{H_0}$ prevede che la proporzione dei successi nel campione (per esempio, essere donna) **sia una fluttuazione casuale della proporzione prevista in popolazione**, in cui la proporzione dei successi è uguale alla loro probabilità di verificarsi solo per caso, cioè **.50** (H_0: $prop_{♀} = .50$). **Quanto è probabile avere un campione con una proporzione di donne pari .875**, se il campione è rappresentativo di una popolazione in cui $probabilità_{♀} = probabilità_{♂} = .50$? Ce lo dice il **test della binomiale**, che **stima la probabilità del dato sotto condizione di ipotesi nulla** quando la variabile ha distribuzione binomiale, come nel nostro caso. La funzione in R è **`binom.test(x, numerosità, probabilità)`**, in cui `x=` è la numerosità dei successi oggetto del test, `n=` è la numerosità complessiva delle osservazioni e `p=` è la probabilità **teorica** dei successi. L'argomento opzionale `alternative=` consente di specificare l'ipotesi alternativa come bidirezionale (di default; nel nostro caso, la probabilità del successo è $\neq .\mathbf{50}$) o monodirezionale (nel nostro caso `"greater"` se la supponiamo $> .50$ o `"less"` se la supponiamo $< .50$).

Scriviamo:

```
binom.test(x = 35, n = 40, p = .50)
        Exact binomial test
data:  35 and 40
number of successes = 35, number of trials = 40, p-value = 1.383e-06
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.7319671 0.9581404
sample estimates:
probability of success
0.875
```

Notiamo che il CI al 95% non contiene il valore atteso secondo l'ipotesi nulla, cioè una probabilità di .50: la probabilità di essere donna nella popolazione dei caregiver, secondo il campione che abbiamo estratto, varia dal 73.2% al 95.8%.

Quindi, se si è un caregiver la probabilità di essere una donna è superiore a quella prevista dal caso (se preferite, uomini e donne non hanno la stessa probabilità di essere caregiver). Il $p - value$ dice che, se avessimo estratto il campione da una popolazione in cui uomini e donne hanno la stessa probabilità di essere presenti, la probabilità di ottenere un numero di donne $N_D = 35$ o un numero ancora maggiore, su 40 osservazioni, sarebbe infinitesimale, uguale a 0.00000138 (attenti alla notazione scientifica...) – evento decisamente raro!

Avevamo detto (§6.3) che nel caso di proporzioni il CI **non è simmetrico**, e infatti:

```
.875 - .7319671
[1] 0.1430329
.9581404 - .875
[1] 0.0831404
```

Attenzione: nel capitolo 5 abbiamo usato la distribuzione di probabilità binomiale, e in particolare la **funzione di ripartizione**, per calcolare la probabilità cumulata da un certo punto della distribuzione in su (`lower.tail= FALSE`). La logica è esattamente la stessa del test della binomiale, ma con una **differenza essenziale**: la funzione di ripartizione con `lower.tail= FALSE` calcola la **probabilità cumulata di un quantile maggiore di** x_i, cioè $P(X > x)$, mentre `binom.test` , specificando `alternative= "greater"`, calcola la probabilità di ottenere un **quantile uguale o maggiore di** x_i, cioè $P(X \geq x)$, quella corrispondente alla logica della verifica di H_0.

Nel nostro esempio, scrivendo:

```
pbinom(q = 35, size = 40, prob = .5, lower.tail = F)
[1] 9.285122e-08
```

calcoliamo la probabilità di avere **più di 35 donne su 40**, per una proporzione attesa $p_A = .50$, mentre con

```
binom.test(x = 35, n =40, p = .5,alternative = "g")
       Exact binomial test
data:  35 and 40
number of successes = 35, number of trials = 40, p-value = 6.913e-07
```

abbiamo calcolato la probabilità di avere **35 donne o più su 40**, per $p_A = .50$.

Per avere la stessa informazione con `pbinom`, dovremmo calcolare la **probabilità di avere più di 34 donne su 40**, per una proporzione attesa $= .50$:

```
pbinom(q = 34, size = 40, prob = .5, lower.tail = F)
[1] 6.91306e-07
```

oppure, più creativamente:

```
1-(pbinom(q = 34, size = 40, prob = .5))
[1] 6.91306e-07
```

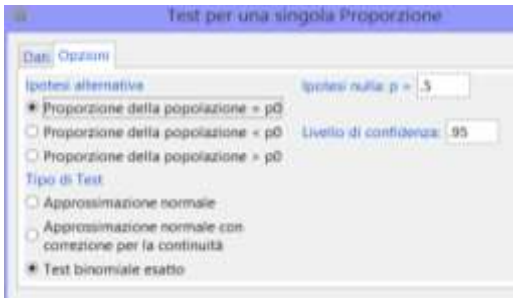
o anche (probabilità cumulata da 35 a 40):

```
sum(dbinom(35:40, size=40, prob = .50))
[1] 6.91306e-07
```

I più curiosi potrebbero essere tentati di calcolare in autonomia il CI della proporzione (?), usando la formula di Wald proposta nel §6.3: scoprirebbero che i limiti sono leggermente diversi, perché `binom.test` usa il **metodo esatto di Clopper-Pearson**, particolarmente raccomandabile in caso di eventi molto rari - $\hat{p} < .1$ - o molto comuni - $\hat{p} > .99$. Non c'è possibilità di cambiarlo in `binom.test`: se dovesse servire un diverso metodo di calcolo (?), potreste usare `BinomCI(x=, N=, method=)`, di `DescTools`, nel cui argomento `method =` possono essere indicati vari metodi alternativi (di default `"wilson"`), tra cui `"wald"` e `"clopper-pearson"`.

```
BinomCI(x = 35, n = 40, method = "wilson")
      est    lwr.ci   upr.ci
[1,] 0.875 0.7388788 0.945405

BinomCI(x = 35, n = 40, method = "wald")
      est   lwr.ci   upr.ci
[1,] 0.875 0.772511 0.977489

BinomCI(x = 35, n = 40, method = "clopper-pearson")
      est    lwr.ci    upr.ci
[1,] 0.875 0.7319671 0.9581404
```

Se volete usare `RCommander`, una volta caricato il dataframe scegliete `Statistiche` → `Proporzioni` → `test per una proporzione`: indicate la variabile oggetto dell'analisi nel folder `Dati` e `test binomiale esatto` nel folder `Opzioni`, specificando anche la direzione di H_1 e la probabilità di verificarsi dell'evento atteso (per questo esempio, .50):



Create l'oggetto `modello_genere` dalla funzione `binom.test`, verificatene la classe e prendete nota degli elementi che compongono la lista dell'oggetto `modello_genere`

Quando **una variabile categoriale non** ha distribuzione **binomiale**, possiamo verificare se le **frequenze delle categorie si presentino nel campione con una distribuzione affine a quella determinata dal solo caso** (H_0) o se almeno una delle categorie mostri un diverso numero di osservazioni rispetto a quello previsto dal caso (H_1 bidirezionale). Generalizzando:

- ✓ H_0= la forma della distribuzione dei dati è rettangolare, cioè tutte le categorie si manifestano con la medesima frequenza, ovvero la diversità delle frequenze è solo una fluttuazione casuale;
- ✓ H_1= la forma della distribuzione dei dati **non** è rettangolare, cioè le categorie di eventi si presentano con frequenze non casualmente diverse.

Per confermare o rifiutare H_0, dobbiamo usare il **test del chi quadrato a una via**Errore. Il segnalibro non è definito. (**o test di bontà dell'adattamento**; Pearson, 1900; Fisher, 1922), che si applica a una **tabella di contingenza, con una sola riga e k colonne**, in cui le osservazioni sono indipendenti, ovvero cadono all'interno di una cella O di un'altra cella (destrimani O mancini, con occhiali O senza occhiali, a favore O contro...). Il test usa la distribuzione χ^2 per attribuire un $p - value$ al risultato.

La logica del test è semplice: si calcola la **differenza** tra la frequenza empirica **osservata** (***O***) in ogni cella con la **frequenza attesa**Errore. Il segnalibro non è definito. in base al solo caso (***A***): tali scarti si chiamano **residui di cella** (**cell residuals**).

$$\chi^2_{k-1} = \sum \frac{(O-A)^2}{A}$$

Nel χ^2 a una via, la frequenza attesa è data dal **numero di osservazioni diviso per il numero di categorie**. Se fosse vera H_0, lo scarto in ogni cella sarebbe $= 0$, o comunque molto piccolo: sommando tutti i residui, avremmo una quantificazione complessiva **della distanza dei dati dallo "0" previsto da H_0 - modello**. Per questo il test del χ^2 a una via si definisce "test di bontà di adattamento/**goodness of fit**".

Notate che la formula del test può essere generalizzata nello schema: $\chi^2 = \sum \frac{(\text{osservate}-\text{modello})^2}{\text{modello}}$, mentre il residuo di ogni cella è: $res_{ij} = \text{osservate}_{ij} - \text{modello}_{ij}$

Però, come ci è già capitato con la media, la somma dei residui è $= 0$: eleviamoli, quindi, al quadrato, prima di sommarli. Inoltre, **rapportiamo ogni scarto al quadrato alla frequenza attesa** della cella, facendo una **sorta** di "standardizzazione": in questo modo, gli scarti al quadrato sono interpretabili come il numero di valori teorici compresi nello scarto. Infine, sommiamo gli scarti "standardizzati": la sommatoria dei rapporti si distribuisce come un quantile di una **distribuzione χ^2**, **con** $gradi\ di\ libertà = \boldsymbol{numero\ di\ categorie\ -1}$[52].

Guardiamo, per esempio, la distribuzione della variabile `$stato_civile` del dataframe `attaccamento`; ricordiamoci che avevamo creato il dataframe `a` da `attaccamento`, per facilità di lettura:

```
table(a$stato_civile))
   coniugato   convivente divorziato/a       single
          21            4            7            8
```

I coniugati sembrano decisamente essere prevalenti. La frequenza attesa in base al caso A è uguale a N diviso per il numero di categorie:

```
margin.table(table(a$stato_civile))
[1] 40
(A<-40/4)
[1] 10
```

Se la distribuzione dei caregiver fosse casuale, in ogni cella dovremmo aspettarci 10 soggetti; secondo l'ipotesi nulla, il fatto che invece ce ne siano 21, 4, 7 e 8 rappresenta una casuale fluttuazione rispetto alla frequenza attesa 10. Vediamo se è così: calcoliamo il χ^2:

```
((21-10)^2/10)+(4-10)^2/10+(7-10)^2/10+(8-10)^2/10
[1] 17
```

Ora stimiamo la probabilità di ottenere un $\chi^2 = 17$ o uno più grande, per $df = 3$, con la funzione di ripartizione per la distribuzione χ^2:

```
pchisq(q = 17, df = 3, lower.tail = FALSE)
[1] 0.0007067424
```

Beh, in realtà così abbiamo stimato la probabilità di ottenere un $\chi^2 > 17$, come abbiamo visto nel caso del test della binomiale: però la distribuzione χ^2, a differenza della binomiale, è **continua**; quindi, la differenza tra la probabilità cumulata da un quantile x_i in su e la probabilità maggiore o uguale a un quantile x_i è decisamente irrilevante ai fini di decidere su H_0; per esempio:

```
pchisq(q = 17 - .001, df = 3, lower.tail = FALSE)
[1] 0.0007070772
```

Ancora una volta, ottenere la distribuzione delle frequenze che abbiamo osservato sotto condizione di ipotesi nulla è decisamente un evento raro. Detto diversamente, poiché il quantile $\chi^2 = 17$ cade decisamente nella regione di rifiuto di H_0, accettiamo H_1: la distribuzione dello stato civile nei caregiver non è casuale.

R ha una funzione dedicata al test del χ^2, che si usa anche per il **test del χ^2 a due vie** che stima l'associazione tra due variabili categoriali e affronteremo nel prossimo capitolo (§7.1). Nel caso di **`chisq.test`** a una via i suoi argomenti sono **`x=`** vettore delle frequenze osservate e **`p=`** vettore delle **probabilità attese** (attenzione, non delle frequenze attese). La probabilità attesa per ogni cella è data da **1 (evento certo) diviso per il numero di categorie**. Avremo quindi:

```
osservate<-c(21,4,7,8)
```

[52] Gli scarti (O-A) nelle celle possono assumere valori indipendenti in **tutte le celle tranne una**, in cui lo scarto deve far rispettare il vincolo per cui la **somma degli scarti semplici è =0**.

```
prob_attese<-c(1/4,1/4, 1/4, 1/4)
chisq.test(osservate,p = attese)
        Chi-squared test for given probabilities
data:  osservate
X-squared = 17, df = 3, p-value = 0.0007067
```

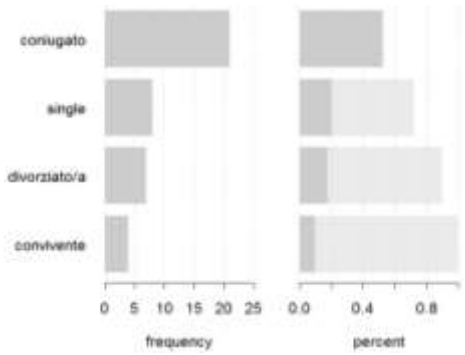
Notate che non è necessario indicare i df: R li stima dal numero di valori inseriti nei due vettori.

C'è una **scorciatoia**: se in `chisq.test` inseriamo come argomento `x=` la `table` del vettore delle frequenze osservate, R fa tutto da solo:

```
chisq.test(x= table(a$stato_civile))
        Chi-squared test for given probabilities
data:  table(a$stato_civile)
X-squared = 17, df = 3, p-value = 0.0007067
```

Applicando `Desc` di `DescTools` a una tabella di contingenza a una via, il suo output contiene, oltre alla descrizione delle frequenze e delle percentuali, anche il test χ^2 – e l'immancabile grafico:

```
Desc(table(a$stato_civile))
-------------------------------------------------------------
table(a$stato_civile) (table)
Summary:
n: 40, rows: 4
Pearson's Chi-squared test (1-dim uniform):
  X-squared = 17, df = 3, p-value = 0.0007067

          level  freq   perc  cumfreq  cumperc
1     coniugato    21  52.5%       21    52.5%
2    convivente     4  10.0%       25    62.5%
3  divorziato/a     7  17.5%       32    80.0%
4        single     8  20.0%       40   100.0%
Desc(table(a$studio))
-------------------------------------------
table(a$studio) (table)
Summary:
n: 40, rows: 3

Pearson's Chi-squared test (1-dim uniform):
  X-squared = 4.55, df = 2, p-value = 0.1028

      level  freq   perc  cumfreq  cumperc
1    laurea    13  32.5%       13    32.5%
2  maturità    19  47.5%       32    80.0%
3     media     8  20.0%       40   100.0%
```



Se volete usare `RCommander`, una volta caricato il dataframe scegliete `Statistiche` → `Informazioni riassuntive` → `Distribuzioni di frequenza`: indicate la variabile oggetto dell'analisi nel folder Dati e selezionate `Test del chi quadrato per la bontà dell'adattamento`.

6.6.3 Una sola distribuzione, ipotesi sulla forma

Nel §4.4 abbiamo usato il grafico Q-Q plot per aiutarci a definire la sovrapponibilità tra la distribuzione campionaria e una distribuzione normale teorica. In molti casi, la sovrapposizione o la mancata sovrapposizione tra i punti del grafico e la linea di riferimento è così palese da rendere facile il giudizio sulla normalità della distribuzione campionaria. In altri casi, però, può essere utile usare **anche** un test inferenziale per decidere sulla sovrapponibilità con la normale della distribuzione: tra i diversi a disposizione (in letteratura potete trovare spesso citato anche il test di **Kolmogorov – Smirnov** per un campione o il test di Lilliefors, sua evoluzione), useremo il test di **Shapiro-Wilks**.

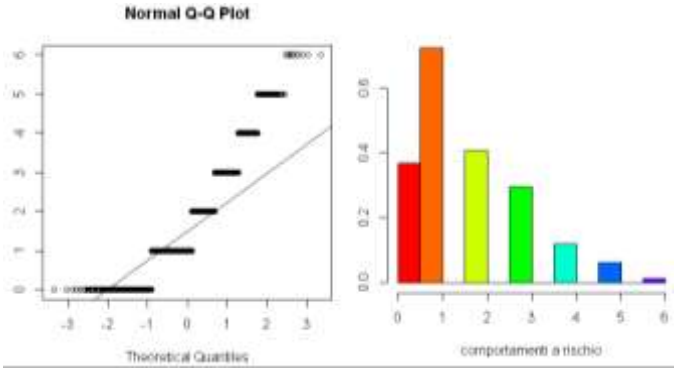
Shapiro e Wilks propongono un test (***W***) basato sulla **regressione** (**predizione**) tra i **valori osservati** e le corrispondenti

statistiche di ordine in una distribuzione normale: la statistica W può essere interpretata come il **quadrato del coefficiente di correlazione** (capitolo 8) in un Q-Q plot. Se i valori sono distribuiti in maniera tendente alla normale, la correlazione al quadrato (R^2, ovvero W) tende a 1; in caso contrario, la correlazione tende a 0.

Attenzione al $p - value$ associato alla statistica: se è **superiore alla soglia** α: $p > .05$, allora la relazione tende a 1 e quindi la distribuzione campionaria è **analoga alla normale**; se il $p - value$ è **inferiore** alla soglia α: $p \leq .05$, allora la relazione tende a 0 e la **distribuzione campionaria non è analoga alla normale**.

In R, la funzione del test di Shapiro-Wilks è semplice: `shapiro.test(distribuzione)`. Per esempio, vediamo se la **distribuzione dei comportamenti a rischio** degli adolescenti è sovrapponibile alla normale:

```
shapiro.test(ad$comportamenti_rischio)
        Shapiro-Wilk normality test
data:  ad$comportamenti_rischio
W = 0.89019, p-value < 2.2e-16
```



Il test è significativo, quindi afferma che la distribuzione non è analoga alla normale. Q-Q plot e istogramma confermano la forte asimmetria positiva.

Attenzione: il **rischio dell'eccessiva potenza** affligge anche questo test. Con molti soggetti (è il caso del campione di adolescenti) anche piccole deviazioni dalla normale possono decretare la significatività del test e quindi la definizione della distribuzione come non normale. È **sempre** indicato affiancare un grafico al test, e viceversa, per aiutare la corretta decisione. Come regola **molto empirica**, se $\boldsymbol{W > .95}$, $p < .05$ e N ampio, considerate seriamente la possibilità di un errore di I tipo e verificate graficamente con un Q-Q plot.

Capitolo 7
Distribuzioni bivariate categoriali: l'associazione

*In questo capitolo useremo il dataset **fumo**: scaricatelo da Elly, apritelo in R e leggetene la descrizione, prima di proseguire con la lettura.*

Nel capitolo precedente ci siamo occupati di descrivere e verificare ipotesi relative a una sola variabile (distribuzione). D'ora in avanti, **descriveremo e verificheremo ipotesi relative a distribuzioni bivariate**, ovvero a due variabili congiuntamente considerate. Vedremo distribuzioni bivariate composte da **due variabili categoriali**, **due variabili intervallari o due variabili ordinali**, **una variabile continua** (intervallare o ordinale) **e una variabile categoriale** a due livelli. Sarà oggetto dell'esame di Tecniche di analisi di dati II il caso di distribuzioni **multivariate**, in cui le variabili esaminate saranno più di due.

In questo capitolo ci occuperemo di una distribuzione bivariata in cui **entrambe le variabili sono categoriali**; le ipotesi saranno relative all'esistenza di una associazione[53] tra le variabili in popolazione.

7.1 Descrivere una distribuzione bivariata categoriale

Nei capitoli precedenti abbiamo descritto una sola variabile categoriale, sia numericamente usando una tabella di contingenza a una via (§3.2.1), sia graficamente con istogrammi, barplot e grafici a torta (§4.2), e abbiamo verificato ipotesi su di essa nel §6.5. In questo paragrafo, vedremo come descrivere numericamente e graficamente una distribuzione bivariata categoriale e come verificare ipotesi sull'associazione tra le due variabili categoriali.

In una distribuzione bivariata categoriale, con $k \geq 2$ categorie in X_1 e $k \geq 2$ categorie in X_2, si considerano **due modalità appaiate dello stesso caso / soggetto**, ovvero la sua appartenenza alla categoria k_a della variabile X_1 **e** alla categoria k_a della variabile X_1: si contano quanti casi del campione condividono le modalità $X_{1a}X_{2a}$, $X_{1b}X_{2a}$, $X_{1b}X_{2b}$ … $X_{1k}X_{2k}$. La distribuzione bivariata categoriale prende quindi la forma di una **tabella di contingenza a due vie**, in cui le categorie di X_1 rappresentano le righe e le categorie di X_2 rappresentano le colonne. Il caso più semplice è una tabella di contingenza $\mathbf{2x2}$.

In R, per rappresentare la distribuzione di frequenze assolute di una distribuzione bivariata useremo una tabella di contingenza a due vie generata dalla ben nota funzione `table(righe= variabile1, colonne= variabile2)`.

Esploriamo **l'associazione tra il genere** (X_1: $a = femmina, b = maschio$) e l'**essere riusciti a smettere di fumare dopo tre mesi di terapia** (X_2: $a = astinente, b = fumatore$): sono le variabili `$genere` e `$outcome_3_mesi` del dataset `fumo`.

```
table(fumo$genere,fumo$outcome_3_mesi)
      astinente fumatore
  F          32       21
  M          54       19
```

Aggiungiamo `margin.table(table)`, che conosciamo, per **calcolare i marginali**, ovvero i totali, di riga e di colonna:

```
margin.table(table(fumo$genere,fumo$outcome_3_mesi),1)
 F  M
53 73
```

[53] La relazione tra due variabili categoriali si definisce **associazione**, quella tra due variabili ordinali (cap. 8) si definisce **cograduazione** e quella tra due variabili metriche (cap. 8) **correlazione**: ma non angosciamoci troppo con i tecnicismi.

```
margin.table(table(fumo$genere,fumo$outcome_3_mesi),2)
astinente  fumatore
       86        40
```

Possiamo usare `prop.table(table)` per leggere le **frequenze relative**, ovvero le **proporzioni**: è utile interpretare le proporzioni invece delle frequenze assolute **soprattutto se il campione è sbilanciato**, come nel nostro caso. Attenzione agli argomenti di `prop.table`: oltre a `table`, dobbiamo indicare se le **proporzioni devono essere calcolate entro la variabile in riga**, **entro la variabile in colonna** o sul **totale**. Aggiungendo `margin=1`, sono usati i marginali di riga come totale su cui calcolare le proporzioni; con `margin=2` saranno usati i marginali di colonna; infine, **se non si aggiunge alcun riferimento** le proporzioni sono calcolate sul **totale delle osservazioni** (default). Scegliere quale marginale usare è, naturalmente, dipendente dal tipo di interpretazione che si intende enfatizzare, ovvero dall'ipotesi. Fate attenzione, perché all'esame gli errori nell'interpretazione di proporzioni e percentuali sono molto comuni.

`table(fumo$genere,fumo$outcome_3_mesi)`

Proporzioni entro righe

	astin.	fumat.	ast.	fum.
F	32	21	→ 32/53	21/53
M	54	19	→ 54/73	19/73

32 F astinenti e 21 F fumatrici su 53 F.
54 M astinenti e 19 M fumatori su 73 M.

```
prop.table(table(fumo$genere,
fumo$outcome_3_mesi), margin=1)
       astinente  fumatore
  F 0.6037736     0.3962264
  M 0.7397260     0.2602740
```

Il 60.4% delle F non fuma, il 39.6% fuma.
Il 73.9% degli M non fuma, il 26.1% fuma.

Proporzioni entro colonne

	ast.	fum.
F	32	21
M	54	19
	↓	↓
F	32/86	21/40
M	54/86	19/40

32 F astinenti e 54 M astinenti su 86 astinenti.
21 F fumatrici e 19 M fumatori su 40 fumatori

```
prop.table (table(fumo$genere,
fumo$outcome_3_mesi), margin=2)
     astinente   fumatore
  F   0.372093   0.525000
  M   0.627907   0.475000
```

Il 37.2% degli astinenti è F, il 62.8% è M.
Il 52.5% dei fumatori è F, il 47.5% è M.

Proporzioni sul totale

	ast.	fum.	ast.	fum.
F	32	21	→ 32/126	21/26
M	54	19	→ 54/126	19/126

32 F astinenti su 126 soggetti, 21 F fumatrici su 126, 54 M astinenti su 126 e 19 M fumatori su 126.

```
prop.table (table(fumo$genere,
fumo$outcome_3_mesi))
     astinente   fumatore
  F  0.2539683  0.1666667
  M  0.4285714  0.1507937
```

Il 25.4% dei soggetti è F e non fuma, il 16.7% è F e fuma, il 42.9% è M e non fuma, il 15.1% è M e fuma.

Possiamo usare `round` per togliere un po' di decimali, ed eventualmente moltiplicare *100 per ottenere le **percentuali**:

```
round(prop.table(table(fumo$genere,fumo$outcome_3_mesi),margin=1),3)*100
       astinente fumatore
  F       60.4     39.6
  M       74.0     26.0
```

Il 60.4% delle donne, a tre mesi dalla terapia, è astinente, contro il 74.0% dei maschi: si direbbe che siano stati gli uomini a trarre più giovamento, almeno a breve termine, dal trattamento.

Quanto è **forte la differenza tra le proporzioni** di donne astinenti e di donne fumatrici? Un coefficiente di effect size della differenza tra proporzioni è il **coefficiente h di Cohen**. Si può ricavare dai dati, come differenza tra le proporzioni **trasformate in $2 \times arcoseno$** (φ): $\varphi_i = 2 \times \arcsin\sqrt{proporzione_i} \rightarrow \boldsymbol{h = \varphi_1 - \varphi_2}$. In R, la funzione arcoseno è combinata: `asin(sqrt(x))`:

```
phi1<-2*(asin(sqrt(.604))); phi2<-2*(asin(sqrt(.396)))
(h<-phi1-phi2)
[1] 0.4190596
```

In alternativa, usiamo la `funzione ES.h(proporzione 1 - proporzione 2)` di `pwr`; abbiamo descritto il package `pwr` nella power analysis, e in effetti uno dei principali ambiti applicativi di questo h è proprio la stima della numerosità campionaria necessaria per verificare ipotesi su differenze di proporzioni o di probabilità:

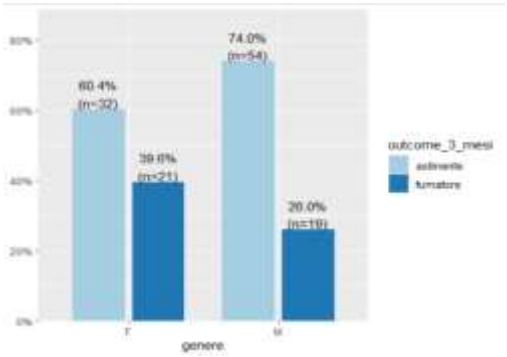
```
ES.h(p1 = .604, p2=.396)
[1] 0.4190596
```

Secondo le usuali convenzioni di Cohen, effetti fino a .2 sono trascurabili, da .2 a .5 sono deboli, da .5 a .8 discreti e da .8 in su forti. La differenza tra le proporzioni di donne astinenti e fumatrici è quindi debole, mentre è decisamente forte l'entità della corrispondente differenza tra le proporzioni nei maschi:

```
ES.h(p1 = .740, .260)
[1] 1.001309
```

Se vi trovate più a vostro agio con i grafici, in `sjPlot` la funzione `plot_xtab (x= variabile in riga, grp= variabile in colonna)` produce eleganti barplot delle percentuali; l'argomento `margin= "row"/"col"/"cell"` calcola le percentuali entro riga / entro colonna (default) / sul totale:

```
plot_xtab(x = fumo$genere, grp=fumo$outcome_3_mesi, margin = "row")
```

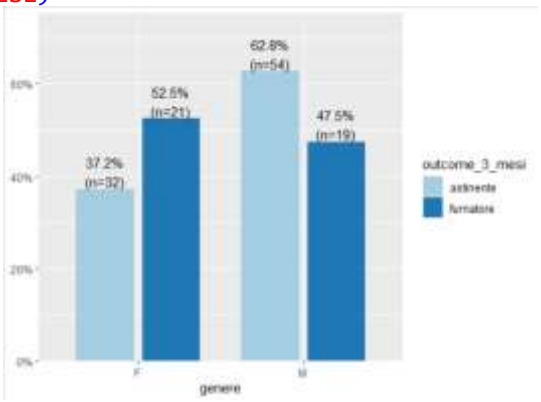


Specularmente, con le proporzioni in **colonna**:

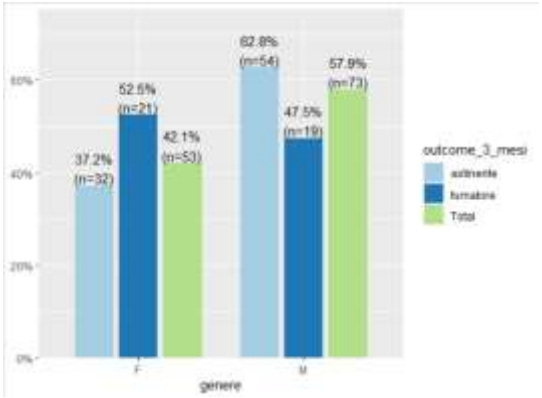
```
round(prop.table(table(fumo$genere, fumo$outcome_3_mesi), 2), 3)*100
       astinente fumatore
F       37.2     52.5
M       62.8     47.5
```

Tra gli astinenti a tre mesi dal trattamento, i maschi sono la maggioranza (62.8%); tra i fumatori, una leggera maggioranza è composta da donne (52.5%). Nel grafico di `plot_xtab`, quando il marginale indicato è `margin= "col"`, sono visualizzate anche le barre dei marginali di riga: se non le volete, indicate `show.total=FALSE`:

```
plot_xtab(x = fumo$genere, grp=fumo$outcome_3_mesi, margin = "col", show.total = FALSE)
```




```
plot_xtab(x = fumo$genere, grp=fumo$outcome_3_mesi, margin = "col")>
```

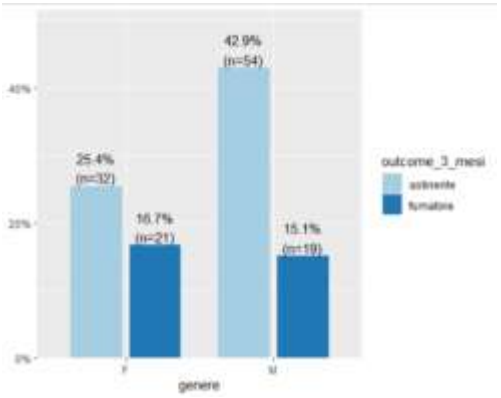


E infine, sul totale:

```
round(prop.table(table(fumo$genere, fumo$outcome_3_mesi)), 3)*100
   astinente fumatore
 F      25.4     16.7
 M      42.9     15.1
```

```
plot_xtab(x = fumo$genere, grp=fumo$outcome_3_mesi, margin = "cell")
```

A tre mesi dal trattamento, i maschi astinenti sono la categoria più rappresentata.

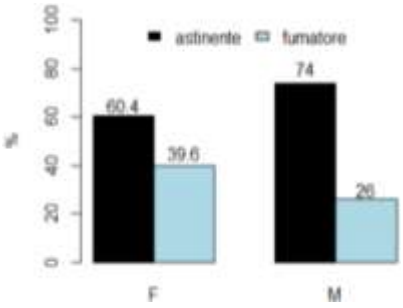


La funzione di base `barplot(prop.table)` o `barplot(table)`, che già conosciamo, produce grafici a barre efficaci, ma più essenziali. In particolare, può essere fastidiosa la posizione della legenda del grafico, che spesso si sovrappone alle barre, e quindi va gestita a parte. Si può aggiungere una legenda a un qualsiasi grafico (dopo averlo costruito) con `legend`, ma `barplot` consente di inserirla direttamente tra i propri argomenti con `legend= TRUE`. Può essere utile aggiungere anche `args.legend= list(,)`, che, tra le molte specificazioni possibili, ne definisce la posizione con `x= "top"/ "topright"/ "topleft"/ "bottom"/ "bottomright"/ "bottomleft"` (è possibile anche posizionare esattamente l'angolo sinistro della legenda con le coordinate `x=`, `y=`, in pollici). Nella stessa `list` possiamo anche gestire:

- Bordo e sfondo della legenda: `bty=` ; per non visualizzarli, `bty="n"`
- Colore del bordo dei quadratini: `border= "color"`
- Grandezza dei caratteri: `cex.names=` per le categorie in X, `cex.names=` per le voci della legenda
- Colonne in cui disporre la legenda: `ncol=` ; di default è `ncol= 1`

… e molto altro. Potete anche costruire il vettore desiderato dei colori di ogni barra e indicarlo nell'argomento `col=` del barplot:

```
colori_legenda<-c("black", "light blue")
barplot(prop.table(table(fumo$outcome_3_mesi,
   fumo$genere),2)*100, col= colori_legenda, beside= TRUE, legend=
   TRUE, args.legend = list(x="top", ncol=2, bty="n"), ylim=
   c(0,100), ylab = "%")
text(x = c(1.5,2.5,4.5,5.5), y = c(65, 45, 80, 30), labels =
   percentuali)
```

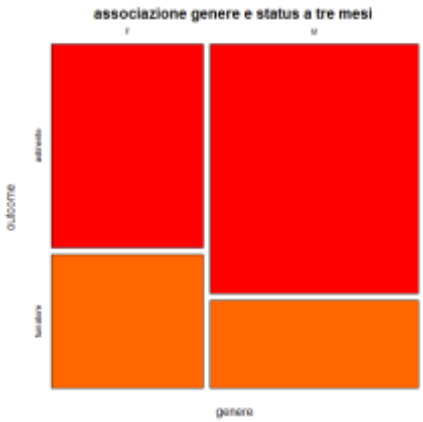


*Usate la variabile che indica lo **status del paziente dopo un anno dalla fine del trattamento**: `fumo$outcome_12_mesi` per sapere se questo apparente vantaggio a breve termine degli uomini si mantenga anche a lungo termine.*

La rappresentazione grafica più accurata di una distribuzione bivariata categoriale prevede un grafico che ancora non conosciamo: il **mosaic plot**, in cui vengono mostrati rettangoli le cui aree sono **proporzionali al numero di casi osservati** per ogni cella della tabella[54]. La funzione in R è immediatamente esplicativa: **`mosaicplot(table)`**: le categorie della variabile in riga vanno in ascissa, quelle della variabile in colonna vanno in ordinata. Gli argomenti opzionali sono i soliti (etichette degli assi, titolo, colore, ecc.).

Nel caso dell'associazione genere e status a tre mesi, avremo:

```
mosaicplot(table(fumo$genere,fumo$outcome_3_mesi),
 col=rainbow(15), xlab="genere", ylab="outcome",
 main="associazione genere e status a tre mesi")
```



Se confrontiamo le aree delle barre con le proporzioni sul totale, già viste, possiamo comprendere facilmente la logica del mosaic plot:

```
   round(prop.table(table(fumo$genere, fumo$outcome_3_mesi)), 3)
    *100
    astinente fumatore
  F      25.4     16.7
  M      42.9     15.1
```

[54] Più precisamente, alla **probabilità condizionale** associata a ogni cella

7.2 Ipotesi sull'associazione tra due variabili categoriali indipendenti

La predominanza dei maschi nella categoria astinenti, naturalmente, **potrebbe essere solo un caso**: come nel test χ^2 a una via (§6.5.2), **l'ipotesi nulla nel caso di una distribuzione bivariata categoriale è che la forma dei dati sia rettangolare**, ovvero che **tutte le categorie si manifestino con la medesima frequenza**. Invece, l'ipotesi alternativa (bidirezionale) è che la forma della distribuzione dei dati **non** sia rettangolare, cioè che le categorie di eventi si presentino con frequenze non casualmente diverse. Se **confermassimo** $\boldsymbol{H_0}$, le due variabili categoriali sarebbero **indipendenti, non associate in popolazione:** essere maschio o femmina non cambia la probabilità di riuscire a smettere di fumare dopo tre mesi dalla terapia. Se **accettassimo** $\boldsymbol{H_1}$, le due variabili categoriali sarebbero **associate in popolazione**: essere maschio o essere femmina cambia, in meglio o in peggio, la probabilità di essere astinente o fumatore dopo tre mesi. Nel mosaic plot, se fosse vera H_0, le barre avrebbero un'area identica, dato che le probabilità condizionali di ogni cella sarebbero uguali: il punto è determinare se le differenze tra le frequenze riscontrabili nella tabella di contingenza o nel mosaic plot siano **fluttuazioni casuali** di quella che è in realtà una distribuzione rettangolare in popolazione, o se esprimano una reale associazione tra le variabili categoriali in popolazione.
Bene: basta attribuire un $p - value$ (e anche un indicatore di effect size, ça va sans dire) al dato sotto condizione di ipotesi nulla per avere un **aiuto** sulla decisione da prendere.
Sono molti i test applicabili a questo tipo di disegno e i coefficienti che stimano la forza dell'associazione: vedremo solo i principali.

7.2.1 Odds e Odds ratio

Una forma di associazione molto usata nelle discipline biomediche e psicologiche, soprattutto per verificare ipotesi sull'efficacia di trattamenti (outcome favorevole), o, a contrario sull'azione di fattori di rischio nel manifestarsi di diverse patologie (outcome sfavorevole), consiste nel **mettere in rapporto la proporzione di casi di un gruppo** $\boldsymbol{G_1}$ che **presentano l'outcome** indagato con la **proporzione di casi di un gruppo** $\boldsymbol{G_2}$ **che presentano lo stesso outcome**.
Questo tipo di rapporto si definisce odds.
L'**odds**[55] (termine traducibile come "probabilità che accada un evento piuttosto che un altro", ma anche come "rapporto tra la somma che deve essere pagata per una scommessa vincente rispetto alla quota scommessa") a **favore** del verificarsi di un evento A è dato dal **rapporto tra la probabilità che l'evento** $\boldsymbol{A}$ **si verifichi :** $\boldsymbol{P(A)}$ **e la probabilità che si verifichi l'evento non atteso:** $\boldsymbol{P(\neg A)}$. Intuitivamente, l'odds contro il verificarsi di un evento A è dato dal rapporto opposto: $\boldsymbol{P(\neg A)/P(A)}$.
Vediamo il caso più semplice, con una tabella 2×2 in cui conteggiamo le frequenze di un outcome atteso ($\boldsymbol{A}$, **successo**) e di un outcome non atteso ($\neg A$, non **successo**) in due gruppi (**sperimentale** e **controllo**). Le celle della tabella contengono le frequenze di chi ha raggiunto l'outcome atteso tra gli sperimentali ($\boldsymbol{a}$) e tra i controlli ($\boldsymbol{c}$), così come le frequenze dei **fallimenti** tra gli sperimentali ($\boldsymbol{b}$) e tra i controlli ($\boldsymbol{d}$).

	Successo	Non successo
Sperimentale	$\boldsymbol{a}$	$\boldsymbol{b}$
Controllo	$\boldsymbol{c}$	$\boldsymbol{d}$

L'odds del successo per il gruppo **sperimentale** è dato dalla $P(A) \rightarrow a/(a+b)$ in rapporto alla $P(\neg A) \rightarrow b/(a+b)$.
L'odds del successo per il gruppo di **controllo** è data dalla $P(A) \rightarrow c/(c+d)$ in rapporto alla $P(\neg A) \rightarrow d/(c+d)$.
Con un piccolo ripassino sulle frazioni, possiamo semplificare:

[55] Lo ritroveremo nella regressione logistica

$$\boldsymbol{odds_{sper}} = \frac{\frac{a}{a+b}}{\frac{b}{a+b}} = \frac{a}{a+b} \times \frac{a+b}{b} = \frac{a}{\cancel{a+b}} \times \frac{\cancel{a+b}}{b} = \boldsymbol{\frac{a}{b}} \qquad \boldsymbol{odds_{crl}} = \frac{\frac{c}{c+d}}{\frac{d}{c+d}} = \frac{c}{c+d} \times \frac{c+d}{d} = \frac{c}{\cancel{c+d}} \times \frac{\cancel{c+d}}{d} = \boldsymbol{\frac{c}{d}}$$

e concludere che **l'odds del successo** per il gruppo **sperimentale** è dato da $\frac{a}{b}$ e che **l'odds del successo** per il **gruppo di controllo** è dato da $\frac{c}{d}$.

Il **range** dei possibili valori di un odds **va da 0 a infinito**: se al numeratore dell'odds c'è una cella priva di occorrenze, il rapporto sarà $= 0$; se al denominatore dell'odds c'è una cella vuota, il rapporto sarà indeterminato:

```
0/27; 27/0
[1] 0
[1] Inf
```

Decisamente importante è il passo successivo alla costruzione degli odds: il **rapporto tra due odds** si definisce **odds ratio (OR)** ed è una **misura di rischio**: esprime **quanto è più (o meno) probabile che si verifichi l'evento atteso nel gruppo G_1 rispetto a quanto è probabile nel gruppo G_2**.

$$OR = \frac{odds_{G_1}}{odds_{G_2}}$$

Come gli odds da cui è ricavato, **l'odds ratio varia da 0 a infinito**: se al numeratore c'è un $odds = 0$, il rapporto tra gli odds sarà $= 0$; se al denominatore dell'odds ratio c'è un $odds = 0$, l'odds ratio sarà indefinito. Riprendendo l'esempio del gruppo Sperimentale e del gruppo di Controllo, potremmo trovare:

- $\boldsymbol{OR = 1}$: la **probabilità di avere un successo nel gruppo sperimentale è uguale a quella riscontrata nel gruppo di controllo**: appartenere a un gruppo o all'altro **non cambia la probabilità di raggiungere l'obiettivo**.
- $\boldsymbol{OR > 1}$: la **probabilità di ottenere un successo nel gruppo sperimentale sarebbe maggiore** di quella riscontrata nel gruppo di controllo.
- $\boldsymbol{OR < 1}$: la **probabilità di ottenere un successo nel gruppo sperimentale sarebbe minore** di quella riscontrata nel gruppo di controllo.

Quindi, il **valore previsto dall'ipotesi nulla per l'OR è $OR = 1$**: in questa evenienza, gruppo sperimentale e di controllo appartengono alla stessa popolazione, rispetto alla capacità di produrre un outcome desiderato. In un tipico trial clinico, questo significa che l'intervento non ha fatto sì che i due gruppi, alla baseline appartenenti alla medesima popolazione, al termine del trattamento si ritrovassero appartenenti a due popolazioni diverse – ovvero, significa che l'intervento non è stato efficace.

È possibile **assegnare un $p - value$ all'OR ottenuto**, per stabilire se sia una **casuale fluttuazione da $H_0: OR = 1$** o se invece dovremmo accettare 1_1? Certo, **in più modi**: nel §7.3 lo vedremo fare al **test di Fisher**, nell'insegnamento di Tecniche di Analisi di Dati II lo faremo fare al metodo della **massima verosimiglianza** nella **regressione logistica**, che ci fornirà anche il suo **CI**; in entrambi i casi, sarà usata la **distribuzione di probabilità** χ^2.

Applichiamo un esempio concreto, calcolando un OR su variabili della ricerca sintetizzata nel dataframe `fumo`, in cui è stato applicato un intervento: verifichiamo se la **probabilità di ottenere un successo** (**smettere** di fumare) in chi alla baseline aveva una **forte dipendenza** da nicotina è la **stessa probabilità di smettere di fumare** per chi alla baseline aveva una **ridotta dipendenza** da nicotina ($H_0: OR = 1$; $H_1: OR \neq 1$). La gravità della dipendenza da nicotina è stimata dal test di Fagerstrom, e il livello di gravità cui appartiene il paziente è contenuto in `$Fagerstrom_categorie`. Associamo questa variabile categoriale all'esito del trattamento dopo tre mesi dalla fine dello stesso: il paziente può essere diventato astinente (successo) o restato fumatore (fallimento). Lo status dei pazienti è indicato nella variabile `$outcome_3_mesi`.

Costruiamo la tabella di contingenza 2 x 2:

```
table(fumo$Fagerstrom_categorie, fumo$outcome_3_mesi)
                     astinente fumatore
  alta dipendenza           37       28
  bassa dipendenza          49       12
```

Quindi, se trasferiamo le frequenze ottenuto nella tabella $abcd$:

	Astinente (successo)	Fumatore (non successo
Alta dipendenza	$a = 37$	$b = 28$
Bassa dipendenza	$c = 49$	$d = 12$

Ne ricaviamo che **l'odds di essere astinente per chi ha alta dipendenza** è dato dal rapporto tra $P(Astinente)$ e $P(Fumatore)$, ovvero: $\frac{a/a+b}{b/a+b} = a/b = 37/28$

```
(odds_alta_dipendenza<-37/28)
[1] 1.321429
```

L'odds di **essere astinente** per chi ha **bassa dipendenza** è dato dal rapporto tra $P(Astinente)$ e $P(Fumatore)$, ovvero:

$\frac{c/c+d}{d/c+da+b} = c/d = 49/12$

```
(odds_bassa_dipendenza<-49/12)
[1] 4.083333
```

In entrambi i gruppi, la probabilità di essere astinenti è maggiore di quella di non ricavare beneficio del trattamento, ma quella del gruppo di pazienti con scarsa dipendenza è oltre tre volte più grande: questo rapporto $1.3 : 4.1$ è appunto l'OR.

R non ha funzioni tra le statistiche di base dedicate al calcolo dell'OR; è possibile usare package dedicati (ad esempio, **oddsratio**), ma è comunque molto facile calcolarlo dai dati. Se, come nel nostro caso, si sono precedentemente calcolati gli odds, basta dividere il primo per il secondo:

```
odds_ratio<-odds_alta_dipendenza/odds_bassa_dipendenza
odds_ratio
[1] 0.3236152
```

Ma lo si può anche **calcolare direttamente dalle frequenze osservate**, sapendo che è dato dal prodotto del numero di successi nel gruppo G_1 e di fallimenti nel gruppo G_2, rapportato al prodotto del numero di fallimenti nel gruppo G_1 e di successi nel gruppo G_2. È il cosiddetto **prodotto incrociato**:

	Astinente	Fumatore	
Alta dipendenza	37	28	$OR = \frac{a \times d}{b \times c}$
Bassa dipendenza	49	12	

Perciò, se vogliamo essere raffinati, possiamo creare l'oggetto di classe `table` relativo alla tabella di contingenza in analisi e calcolare l'OR usando la struttura della tabella: moltiplichiamo la cella $r1c1_a$ per la cella $r2c2_d$, dividendo questo prodotto per il risultato della cella $r1c2_b$ moltiplicata per la cella $r2c1_c$:

```
tabella <- (table(fumo$Fagerstrom_categorie,fumo$outcome_3_mesi))
odds_ratio <- (tabella[1,1]*tabella[2,2])/(tabella[1,2]*tabella[2,1])
round(odds_ratio,3)
[1] 0.324
```

Oppure possiamo essere meno raffinati e usare le frequenze:

```
(OR<-(37*12)/(28*49))
[1] 0.3236152
```

L'odds ratio è $OR < 1$: la probabilità di essere astinenti invece che fumatori nel gruppo Alta dipendenza è di circa **tre volte minore** della probabilità di essere astinenti nel gruppo bassa Dipendenza.

È possibile **calcolare la significatività** dell'OR ottenuto (sotto condizione di $H_0: OR = 1$) usando la distribuzione normale standardizzata: però, il rapporto tra proporzioni segue una **distribuzione log-normale**, con forte **asimmetria destra**, che può essere **ricondotta alla normale** solo **facendone il logaritmo**.

Possiamo quindi calcolare: $Z_{OR} = \frac{\log(OR)}{ES_{\log(OR)}}$, dove: $ES_{\log(OR)} = \sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}$

, e attribuire un valore di probabilità allo z_{OR} ottenuto (o a uno più estremo) con la ben nota funzione di ripartizione per la distribuzione di probabilità normale standardizzata.

Grazie all'errore standard del logaritmo dell'OR possiamo anche **calcolare il suo CI** (se non contiene $H_0: OR = 1$, confermerà H_0):

$$CI_{\log(OR)} = \log(OR) \pm z\alpha_{/2}\, ES_{\log(OR)}$$

Questo CI_{log} deve ritornare alla sua base naturale con la funzione esponenziale, inverso della funzione logaritmica; in R, `exp(logaritmo)`:

$$CI_{OR} = e^{\log(OR) \pm z\alpha_{/2} ES_{\log(OR)}}$$

Calcoliamo il $p - value$ e il CI dell'OR relativo all'associazione $dipendenza\ a\ T_0 - outcome$. Cominciamo con l'errore standard del logaritmo dell'OR:

```
ES_ln<-sqrt(1/37+1/28+1/49+1/12)
```

Usiamo lo SE per calcolare il quantile z della normale standardizzata e, con la funzione di ripartizione, otteniamo il valore di probabilità di questo quantile o di uno più estremo:

```
log(odds_ratio)/ES_ln
[1] -2.76504
pnorm(-2.76504,0,1, lower.tail = TRUE)
[1] 0.002845791
```

Rifiutiamo H_0: la probabilità di riuscire a smettere di fumare per chi ha alta dipendenza è **significativamente più bassa** di quella di chi ha bassa dipendenza.

Vediamo il CI al 95%; abbiamo tutti i dati che ci servono, possiamo inserirli nella formula per calcolare i limiti superiore (UL) e inferiore (LL), ricordandoci di usare la funzione esponenziale per ridurre alla base il logaritmo da cui partiamo:

```
UL<-exp(log(odds_ratio)+(1.96*ES_ln))
LL<-exp(log(odds_ratio)-(1.96*ES_ln))
LL;odds_ratio;UL
[1] 0.1454496
[1] 0.3236152
[1] 0.7200209
```

Notate che il **CI dell'OR non è simmetrico**: come avevamo accennato nel §6.3, questo **è vero per i CI di qualsiasi proporzione** o statistica basata sulle proporzioni, come l'OR.

È possibile attribuire un $p - value$[56] anche usando la distribuzione χ^2, così come possiamo calcolare il CI dell'OR con il metodo di Woolf (1955) o il metodo di Miettinen (1985), ma possiamo evitare di approfondire ulteriormente, dato che nel §7.2.3 useremo una comoda funzione che farà tutta questa fatica per noi (potete sbirciare, se non reggete all'ansia dello spoiler) e nel Capitolo 14 la regressione logistica ci fornirà odds ratio, la loro significatività e i loro intervalli di fiducia.

Ora che abbiamo visto i passaggi del calcolo, rilassiamoci con `DescTools`: se serve solo l'OR con relativo CI, potete usare `OddsRatio(table(x1,x2), interval= .95)`; se oltre all'OR servono anche le altre descrittive della tabella di

[56] $\chi^2 = \frac{(|a - E(a)|) - 0.5^2}{var(a)}$, dove il valore atteso di A è $\frac{(a+b)(a+c)}{N}$ e *var(a)* = $\frac{(a+b)(a+c)(c+d)(b+d)}{N^2(N+1)}$ è la varianza di *a*

contingenza e i test inferenziali che affronteremo nel prossimo paragrafo, inseriamola in `Desc(table(x1,x2))`: in questo caso non serve specificare l'argomento relativo al CI, che compare di default nel ricco output:

```
OddsRatio(table(fumo$Fagerstrom_categorie, fumo$outcome_3_mesi), conf.level = .95)
odds ratio     lwr.ci     upr.ci
 0.3236152  0.1454518  0.7200103
Desc(table(fumo$Fagerstrom_categorie, fumo$outcome_3_mesi),plotit=FALSE)
----------------------------------------------------
table(fumo$Fagerstrom_categorie, fumo$outcome_3_mesi) (table)
Summary:
n: 126, rows: 2, columns: 2
Pearson's Chi-squared test (cont. adj):
  X-squared = 6.912, df = 1, p-value = 0.008562
Fisher's exact test p-value = 0.007021
McNemar's chi-squared = 5.1948, df = 1, p-value = 0.02265
                    estimate lwr.ci upr.ci'
odds ratio             0.324  0.145  0.720
rel. risk (col1)       0.709  0.555  0.906
rel. risk (col2)       2.190  1.227  3.907
[omissis]
```

Il dataframe contiene anche l'outcome a lungo termine, cioè a distanza di un anno dalla fine del trattamento: `fumo$outcome_12_mesi`. Calcolate gli odds e l'OR per questo outcome rispetto alla gravità di dipendenza: quali conclusioni potete trarre?

Anche la depressione può essere una motivazione per fumare: calcolate gli odds e l'OR degli outcome, sia a breve sia a lungo termine, considerando la variabile categoriale `$Zung_categorie` e ricategorizzando i livelli dal fattore in modo da renderlo dicotomico ("depresso" e "non depresso)": quali conclusioni potete trarre?

7.2.2 Il test chi quadrato (χ^2) a due o vie o test di indipendenza

Per verificare l'ipotesi di associazione in popolazione potremo useremo il **test del χ^2 a due vie** (Pearson, 1900; Fisher, 1922) applicato quando i **casi sono indipendenti**: chi appartiene alla categoria $X_{1k}X_{2k}$ non appartiene ad alcun'altra cella della tabella.

La formula e la logica del test sono le stesse del test χ^2a una via: confrontare le frequenze osservate (***O***) con quelle attese in base al solo caso (***A***).

$$\chi^2_{r-1\times c-1} = \sum \frac{(O-A)^2}{A}$$

Cambia il modo di calcolare le frequenze attese: la frequenza attesa di ogni cella è data dal **prodotto del marginale di riga e del marginale di colonna**, diviso per il numero totale di osservazioni. Ogni cella avrà quindi una diversa A, a differenza del test a una via.

Nel nostro esempio, potremmo divertirci (?) a calcolare le frequenze attese per ogni cella ripassando i comandi che gestiscono la struttura e le operazioni tra matrici, per stimare **l'associazione tra genere e outcome a tre mesi**. Cominciamo a creare gli oggetti `marginali_riga` e `marginali_tabella` da `table`, dando loro la struttura di **matrici con una colonna e due righe**:

```
marginali_riga<-as.matrix(margin.table(table(fumo$genere,fumo$outcome_3_mesi),1))
marginali_colonna<-as.matrix(margin.table(table(fumo$genere,fumo$outcome_3_mesi),2))
```

```
marginali_riga      marginali_colonna
  [,1]                          [,1]
F   53              astinente   86
M   73              fumatore    40
```

Ora creiamo la **matrice delle frequenza attese**: prima, la riga 1 della matrice `marginali_colonna` è moltiplicata per ciascuna riga della matrice `marginali_riga` (86*53; 86*73) e divisa per il numero totale di soggetti (126); poi la riga 2 della matrice `marginali_colonna` è moltiplicata per ciascuna riga della matrice marginali_riga (40*53; 40*73) e divisa per il numero totale:

```
attese<-cbind(marginali_colonna[1]*marginali_riga/126,marginali_colonna[2]*marginali_riga/126)
colnames(attese)<-c("astinente","fumatore")
```

Se la confrontiamo con la tabelle delle frequenze osservate:

```
attese                          table(fumo$genere,fumo$outcome_3_mesi))
  astinente fumatore      →           astinente fumatore
F   36.1746  16.8254      ←     F          32       21
M   49.8254  23.1746            M          54       19
```

notiamo che, arrotondando, ci sono 4 donne astinenti in meno e 4 donne fumatrici in più di quelle attese in base al caso; ci sono 4 uomini astinenti in più e 4 uomini fumatori in meno di quelli attesi in base al caso.

Calcoliamo i **residui di cella al quadrato**, cioè gli **scarti al quadrato tra O e A, e poi dividiamoli per A**: trasformiamo la tabella delle frequenze osservate in **matrice** per sfruttare il calcolo tra matrici:

```
osservate<-as.matrix(table(fumo$genere,fumo$outcome_3_mesi))
scarti<-(osservate-attese)^2/attese
scarti
       astinente  fumatore
  F    0.4817554 1.0357742
  M    0.3497676 0.7520004
```

Ora sommiamo gli elementi della matrice `scarti`, cioè gli scarti al quadrato standardizzati, per ricavare la statistica χ^2:

```
chi<-scarti[1,1]+scarti[1,2]+scarti[2,1]+scarti[2,2]
chi
[1] 2.619298
```

Dobbiamo calcolare la probabilità di ottenere questo quantile χ^2 o uno più estremo: i **gradi di libertà nella distribuzione bivariata** di variabili categoriali sono dati dal $\boldsymbol{numero\ di\ righe - 1}$ **per il** $\boldsymbol{numero\ di\ colonne - 1}$ (si toglie un grado di libertà per ogni variabile): nel nostro caso $(2-1)x\,(2-1) = 1$. Quindi:

```
pchisq(q = 2.619298, df = 1,lower.tail = FALSE)
[1] 0.1055711
```

L'evento "distribuzione bivariata genere per outcome a tre mesi" che abbiamo ottenuto non sembra così raro: il $p - value$ cade nella **regione di accettazione dell'ipotesi nulla** $(p - value > \alpha = .05)$, perciò dovremmo concludere che il **genere di appartenenza non sia associato alla probabilità di smettere di fumare**.

Una volta riscontrata l'esistenza di una associazione significativa, è possibile ottenere ulteriori informazioni approfondendo l'analisi **delle singole celle** della tabella; a questo scopo si calcolano i **residui di cella standardizzati** e i **residui di cella standardizzati corretti**.

I primi si ottengono dividendo lo scarto $O-A$ di ogni cella per la frequenza attesa della cella: $r_{stn} = \frac{O-A}{\sqrt{A}}$, ovvero $r_{stn_{ij}} = \frac{osservata_{ij} - modello_{ij}}{\sqrt{modello_{ij}}}$. I residui standardizzati sono interpretabili come se fossero punti z: il segno **positivo** indica che nella cella ci sono più **osservazioni di quelle attese** in base al caso; il segno **negativo** indica che ci sono **meno osservazioni di quelle attese** in base al caso. La grandezza del residuo, in valore assoluto, indica quali celle hanno maggiormente contribuito all'emergere dell'associazione.

Un'ulteriore correzione ai residui standardizzati li rende ancora più informativi: **diviso per la variabilità di tutti i residui**, ogni **residuo standardizzato corretto** si distribuisce come **un quantile z di una distribuzione normale standardizzata**.

$$r_{adj_{ij}} = \frac{O_{ij} - A_{ij}}{\sqrt{A_{ij}\left(1 - \frac{marginale_{ri}}{N}\right)\left(1 - \frac{marginale_{ci}}{N}\right)}}$$

In questo modo, è possibile usare i residui standardizzati corretti per sapere in quali celle ci sono **significativamente** più o meno frequenze osservate di quelle attese in base al caso. Basta ricordarsi il "magico" quantile $\boldsymbol{z = |1.96|}$: tutti i residui standardizzati più grandi di $|1.96|$ avranno una probabilità inferiore alla soglia $\alpha = .05$.

Prima di terminare con la teoria alle spalle del test, un ultimo dettaglio. La distribuzione di probabilità χ^2, come dovreste ricordare, è continua, ma le variabili oggetto del test non lo sono: possiamo rilevare una frequenza osservata $O = 10$ oppure $O = 11$, ma non una $O = 10.5$ oppure $O = 10.9$. Se i gradi di libertà sono > 1 e le frequenze attese in tutte le celle sono > 5 (o, meglio, > 10), questo **non rappresenta un problema** per l'interpretazione del $p - value$ associato al test. Ad esempio, la differenza tra $O = 100$ e $O = 101$ è piccola: rappresenta l'1% della frequenza osservata 100. Invece, la stessa differenza unitaria tra $O = 5$ e $O = 6$ rappresenta il 20% della frequenza osservata $O = 5$, e quindi passare da 5 a 6 è un bel salto in avanti. Inoltre, dato che non possiamo variare per unità minori dei numeri interi, qualunque residuo $O - A$, anche quando $= |1|$, apparirà grande quando le O sono piccole: la somma di grandi residui porterà a χ^2 grandi, quindi probabilmente significativi e probabilmente **afflitti da errori di I tipo**.

Per ovviare al problema, in tabelle $2x2$ $(df = 1)$ Yates propone una **correzione per la continuità** al **calcolo degli scarti**, sottraendo a ciascuno, in valore assoluto, 0.5, prima di elevarli al quadrato e dividerli per le A.

$$\chi^2_{corretto} = \sum \frac{(|O - A| - 0.5)^2}{A}$$

In questo modo, il χ^2 ottenuto è più piccolo di quello senza correzione, quindi è meno probabile che il $p - value$ associato al χ^2 o a uno più estremo cada nella regione di rifiuto di H_0, riducendo la probabilità di errori di I tipo. Però, alcune evidenze[57] dimostrano che in questo modo si ricade nell'estremo opposto, cioè in un **incremento dell'errore di II tipo**! Quindi, non preoccupatevi troppo della correzione.

I passaggi dei calcoli per il test χ^2 sembrano brutti? Un po' lo sono, ma una volta compreso il senso della formula, potrete serenamente usare **ben due funzioni di R per evitarli**. In questo paragrafo vediamo la prima, nel §3.3 la seconda.

La prima funzione è la **funzione di base** `chisq.test(variabile1, variabile2, correct= correzione per la continuità)` che abbiamo visto nel test del χ^2a una via. Attenzione agli argomenti: oltre a indicare le due variabili oggetto del test, dovete decidere se l'argomento logico `correct=` sia `TRUE` (di default) o `FALSE`. Questo argomento gestisce la correzione per la continuità di Yates.

Applicando la formula all'esempio dell'associazione tra genere e outcome (§7.2.1), il χ^2 non corretto coincide (per fortuna) con i nostri calcoli; l'output della funzione è essenziale, dato che riporta solo il test eseguito e le variabili coinvolte, il quantile χ^2, i df e il $p - value$.

```
chisq.test(fumo$genere,fumo$outcome_3_mesi, correct = FALSE)
        Pearson's Chi-squared test
data:  fumo$genere and fumo$outcome_3_mesi
X-squared = 2.6193, df = 1, p-value = 0.1056
```

[57] Per esempio, Howell, 2006

Il χ^2 con correzione per la continuità è più piccolo, e quindi il $p-value$ associato è maggiore del precedente:

```
chisq.test(fumo$genere,fumo$outcome_3_mesi, correct=TRUE)
        Pearson's Chi-squared test with Yates' continuity correction
data:  fumo$genere and fumo$outcome_3_mesi
X-squared = 2.0294, df = 1, p-value = 0.1543
```

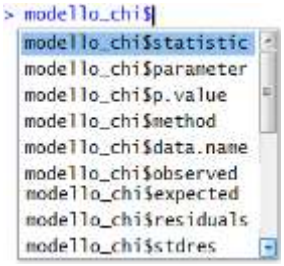
Avremmo potuto ottenerlo calcolando:

```
scarti_corretti<-(abs(osservate-attese)-.5)^2/attese
chi_corretto<-scarti_corretti[1,1] + scarti_corretti[1,2] + scarti_corretti[2,1] +
  scarti_corretti[2,2]
chi_corretto
[1] 2.029436
```

Notate la funzione **abs**, che restituisce il valore assoluto del suo argomento

Nell'output della funzione `chisq.test` mancano cose importanti: prima fra tutte, i residui standardizzati corretti, che possiamo però facilmente recuperare. Come abbiamo visto con il t-test per campione unico e il test binomiale, possiamo **salvare** `chisq.test` **come un oggetto** (la sua classe è `htest`); mentre l'output di `t.test` e `binom.test` elenca tutti gli elementi dell'oggetto, l'output di `chisq.test` ne omette un bel po'. Verifichiamolo creando l'oggetto **`modello_chi`**:

```
modello_chi<-chisq.test(fumo$genere,fumo$outcome_3_mesi)
class(modello_chi)
[1] "htest"
```



Gli elementi dell'oggetto `modello_chi` possono essere richiesti con `help(chisq.test)` oppure sfruttando i suggerimenti di RStudio:

Richiamando separatamente ciascuno di questi elementi, possiamo comporci un output "disaggregato" che contenga solo quello che ci serve. Solo i primi tre della lista sono compresi nell'output di `chisq.test`: la statistica χ^2, i df e il $p-value$.

```
modello_chi$statistic;modello_chi$parameter;modello_chi$p.value
X-squared
 2.029436
df
 1
[1] 0.1542779
```

A completamento dell'informazione precedente ci servono **i residui standardizzati corretti**, che sono gli elementi **`$stdres`** del modello (per confondere le acque, non sono denominati *adjusted*... ma lo sono!), mentre i `$residuals` sono i residui standardizzati. Quindi:

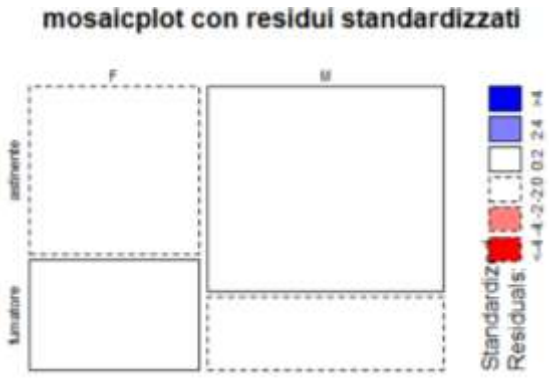
```
modello_chi$stdres
           fumo$outcome_3_mesi
fumo$genere astinente  fumatore
          F -1.618424  1.618424
          M  1.618424 -1.618424
```

Non sono superiori a $z = |1.96|$: in nessuna cella la frequenza osservata è significativamente differente dalla frequenza attesa in base al caso.

Una curiosità: indicando l'argomento `shade=TRUE` nel `mosaicplot`, il grafico viene predisposto per **colorare le tessere a seconda delll'intensità dei residui standardizzati**: **bianche** per residui standardizzati tra |0| e |2|, **azzurre e blu** per residui standardizzati positivi da 2 a >4, **rosa e rosse** per residui standardizzati negativi da -2 a < -4. Può essere utile per visualizzare la direzione della relazione e quale cella pesi di più nel determinare l'associazione, ma, non essendo rappresentati i residui standardizzati **corretti**, non si può utilizzare per inferire sulla significatività del residuo di cella.

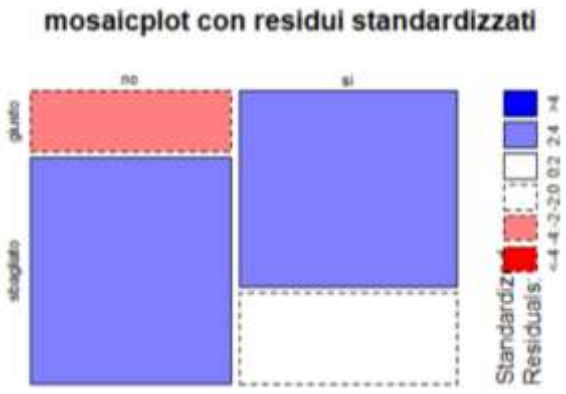
Ad esempio, questo è il mosaicplot con residui standardizzati dell'associazione `$genere` e `$outcome_3_mesi`:

```
mosaicplot(table(fumo$genere, fumo$outcome_3_mesi),
shade = TRUE, main="mosaicplot con residui
standardizzati")

modello_chi$residuals
           fumo$outcome_3_mesi
fumo$genere  astinente   fumatore
          F -0.6940860  1.0177299
          M  0.5914116 -0.8671796
```



Invece, descrivendo l'associazione tra il vivere con un gatto e riconoscere correttamente il miagolio di un gatto chiuso in un trasportino, troviamo:

```
miao<-chisq.test(gatti$vive_con_gatto,
   gatti$riconosce_miao_isolamento)
miao$residuals ← standardizzati
gatti$vive_con_gatto    giusto sbagliato
                  no -2.238831  2.048511
                  si  2.155366 -1.972142

miao$stdres    ← standardizzati corretti
gatti$vive_con_gatto    giusto sbagliato
                  no -4.212328  4.212328
                  si  4.212328 -4.212328

mosaicplot(table(gatti$vive_con_gatto,
   gatti$riconosce_miao_isolamento),   shade   =   TRUE,
   main="mosaicplot con residui standardizzati")
```



Tra coloro che non vivono con un gatto, troviamo meno riconoscimenti corretti e specularmente più errori, mentre nel gruppo di chi vive con un gatto troviamo più riconoscimenti corretti: sono queste tre celle a pesare di più nel determinare l'associazione.

7.2.3 Il test della probabilità esatta di Fisher

Il **test della probabilità esatta di Fisher** risolve un problema non irrilevante del test χ^2: con campioni relativamente grandi, la statistica χ^2 si approssima a quella di una distribuzione di probabilità χ^2, ma in piccoli campioni l'approssimazione non è adeguata, il che rende abbastanza inaffidabili i $p - value$ associati alle statistiche χ^2 calcolate su pochi soggetti. Questa difficoltà spiega perché, oltre al già citato requisito di indipendenza delle osservazioni, l'altro **requisito** (che, come spesso accade, è piuttosto contestato) **per applicare il test χ^2 è che le frequenze attese in ogni cella siano > 5**. Più il campione è grande, più è probabile che il requisito sia soddisfatto e si possa usare la distribuzione χ^2, ma quando le frequenze attese sono basse è probabile che il campione sia troppo piccolo per le esigenze del test. In tabelle di contingenza più grandi di 2×2, si considera accettabile che la **percentuale di celle con frequenze attese <5 non sia superiore al 20% delle celle della tabella**, anche se il test perde comunque in potenza, ma **nessuna** cella deve, in ogni modo, **avere frequenze attese <1** (Howell, 2006).

Nel caso in cui il requisito sia violato, se si lavora su una tabella 2×2 è meglio interpretare il **test esatto di Fisher**, che in realtà è un **metodo per stimare la probabilità esatta** usando il calcolo **combinatorio** in piccoli campioni. Gli elementi del calcolo sono i fattoriali dei quattro marginali di riga e di colonna (M), della numerosità complessiva (N) e delle frequenze osservate in ogni cella (a, b, c, d).

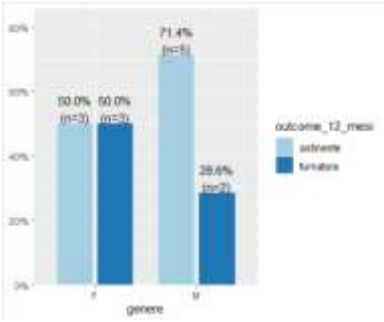
$$p_F = \frac{M_{r1}!\, M_{r2}!\, M_{c1}!\, M_{c2}!}{N!\, a!\, b!\, c!\, d!}$$

Tenuti fissi i marginali di riga e di colonna, il risultato del rapporto si distribuisce secondo la distribuzione **ipergeometrica**: la formula dà la probabilità di ottenere la disposizione delle frequenze osservate, e di ogni altra disposizione che dia altrettanta o una maggiore evidenza dell'associazione tra le due variabili, tenuti fissi (costanti) i marginali di riga e colonna. La somma delle probabilità di tutte le tabelle di contingenza di questo tipo è il $p - value$ esatto di Fisher.

Il test di Fisher si può **calcolare anche per tabelle più ampie** di 2×2, e per campioni più grandi, grazie alla sua estensione ad opera di **Freeman e Halton** (1951), ma in questi casi il test χ^2 è comunque preferibile.

Vedremo un modo diverso di leggere il $p - value$ del test, come proposto da R, nel §7.5. Per ora, usiamo la semplice funzione **fisher.test**(X1, X2, CI= verosimiglianza): notate che è possibile richiedere il 95%CI dell'OR con l'argomento **confint= verosimiglianza**, e che nell'output ci sarà proposto anche l'OR campionario (**or=TRUE**, di default), liberandoci dall'ansia di calcolarlo. Applichiamo il test di Fisher all'**associazione tra genere e outcome a 12 mesi nel solo campione che ha seguito il percorso di counseling**.

```
counseling<-subset(fumo,
   fumo$terapia=="counseling")table(counseling$genere,
   counseling$outcome_12_mesi)
      astinente fumatore
  F         3        3
  M         5        2
plot_xtab  (x=  counseling$genere,  grp=counseling$outcome_12_mesi,
   margin = "row")
```



Il campione è molto piccolo, ma sembra che gli uomini abbiano conservato un certo vantaggio.

Il test chi quadrato non è correttamente interpretabile:

```
gen_out<-chisq.test(counseling$genere, counseling$outcome_12_mesi, correct = FALSE)
Warning message:
In chisq.test(counseling$genere, counseling$outcome_12_mesi, correct = FALSE) :
  L'approssimazione al Chi-quadrato potrebbe essere inesatta
gen_out$expected
                counseling$outcome_12_mesi
counseling$genere astinente fumatore
                F  3.692308 2.307692
                M  4.307692 2.692308
```

Verichiamo, allora, l'associazione con il test di Fisher:

```
fisher.test(counseling$genere, counseling$outcome_12_mesi)
       Fisher's Exact Test for Count Data
data:  counseling$genere and counseling$outcome_12_mesi
p-value = 0.5921
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.02225065 6.32392290
sample estimates:
odds ratio
 0.4303728
```

L'OR nel campione è inferiore a 1: la probabilità di smettere di fumare per le donne è inferiore rispetto alla probabilità di smettere di fumare per gli uomini. Tuttavia, questa differenza tra le probabilità non è significativa. Concentriamoci sul CI: in popolazione, l'OR varia con il 95% di verosimiglianza da .02 a 6.32: in popolazione, è quindi possibile tanto che la probabilità di smettere sia **minore** per le donne quanto che la probabilità di smettere sia **maggiore** per le donne. Oltre a essere troppo **ampio**, l'OR in popolazione è quindi **contradditorio**. Ricordiamo una regola generale: **quando il 95%CI comprende al suo interno il valore previsto da H_0 ($OR = 1$, nel caso dell'OR), confermiamo H_0.**

A differenza del test χ^2, che prevede solo H_1 bidirezionali, il test di Fisher consente di formulare H_1 **monodirezionali**: $OR > 1$ o $OR < 1$. È necessario impostare rispettivamente l'argomento `alternative="greater"` (anche `"g"`) o `"lesser"` (anche `"l"`).

```
fisher.test(counseling$genere, counseling$outcome_12_mesi, alternative = "l")

        Fisher's Exact Test for Count Data

data:  counseling$genere and counseling$outcome_12_mesi
p-value = 0.4126
alternative hypothesis: true odds ratio is less than 1
95 percent confidence interval:
 0.000000 4.473644
sample estimates:
odds ratio
 0.4303728
```

Notiamo che nel caso di $H_1: OR < 1$, il lower limit del CI è il **minimo** teorico del range di variazione dell'OR, cioè 0 (§7.2.1); nel caso di $H_1: OR > 1$, l'upper limit del CI è il **massimo** teorico del range di variazione dell'OR, cioè in[de]finito:

```
fisher.test(counseling$genere, counseling$outcome_12_mesi, alternative = "g")

        Fisher's Exact Test for Count Data

data:  counseling$genere and counseling$outcome_12_mesi
p-value = 0.9138
alternative hypothesis: true odds ratio is greater than 1
95 percent confidence interval:
 0.03388761        Inf
sample estimates:
odds ratio
 0.4303728
```

Una modalità alternativa al test di Fisher per stimare l'associazione in piccoli campioni è il ***rapporto di verosimiglianza*** o ***likelihood ratio***, basato sulla teoria della massima verosimiglianza (Maximum Likelihood, ML), che affronteremo nella regressione logistica (capitolo 14).

7.3 Verificare ipotesi sull'associazione tra due variabili categoriali appaiate

Un requisito di applicabilità del test χ^2 a due vie è l'indipendenza dei dati: un soggetto / un caso che appartiene alla cella $X_{1a}X_{2a}$ non può essere conteggiato in alcun'altra cella della tabella. In alcuni casi, però, il disegno sottostante la ricerca è **within subjects**: è il tipico caso dei disegni **longitudinali**, in cui viene conteggiata l'appartenenza dello stesso individuo (almeno) due volte: alla baseline e a T_1. Per esempio, la prestazione di uno studente (misurata come "sufficiente" o "insufficiente") in due successive verifiche (T_0 e T_1) dello stesso esame può ricadere in quattro possibilità: $sufficiente_{T_0}$ – $sufficiente_{T_1}$ (coerente), $sufficiente_{T_0}$ – $insufficiente_{T_1}$ (incoerente), $insufficiente_{T_0}$ – $sufficiente_{T_1}$ (incoerente), $insufficiente_{T_0}$ – $insufficiente_{T_1}$ (coerente). Il campione degli studenti si distribuirà quindi:

		Seconda verifica	
		Sufficiente	***Insufficiente***
Prima verifica	***Sufficiente***	***a***	***b***
	Insufficiente	***c***	***d***

Le frequenze delle celle b e c rappresentano i soggetti che si sono dimostrati **diversi** tra prima e seconda verifica, mentre le frequenze delle celle a e d rappresentano gli studenti che non hanno mostrato cambiamenti. L'ipotesi nulla è che la prestazione di uno studente alla prima verifica sia indipendente dalla sua prestazione alla seconda verifica: quindi, se le frequenze si distribuiscono casualmente, cioè in maniera omogenea tra le quattro celle, i dati offrono sostegno a H_0. Al contrario, l'ipotesi alternativa sostiene che la prestazione alla prima verifica sia associata alla seconda: quindi,

dovremmo verificare un **addensamento delle frequenze attorno a una delle due diagonali** $\overleftrightarrow{ad}$ o $\overleftrightarrow{bc}$, e pochi casi nell'altra.

Per tabelle 2×2 di questo tipo, il test che consente di attribuire un $p - value$ ai dati sotto condizione di H_0, utilizzando la distribuzione di probabilità χ^2 per $df = 1$, è il **test di McNemar**, che si basa sulla diagonale $\overleftrightarrow{bc}$ delle celle in cui sono raccolti i casi **incoerenti** tra T_0 e T_1:

$$\chi^2_M = \frac{(b-c)^2}{b+c}$$

Nel caso di tabelle maggiori di 2×2, si usa un'estensione, il **test di McNemar-Bowker**, che valuta la **simmetria dei dati attorno alla diagonale che rappresenta le frequenze coerenti** nelle due somministrazioni.

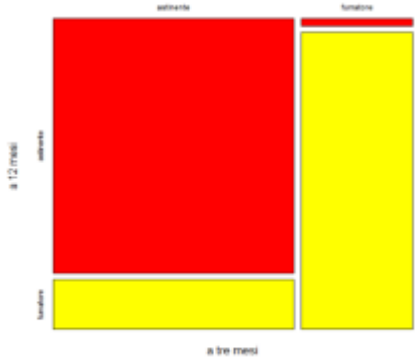
$$\chi^2_{MB} = \sum \frac{(n_{ij} - n_{ji})^2}{n_{ij} + n_{ji}}$$

Come nel test di McNemar, $\boldsymbol{H_0}$ **è verificata tramite chi ha cambiato categoria di appartenenza** tra le rilevazioni, cioè valutando le frequenze che si dispongono simmetricamente attorno alla diagonale ↘ delle frequenze coerenti: per confermare H_0, queste frequenze dovrebbero essere uguali.

		T_1		
		A favore	***Non so***	***Contrario***
T_0	***A favore***	X_{11}	X_{12}	X_{13}
	Non so	X_{21}	X_{22}	X_{23}
	Contrario	X_{31}	X_{32}	X_{33}

Per il **test di McNemar**, sia in tabelle 2×2 sia in tabelle più grandi (McNemar-Bowker), tra le statistiche di base è disponibile la semplice funzione `mcnemar.test(variabile1, variabile2, correzione di Yates)`. Applichiamola a un'informazione importante sul dataframe `fumo`: **l'effetto del trattamento dura nel tempo?** Chi è astinente a tre mesi lo rimane anche a un anno di distanza? E chi a tre mesi non era riuscito a smettere può recuperare con il tempo? H_0 è che tra l'outcome a 3 e a 12 mesi non ci sia alcuna associazione; H_1 è che in popolazione l'associazione esista. Il disegno è longitudinale, quindi il test χ^2 non è applicabile. Descriviamo l'associazione e verifichiamo l'ipotesi:

```
mosaicplot(table(fumo$outcome_3_mesi,fumo$outcome_12_mesi),
 xlab="a tre mesi", ylab="a 12 mesi", col=rainbow(6))
table(fumo$outcome_3_mesi,fumo$outcome_12_mesi)
            astinente fumatore
  astinente        72       14
  fumatore          1       39

(mcnemar<-(14-1)^2/(14+1))
[1] 11.26667

pchisq(11.26667,df = 1,lower.tail = FALSE)
[1] 0.0007891116
```



Ovvero:

```
mcnemar.test(fumo$outcome_3_mesi,fumo$outcome_12_mesi, correct = FALSE)
       McNemar's Chi-squared test
data:  fumo$outcome_3_mesi and fumo$outcome_12_mesi
McNemar's chi-squared = 11.267, df = 1, p-value = 0.0007891

mcnemar.test(fumo$outcome_3_mesi,fumo$outcome_12_mesi)
       McNemar's Chi-squared test with continuity correction
data:  fumo$outcome_3_mesi and fumo$outcome_12_mesi
McNemar's chi-squared = 9.6, df = 1, p-value = 0.001946
```

La **coerenza sembra decisamente prevalere sull'incoerenza**: quasi tutti i fumatori a tre mesi restano fumatori dopo un anno, ahimé, ma per fortuna l'84% degli astinenti a breve termine non riprende a fumare. Il test, con e senza correzione di Yates, conferma che non dovremmo confermare l'ipotesi nulla, dato che il $p - value$ è inferiore alla soglia

di significatività alfa: lo **status a breve termine è non casualmente associato**, nel bene e nel male, a **quello a lungo termine**.

7.4 Limiti dei test e coefficienti di intensità dell'associazione

Come ripetutamente sottolineato nel capitolo precedente, la significatività può, con un certo margine di errore, aiutarci a stabilire se una associazione sia o meno casuale, ovvero non presente o presente in popolazione, ma non è un indice di forza dell'associazione. Tutti i test basati sulla distribuzione χ^2 (quindi tutti quelli trattati in questo capitolo[58]) sono **molto potenti in caso di grandi campioni** (quindi espongono al rischio di errori di I tipo), quanto **poco potenti nel caso di piccoli campioni** (e quindi espongono al rischio di errori di II tipo).
Vediamo un esempio: creiamo due distribuzioni bivariate in cui la seconda (**AB_2**) è ottenuta moltiplicando per due le frequenze osservate in ogni cella della prima (**AB**). I due campioni hanno **diverse frequenze assolute, cioè diverso** N ($N_A = 100, N_B = 200$)**, ma le frequenze relative sono identiche in ogni cella**.

```
A<-c(rep("a1",50), rep("a2",50))
B<-c(rep("b1",15), rep("b2",35), rep("b1",30), rep("b2", 20))
```

```
(AB<-table(A, B))
    B
A    b1 b2
  a1 15 35
  a2 30 20
prop.table(AB)
     B
A      b1   b2
  a1 0.15 0.35
  a2 0.30 0.20
```

```
(AB_2<-AB*2)
    B
A    b1 b2
  a1 30 70
  a2 60 40
prop.table(AB_2)
     B
A      b1   b2
  a1 0.15 0.35
  a2 0.30 0.20
```

Se calcoliamo l'associazione nelle due tabelle, notiamo che il quantile χ^2 della seconda è il doppio del quantile della prima e che il $p - value$ associato più lontano dalla soglia di significatività:

```
chisq.test(AB, correct = FALSE)
        Pearson's Chi-squared test
data:  AB
X-squared = 9.0909, df = 1, p-value = 0.002569
```

```
chisq.test(AB_2, correct = FALSE)
        Pearson's Chi-squared test
data:  AB_2
X-squared = 18.182, df = 1, p-value = 2.008e-05
```

In pratica, il **quantile del test aumenta quando lo scarto tra O e A è moltiplicato per una quantità costante**, anche se le f delle classi restano uguali sia come percentuale sul totale, sia nei loro rapporti:

```
(OR_AB<-(15*20)/(35*30))
[1] 0.2857143
```

```
(OR_AB_2<-(30*40)/(70*60))
[1] 0.2857143
```

All'opposto, tabelle con N **più** ridotti, pur avendo uguali proporzioni nelle celle, rendono **più probabile accettare** H_0. Creiamo la distribuzione bivariata **CD**, in cui le frequenze relative in ogni cella sono identiche a quelle di AB e AB_2, ma le frequenze assolute sono inferiori ($N_c = 40$): il $p - value$ è appena superiore alla soglia alfa.

```
C<-c(rep("c1",20), rep("c2",20))
```

[58] Ci sono altri test e metodi di stima di associazione: correlazione tetracorica, indice D di Somers, indice t-b di Kendall in tabelle 2x2; K di Cohen, coefficienti λ e γ di Goodman e Kruskall, coefficiente di incertezza U di Theil... in tabelle *rxc*. Per vostra fortuna, non fanno parte del nostro programma.

```
D<-c(rep("d1",6), rep("d2",14), rep("d1",12), rep("d2",8))
```

```
(CD<-table(C,D))
   D
C   d1 d2
  c1  6 14
  c2 12  8
```

```
prop.table(CD)
      D
C       d1   d2
  c1 0.15 0.35
  c2 0.30 0.20
```

```
(OR_CD<-(6*8)/(14*12))
[1] 0.2857143
```

```
chisq.test(CD, correct = FALSE)
        Pearson's Chi-squared test
data:  CD
X-squared = 3.6364, df = 1, p-value = 0.05653
```

Perciò (come anche nel caso in cui l'interesse del ricercatore sia quello di confrontare l'intensità di associazione delle stesse variabili in campioni diversi), il test dovrà essere affiancato da **indici**, perlopiù **ponderati per N**, che stimino **l'intensità dell'associazione** facendo riferimento a un range di variazione noto e comune, **da 0** (indipendenza) **a 1** (perfetta associazione). Ne elenchiamo solo alcuni tra quelli a disposizione in letteratura: vediamo le facili procedure di calcolo derivanti dalle loro formule, ma potremo ottenerli sfruttando le funzioni contenute in `DescTools`.

Salviamo il test condotto sulla tabella `AB` come oggetto, per evitare di riportare il valore di χ^2 nelle formule successive.

```
chi_AB<-chisq.test(AB, correct = FALSE)
```

Storicamente, il primo coefficiente di intensità dell'associazione è stato il **coefficiente di contingenza C di Pearson**, dato dalla radice quadrata del valore χ^2 (senza correzione di Yates) diviso per $\chi^2 + N$.

$$C = \sqrt{\frac{\chi^2}{\chi^2 + N}}$$

Il suo problema è che il **limite superiore di variazione** è dato da $\sqrt{(k-1)/k}$, in cui k è il **numero minore** tra righe e colonne: è facile verificare che **tende a (ma non raggiunge) 1 solo per tabelle molto grandi**, tanto che si suggerisce di interpretarlo per tabelle almeno 5×5. Proviamo a calcolare il limite superiore (C_{max}) per alcuni k:

C_{max} se $k = 2$	C_{max} se $k = 3$	C_{max} se $k = 5$	C_{max} se $k = 12$
`sqrt((2-1)/2)`	`sqrt((3-1)/3)`	`sqrt((5-1)/5)`	`sqrt((12-1)/12)`
`[1] 0.7071068`	`[1] 0.8164966`	**`[1] 0.8944272`**	`[1] 0.9574271`

Il **problema è stato risolto** da **Sakoda**: la sua correzione (C_{adj}) consente al range di variazione di C di estendersi fino a 1, indipendentemente dalla grandezza della tabella.

$$C_{adj} = \frac{C}{C_{max}}$$

Quindi, il C **non corretto** di $A \times B$ è:

```
sqrt(chi_AB$statistic/(chi_AB$statistic+100))
0.2886751
```

Ma, **applicando la correzione**, diventa:

```
sqrt(chi_AB$statistic/(chi_AB$statistic+100)) / 0.7071068
0.4082483
```

Nella funzione `ContCoef(x1, x2, correct = TRUE / FALSE)` basta specificare le due variabili e indicare `correct=TRUE` per ottenere il coefficiente C_{adj}. Se avete salvato precedentemente la tabella di contingenza come oggetto, potete inserirla nella funzione al posto delle due distribuzioni.

```
ContCoef(A,B,correct=FALSE)
[1] 0.2886751
```

```
ContCoef(A,B,correct=TRUE)
[1] 0.4082483
```

```
ContCoef(AB,correct=TRUE)
[1] 0.4082483
```

C può essere usato anche per **stimare quale sia l'apporto di N alla significatività del test χ^2**: lo propone Cohen (1977) con l'**indice w.**

$$w = \sqrt{\frac{C^2}{1 - C^2}}$$

Cohen suggerisce (ma sono limiti arbitrari!) che l'effetto di N sulla significatività del test è **piccolo** (*small*) se $w \leq .30$, medio (*medium*) se compreso **tra $w = .30$ e $w = .50$**, **grande** (*large*) se $w > .50$.

```
C_AB<-ContCoef(A,B,correct=FALSE)
w_AB<-sqrt(C_AB^2/(1-(C_AB^2)))
w_AB
[1] 0.3015113
```

In tabelle 2×2 che contengono dati realmente dicotomici[59], il **coefficiente phi** di Pearson (φ, coefficiente di **correlazione punto-tetracorica**) esprime la media geometrica delle differenze tra le proporzioni del fattore nelle righe e il fattore nelle colonne.

$$\varphi = \frac{(ad) - (bc)}{\sqrt{n_1 \times n_2 \times n_3 \times n_4}}$$

Può essere ricavato direttamente dal quantile χ del test, calcolato senza la correzione di Yates, velocizzando il calcolo.

$$r_\varphi = \sqrt{\frac{\chi^2}{N}}$$

Il **range di variazione di phi va da 0 a $\sqrt{(min(r, c) - 1)}$**, ovvero la radice quadrata del numero più piccolo tra righe e colonne meno 1. Evidentemente, quindi, in tabelle 2×2, o con due righe o due colonne il suo range di variazione **va da 0 a $\sqrt{(2 - 1)} = 1$,** rendendone semplice l'interpretazione; in **tabelle più grandi, invece, il limite superiore della variazione è maggiore di 1** (per esempio, in una tabella 3×3: $\sqrt{(3 - 1)} = 1.4$) **e dipende dalla grandezza della tabella**, il che lo rende **inadeguato** per l'interpretazione dell'intensità dell'effetto.
Applicato alla tabella AB, conferma che la forza dell'associazione è debole:

```
sqrt(chi_AB$statistic/100)
0.3015113
```

Possiamo usare la funzione `Phi(x1, x2)` o `Phi(table(x1,x2)`:

```
Phi(A, B); Phi(AB)
[1] 0.3015113
[1] 0.3015113
```

Avete notato? Il coefficiente phi, in **tabelle 2×2**, **coincide sempre con il w di Cohen**: il suo valore interpretativo si arricchisce, quindi.

Per il problema del range di variazione indefinito, in tabelle $r \times c$ possiamo preferire il **coefficiente V[60] di Cramér** (1946) che, come si intuisce facilmente dalla formula, è l'estensione del coefficiente φ:

$$V_C = \sqrt{\frac{\varphi}{min(r - 1, c - 1)}} = \sqrt{\frac{\chi^2}{N(k - 1)}}$$

k è il minore tra il numero di righe e il numero di colonne della tabella di contingenza (naturalmente, in tabelle 2×2, oppure in cui $r = 2$ o $c = 2$, φ e V coincidono, dato che $k - 1 = 2 - 1 = 1$).
V ha dei problemi: **se i marginali di riga e colonna sono uguali** (cioè, se il disegno è bilanciato), i suoi limiti si **approssimano bene a 0 e 1** – ma, in caso contrario, il limite inferiore è leggermente > 0 e quello superiore non

[59] Se i dati originassero da distribuzioni continue dicotomizzate, si dovrebbe preferire la correlazione tetracorica, sempre di Pearson (1901).
[60] V^2 è una statistica a sé stante, la **correlazione canonica media al quadrato**, che non fa parte del nostro programma.

raggiunge 1. Il bias è particolarmente sensibile per piccoli campioni, e porta a sovrastimare la reale associazione in popolazione.

```
sqrt(chi_AB$statistic/(100*(2-1)))
X-squared
0.3015113
```

Useremo `CramerV(x1, x2)` o `CramerV(table(x1,x2)`. L'argomento `correct = TRUE` (di default è `FALSE`) consente di apportare la correzione al bias proposta da **Bergsma** (2013), che compensa la sovrastima dell'associazione modificando la stima di phi e di k. Questa è l'unica, tra le funzioni che `DescTools` dedica ai coefficienti di associazione, a consentire la visualizzazione del CI, con `conf.level= alfa`.

```
CramerV(A,B)
[1] 0.3015113
CramerV(A,B, correct=TRUE,conf.level = .95)
 Cramer V    lwr.ci    upr.ci
0.2857143 0.1054848 0.4975080
```

Non ha invece problemi di variazione il **coefficiente *Q* di Yule** per tabelle 2×2 (Yule, 1900; Q è un omaggio a Quetelet, capitolo 4), che si si ricava dall'***OR*** della tabella:

$$Q = \frac{(ad) - (bc)}{(ad) + (bc)} = \frac{OR - 1}{OR + 1}$$

Il range va da $Q = 0$ se l'associazione è assente in popolazione a $Q = 1$ se l'associazione è perfetta, e non dipende dalle distribuzioni marginali come V (Warren, 2008). Anche se può avere segno negativo, per l'interpretazione usiamo il valore assoluto.

```
AB
        B
A     b1 b2
  a1 15 35
  a2 30 20
((15*20)-(35*30))/((15*20)+(35*30))
[1] -0.5555556
```

Oppure:

```
OR<-(15*20)/(35*30)
(OR-1)/(OR+1)
[1] -0.5555556
```

La funzione `YuleQ(x1, x2)` o `YuleQ(table(x1,x2))` è molto semplice:

```
YuleQ(AB)
[1] -0.5555556
```

Q non richiede l'assunzione di una sottostante distribuzione normale bivariata come il coefficiente χ^2, ed è stato terreno di una lunga, aspra e irrisolta tenzone tra Yule e Karl Pearson (alcuni esempi dei giudizi che si ripetono per 160 pagine esclusivamente dedicate a Yule: "*If Mr. Yule's views are accepted, irreparable damage will be done to the growth of modern statistical theory*"; "*We feel convinced he will have to withdraw [his theory], if he wishes to maintain any reputation as a statician*"; Pearson e Heron, 1913, pagg. 159; 301).

Ci possiamo fermare qui, anche se i coefficienti di associazione, per tabelle 2×2 o $r \times c$, sono molti di più.

7.5 Altre funzioni per i test di associazione in R

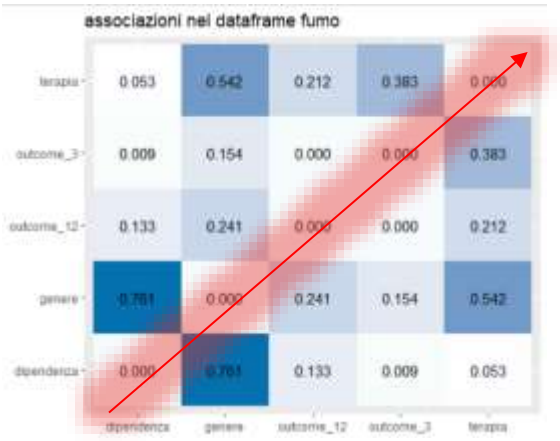
Se dovete fare una **lunga serie di test** χ^2, e volete una rozza scrematura dei soli test significativi, per poi approfondirli, potete usare `sjp.chi2(variabili da associare)` di `sjPlot`: il suo oggetto è un dataframe composto dalle sole

variabili da associare a coppie, e il suo prodotto è un grafico. Nel **grafico**, si legge la matrice dei **soli** $p - value$ di ogni associazione a coppie: è quindi ridondante attorno alla diagonale in cui ogni variabile è associata con se stessa, come la matrice delle correlazioni che vedremo nel prossimo capitolo. La scala di colori indica una graduazione nella lontanza dalla soglia di significatività, dal bianco (significativo) al tono più scuro.

Vediamo, per esempio, le associazioni tra le molte variabili categoriali di `fumo`: prima le uniamo in un dataframe, che poi inseriamo nella funzione `sjp.chi2`:

```
fum<-data.frame(terapia= fumo$terapia, genere=fumo$genere, dipendenza=
   fumo$Fagerstrom_categorie, outcome_3=fumo$outcome_3_mesi, outcome_12=fumo$outcome_12_mesi)
```

```
sjp.chi2(fum)
```



La terapia ha una sola, debole associazione significativa con la gravità della dipendenza da nicotina; l'outcome a tre mesi è associato alla gravità della dipendenza e all'outcome a 12 mesi (ma qui sarebbe opportuno usare il test di McNmar), il genere non è associato a nulla.

Se desiderate un'unica funzione che racchiude test del chi quadrato, con tutte le informazioni che desiderate, e test di Fisher, compreso OR e relativo CI, vi serve `CrossTable (molti argomenti)` di `gmodels`. Un suo limite è che `CrossTable` esegue i test di McNemar e di Fisher solo per tabelle 2×2, mentre `fisher.test` e `mcnemar.test` si applicano anche tabelle $r \times c$.

Gli argomenti di `CrossTable` sono tanti: ciascuno riempie la tabella di contingenza proposta nell'output di elementi più o meno utili per l'interpretazione del risultato; disgraziatamente, per molti la funzione di default è `TRUE`, il che impegna a un lavoro di costruzione della funzione con svariati argomenti `= FALSE` per rendere più leggibile il risultato:

Sempre utili

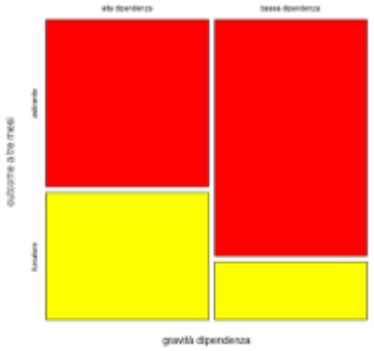
x= variabile in riga; **y=** variabile in colonna
`expected=` TRUE per le frequenze attese
`prop.r` / `prop.c` / `prop.` `T= TRUE` per le percentuali sui marginali di riga / di colonna / sul totale
`chisq= TRUE` oppure `fisher= TRUE` in tabelle 2x2 oppure `mcnemar= TRUE` per i test
`asresid TRUE` più `format= "SPSS"` per i residui standardizzati corretti

Opzionali

`digits` = numero di decimali
`prop.chisq= TRUE/FALSE` contributo di ogni cella al chi quadrato (le celle con i residui maggiori)
`resid= TRUE/FALSE`, residui di cella
`sresid= TRUE/FALSE`, residui standardizzati di cella

Mettiamo alla prova `CrossTable` sull'ipotesi che **la gravità della dipendenza da nicotina al momento di entrare in trattamento sia associata all'esito a breve termine dello stesso**: l'odds ratio (§7.2.1) ci aveva fatto propendere per il sì, dato che, essendo <1, indicava che tra i poco dipendenti la probabilità di essere astinenti invece che fumatori (c/d) era maggiore rispetto a quella riscontrata tra i molto dipendenti (a/b):

```
mosaicplot(table(fumo$Fagerstrom_categorie,    fumo$outcome_3_mesi),
xlab="gravità dipendenza", ylab="outcome a tre mesi", col=rainbow(6))
```



Si direbbe che chi è poco dipendente abbia più probabilità di trovarsi astinente dopo mesi e poca di essere fumatore; chi è molto dipendente ha una probabilità di essere astinente di molto poco più alta che di continuare a fumare. Un'associazione tra le due variabili sembra plausibile.

In `CrossTable` indichiamo che vogliamo vedere, oltre alle frequenze osservate, quelle attese, le proporzioni per riga, il test χ^2 e il test di Fisher, i residui standardizzati corretti; usiamo due soli decimali:

```
CrossTable(x = fumo$Fagerstrom_categorie,y =fumo$outcome_3_mesi ,digits = 2,expected = TRUE,prop
  .r = TRUE,prop.c = FALSE,prop.t = FALSE,prop.chisq = FALSE,chisq = TRUE,fisher = TRUE,mcnemar =
  FALSE,resid = FALSE ,sresid = FALSE,asresid = TRUE,format = "SPSS")
```

L'output è ricco:

```
Cell Contents                    ← Nelle celle dell'output vedremo O, A, le % per  riga e i residui standardizzati corretti
|-------------------------|
|                   Count |
|         Expected Values |
|             Row Percent |
|           Adj Std Resid |
|-------------------------|
Total Observations in Table:126     ← In totale, abbiamo 126 casi non mancanti

                                fumo$outcome_3_mesi
fumo$Fagerstrom_categorie | astinente |   fumatore | Row Total |
--------------------------|-----------|------------|-----------|
          alta dipendenza |        37 |         28 |        65 |
                          |     44.37 |      20.63 |           |
                          |     56.92% |      43.08% |     51.59% |
                          |     -2.82 |       2.82 |           |
--------------------------|-----------|------------|-----------|
         bassa dipendenza |        49 |         12 |        61 |
                          |     41.63 |      19.37 |           |
                          |     80.33% |      19.67% |     48.41% |
                          |      2.82 |      -2.82 |           |
--------------------------|-----------|------------|-----------|
             Column Total |        86 |         40 |       126 |
--------------------------|-----------|------------|-----------|
```

All'ingresso in trattamento, i pazienti sono piuttosto equamente suddivisi tra chi aveva alta (65; 51.6%) e bassa (61; 48.4%) dipendenza. Dopo tre mesi, il 56.9% dei pazienti più problematici ha smesso di fumare; ancora più alta è la percentuale di chi ha smesso di fumare tra chi aveva una bassa dipendenza: 80.3%. Secondo i **residui standardizzati corretti**, il numero di chi ha smesso di fumare ed era molto dipendente dalla nicotina è **significativamente più basso** di quello atteso in base al caso ($|2.82| > |1.96|$); specularmente, il numero di chi ha smesso di fumare ed era poco dipendente è **significativamente più alto** di quello atteso in base al caso. Ovviamente, tra i fumatori si trova l'andamento opposto. Sembra che l'associazione tra dipendenza all'ingresso e outcome sia realistica.

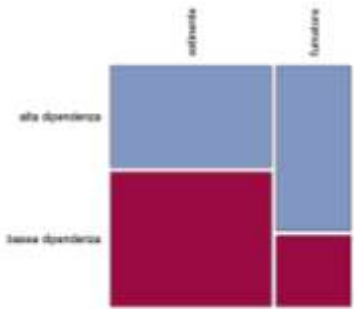
```
Statistics for All Table Factors
Pearson's Chi-squared test
------------------------------------------------------------
Chi^2 =  7.955452     d.f. =  1     p =  0.004794273

Pearson's Chi-squared test with Yates' continuity correction
------------------------------------------------------------
Chi^2 =  6.911959     d.f. =  1     p =  0.008562115
```

`CrossTable` riporta il test χ^2 con e senza la correzione di Yates; in entrambi i casi, il risultato sembra piuttosto solido: **rifiutiamo H_0**, le due variabili sono associate in maniera non casuale.

```
Fisher's Exact Test for Count Data
-----------------------------------------------------------
Sample estimate odds ratio:  0.3265885

Alternative hypothesis: true odds ratio is not equal to 1
p =  0.00702098
95% confidence interval:  0.1324196 0.76718

Alternative hypothesis: true odds ratio is less than 1
p =  0.004012281
95% confidence interval:  0 0.6793766

Alternative hypothesis: true odds ratio is greater than 1
p =  0.9988408
95% confidence interval:  0.1519279 Inf
```

Come in `fisher.test`, nel test di Fisher abbiamo la significatività dell'OR (ma non il suo valore), per tutte le H_1 possibili. Come anticipato nel paragrafo precedente, anche `Desc` di `DescTools`, applicata a una tabella di contingenza a due vie, è molto informativa. Nel caso di una **tabella 2×2**, presenta i test di significatività (test χ^2, test di Fisher e di McNemar), l'OR con CI, tre coefficienti di intensità dell'associazione (phi, C, V); addirittura, se sono violati i requisiti di applicabilità informa nell'output "`Warning message: Exp. counts < 5: Chi-squared approx. may be incorrect!!`" Però, **non gestisce i residui di cella**: può quindi essere utile per un primo screening, ma se il test *overall* risultasse significativo, potrebbe essere il caso di approfondire. Il grafico prodotto da `plotit= TRUE` (default) è un mosaic plot.

```
Desc(table(fumo$Fagerstrom_categorie, fumo$outcome_3_mesi))
-----------------------------------------------------------
table(fumo$Fagerstrom_categorie, fumo$outcome_3_mesi) (table)

Summary:
n: 126, rows: 2, columns: 2

Pearson's Chi-squared test (cont. adj): ← continuity adjustement - corr
 ezione di Yates
  X-squared = 6.912, df = 1, p-value = 0.008562
Fisher's exact test p-value = 0.007021
McNemar's chi-squared = 5.1948, df = 1, p-value = 0.02265

                    estimate lwr.ci upr.ci'

odds ratio             0.324  0.145  0.720
rel. risk (col1)       0.709  0.555  0.906     ← vedi sotto
rel. risk (col2)       2.190  1.227  3.907


Phi-Coefficient        0.251
Contingency Coeff.     0.244
Cramer's V             0.251

                          astinente   fumatore    Sum

alta dipendenza    freq          37         28     65
                   perc       29.4%      22.2%  51.6%   ← % sul totale N
                   p.row      56.9%      43.1%      .   ← % entro riga (livello di dipendenza)
                   p.col      43.0%      70.0%      .   ← % entro colonna (outcome)

bassa dipendenza   freq          49         12     61
                   perc       38.9%       9.5%  48.4%
                   p.row      80.3%      19.7%      .
                   p.col      57.0%      30.0%      .

Sum                freq          86         40    126
                   perc       68.3%      31.7% 100.0%
                   p.row         .          .      .
                   p.col         .          .      .
----------
' 95% conf. level
```



Il **rischio relativo** (`rel.risk`) è affine all'OR: negli studi epidemiologici **prospettici** esprime **l'incidenza**, cioè la proporzione di **nuovi casi in un gruppo di soggetti esposti a un fattore di rischio** rispetto alla proporzione di nuovi casi in un gruppo di soggetti non esposti.

		Sviluppo Malattia	
		SÌ	NO
Esposizione al rischio	SÌ	a	b
	NO	c	d

Quindi, considerando una tabella come quella a fianco, il rischio relativo è dato da: $RR = \frac{a/(a+b)}{c/(c+d)}$

I dati di `fumo` sono **retrospettivi**, **non** si può parlare di una reale **incidenza**; comunque, il calcolo è presto fatto:

	astinente	fumatore
alta dipendenza	37	28
bassa dipendenza	49	12

```
(37/(37+28))/(49/(49+12))
[1] 0.7086342
```

Il RR è pari a .709 per gli astinenti (colonna 1).

```
(28/(37+28))/(12/(49+12))
[1] 2.189744
```

Il RR è pari a 2.19 per i fumatori (colonna 2).

Nel caso di una **tabella $r \times c$**, nell'output **cambiano solo i test di significatività**: resta il test χ^2, e invece dei test di Fisher e di McNemar troviamo il **log – likelihood ratio test** (lo faremo nel capitolo 14, sulla regressione logistica: per ora non preoccupatevene) e il **Mantel – Heanszel χ^2 test**. Quest'ultimo è in realtà il **generalized Cochran- Mantel - Haenszel test**, utilizzabile per stimare l'associazione tra variabili anche in presenza di stratificazioni, confrontando gli OR degli strati: non fa comunque parte del nostro programma (per i curiosi, il riferimento è Mantel, 1963).

```
Desc(table(fumo$terapia, fumo$outcome_3_mesi))
-----------------------------------------------------
[omissis]
Pearson's Chi-squared test:
  X-squared = 1.9202, df = 2, p-value = 0.3829
Log likelihood ratio (G-test) test of independence:
  G = 1.8812, X-squared df = 2, p-value = 0.3904
Mantel-Haenszel Chi-squared:
  X-squared = 1.8471, df = 1, p-value = 0.1741

[omissis]
```

Se volete usare `Rcommander`, Una volta caricato il dataframe, le tabelle di contingenza sono gestite da `Statistiche→ tabelle di contingenza → tabella a due entrate`. Le variabili da inserire nella tabella sono indicate nella sezione "`Dati`":

La scelta del test da applicare alla tabella avviene nella sezione Statistiche: potete scegliere tra test χ^2 e test di Fisher, ma non c'è il test di McNemar, Potete visualizzare percentuali in base al marginale desiderato, ma non i residui standardizzati di cella. Manca anche un coefficiente di intensità dell'associazione, però, se, scegliete il test di Fisher, viene prodotto l'odds ratio. Insomma, non è un menu soddisfacente.

*Recuperate il dataframe **`gatti`**:*

a) verificate l'ipotesi che vivere con un gatto faciliti il riconoscimento delle intenzioni comunicative del micio, usando le diverse situazioni sperimentali (alla vista del cibo, rinchiuso nel trasportino, durante lo spazzolamento);

b) verificate il luogo comune che vede le zitelle come particolarmente amanti dei gatti.

Capitolo 8

Distribuzioni bivariate continue: correlazione e cograduazione

In questo capitolo useremo ancora il dataframe ***attaccamento****: riapritelo e, prima di proseguire con la lettura, eseguite:*

1. *create il dataframe* ***a*** *associandogli le caratteristiche di* ***attaccamento****, per risparmiare tempo e spazio; se volete provare il brivido del rischio,* ***salvatelo*** *come file delimitato da tabulazioni;*
2. *costruite la variabile* `a$burden_totale` *data dalla somma delle cinque sottoscale del CBI;*
3. *costruite la variabile* `a$QOL_totale` *data dalla somma delle quattro dimensioni della qualità della vita.*

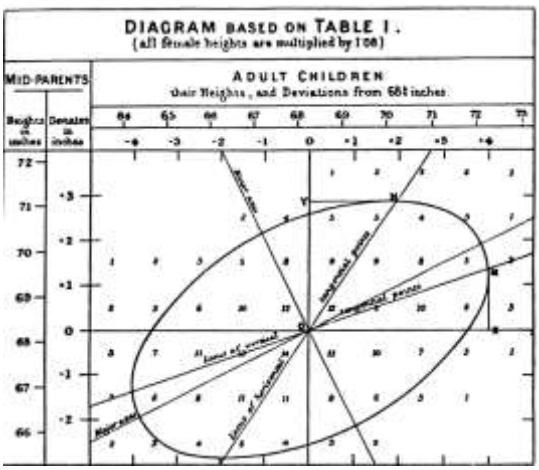
In questo capitolo vedremo come descrivere e quantificare la **relazione bivariata di ordine zero**[61], **lineare** o **monotonica**, esistente fra **due** distribuzioni, a seconda che siano **entrambe metriche** (§8.2.1: coefficiente r di Pearson) o che siano due distribuzioni continue di cui **almeno una ordinale** (§8.3.1 coefficiente rho ρ di Spearman e §8.4.1 coefficiente tau τ di Kendall).

La correlazione indica in che modo due [o più] variabili siano legate l'una all'altra da una relazione funzionale: consente di valutare quanto bene un'equazione di una retta o di una curva descriva la relazione tra le variabili.

8.1 Descrivere una distribuzione bivariata con variabili continue

Il grafico principale per descrivere una relazione tra due variabili continue è il grafico a dispersione o **scatterplot** *xy*: usato fin dal XIX secolo, resta, secondo Tofte (1983): "il più grande di tutti i grafici. Lega almeno due variabili, incoraggiando e addirittura implorando il lettore a valutare la possibile relazione causale[62] tra le variabili nel grafico". Infatti, date due variabili appaiate (due misure prese sullo stesso caso) X_1 e X_2, i punti che indicano i loro valori congiunti sono rappresentati graficamente in un sistema di coordinate cartesiane XY.

Questo a fianco è il ***primo scatterplot bivariato accreditato*** *in una pubblicazione scientifica: è di Galton (1885) e rappresenta la relazione tra l'altezza dei figli e l'altezza della media della coppia genitoriale. Ritroveremo Galton nel capitolo 9 sulla regressione lineare (e nell'Appendice I).*



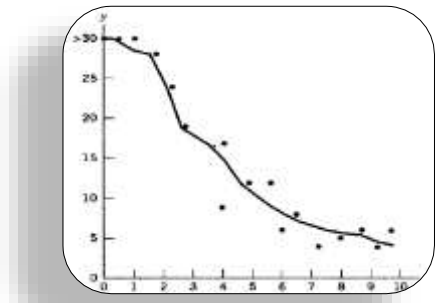
A seconda di come si dispongono i punti nel diagramma, si possono identificare diversi tipi di correlazione. Quando tutti i punti del diagramma sono concentrati attorno a una retta, la correlazione è detta **lineare**; la relazione lineare è un caso

[61] La relazione bivariata di ordine zero è quella tra due sole distribuzioni; alla fine del capitolo vedremo le **correlazioni di ordine uno**.

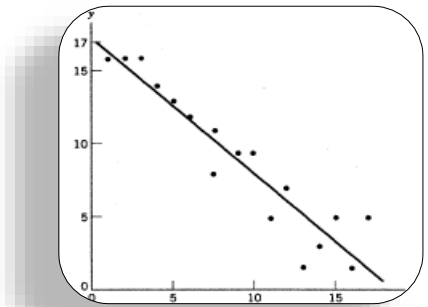
[62] Useremo lo scatterplot per verificare relazioni causa-effetto nel capitolo 9: in questo lo usiamo affrontando ipotesi di semplice co-variazione tra X e Y, in cui il ruolo di causa e il ruolo di effetto non sono attribuiti.

particolare di **relazione monotonica**, in cui **all'aumentare di X_1 i valori di X_2 variano**, di quantità **non costanti, in un'unica direzione** per tutti i valori di X_1: o aumentano, o diminuiscono. Ad esempio, X_1 può diminuire poco per valori bassi di X_1, molto per valori medi di X_1, poco per valori alti di X_1 (v. figura seguente). Nella relazione **lineare**, oltre a essere monotonica la direzione della variazione di X_1 al variare di X_1, **l'entità della variazione di X_2 è costante** per tutti i valori di X_1.

Quando al crescere di X_1 anche X_2 tende a crescere, la correlazione è detta **diretta** o **positiva**; se, al crescere di X_1, X_2 tende a diminuire, la correlazione è detta **inversa** o **negativa**. Se i valori delle variabili si dispongono nel diagramma **senza seguire alcun andamento lineare** preciso, esse non sono correlate – ovvero, sono **linearmente indipendenti**. Poiché le statistiche che applicheremo nel paragrafo §2.3 riguardano ipotesi su relazioni di tipo lineare, esse non saranno applicabili nel caso in cui il grafico a dispersione evidenzi una relazione non lineare



Relazione monotonica: correlazione negativa

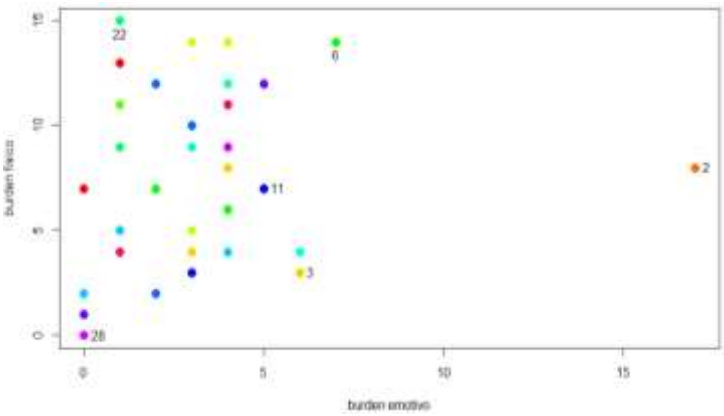


Relazione lineare: correlazione negativa

Nel prossimo capitolo impareremo come tracciare la retta per la relazione lineare

La funzione più semplice per uno scatterplot in R è l'onnipresente `plot(X1,X2)`, cui si possono aggiungere tutti i parametri grafici che abbiamo già più volte usato (etichette degli assi, titolo, grandezza, colore e tipo dei simboli, linea di riferimento, ecc.). Per esempio, vogliamo vedere se tra i caregiver esiste un andamento comune, di tipo lineare, tra lo stress emotivo e lo stress fisico dovuti all'assistenza; identifichiamo nel plot alcuni dei soggetti con `identify(X1,X2)`:

```
plot(a$CBI_burden_emotivo, a$CBI_burden_fis
  ico, pch=19, col=rainbow(15), cex=1.5, xla
  b="burden emotivo", ylab="burden fisico")

identify(a$CBI_burden_emotivo, a$CBI_burden
   _fisico)
[1]  2  3  6 11 22 28
```



La relazione **sembra esistere ed essere positiva**: i **caregiver con bassi valori nel burden emotivo tendono ad avere bassi valori anche nel carico fisico** (ad esempio il soggetto 28), se hanno valori medi in X_1 tendono ad averli medi anche in X_2 (ad esempio il soggetto 11), e **se hanno valori alti in X_1 tendono ad averli alti anche in X_2 (**ad esempio il soggetto 6). Ci sono tuttavia **eccezioni**: il soggetto 2 ha uno stress emotivo estremo, ma un carico fisico moderato, mentre il soggetto 22 concentra praticamente tutto lo stress nella componente fisica e quasi per nulla in quella emotiva. Come approfondiremo successivamente (capitolo 9), i caregiver 2 e 22 sono probabilmente **outlier bivariati**, cioè persone che hanno uno "strano" valore in X_2 rispetto al loro valore in X_1.

La funzione **`xyplot`** di `lattice` svolge le stesse funzioni di `plot`, ma con qualche caratteristica più interessante; per esempio, permette di vedere la relazione tra due variabili a seconda dell'appartenenza dei soggetti all'uno o all'altro

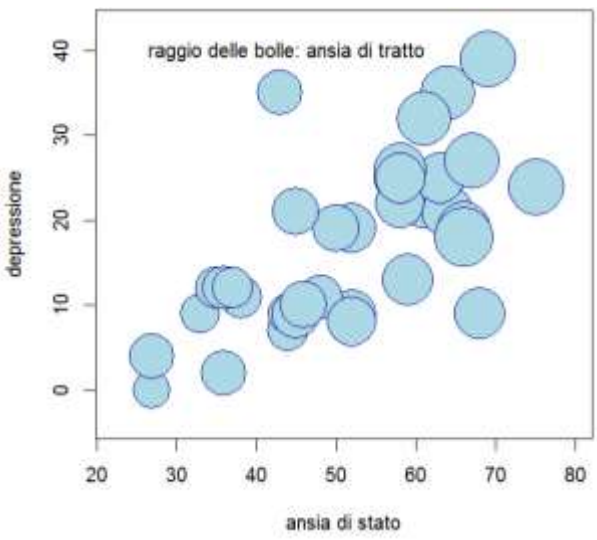
livello di una variabile `factor` (per esempio, in maschi e femmine, o nel gruppo di controllo e nel gruppo sperimentale). La scrittura degli argomenti è un po' diversa: **`xyplot(X1~X2)`** per la relazione tra le due variabili nel campione complessivo, **`xyplot(X1~X2 | fattore)`** per la correlazione tra le due variabili in ognuno dei livelli del fattore. Vedremo un esempio di `xyplot` nel prossimo paragrafo.

Un grafico più informativo, almeno in alcuni casi, è il **bubbleplot** o **grafico a bolle**: le "bolle" rappresentano le coordinate di una distribuzione bivariata X_1X_2; quindi, la loro **disposizione dà la stessa informazione di un grafico a dispersione**, ma, in più, la **loro dimensione non è costante**, bensì **è proporzionale al punteggio** che ogni soggetto ha in **una terza variabile X_3**, che è ragionevolmente correlata a X_1 e X_2 (vedremo le correlazioni parziali nel §8.4.1).
Per esempio, è intuitivo che **l'ansia di stato X_1**, legata a una situazione contingente, sia fortemente e positivamente correlata alla **depressione** X_2: tuttavia, è molto probabile che la predisposizione a percepire ansia (**ansia di tratto**, X_3) sia legata a entrambe le dimensioni. Quindi, chi ha un basso punteggio di ansia di tratto (bolla piccola) dovrebbe manifestare sia bassa ansia di stato (X_1) sia leggera depressione (X_1): le bolle più piccole dovrebbero trovarsi in prevalenza nella parte inferiore del bubbleplot. Al crescere del punteggio di ansia di tratto (bolla sempre più grande), dovremmo trovare maggiore ansia di stato (X_1) e depressione (X_2): le bolle più grandi dovrebbero prevalere nella parte superiore del grafico.
Nel dataframe a abbiamo queste tre variabili: mettiamo alla prova l'ipotesi, usando la funzione `symbols(x1, x2, circles= raggio delle bolle)`, che disegna cerchietti (`circles`) in corrispondenza delle coordinate X_1 e X_2. Nell'argomento `circles=` va specificata quale sia la variabile che determina il raggio delle bolle: possiamo crearla prima e inserirla come oggetto di `circles`, ricordandoci che, conoscendo l'area di un cerchio (il punteggio di X_3), possiamo ricavarne il raggio con la formula $r = \sqrt{\frac{A}{\pi}}$:

```
raggio<-sqrt(a$STAI_tratto/pi)
```

`bg=` e `fg=` gestiscono rispettivamente il colore entro la bolla e quello del bordo. Di default, la dimensione dei `symbols` assegnata alla bolla più grande è `inches=1`: le altre sono scalate in proporzione a X_3. La dimensione di default è spesso piuttosto invadente e rende difficile leggere il grafico: si può ridurre con `inches= .5` o `=.25`, eccetera.

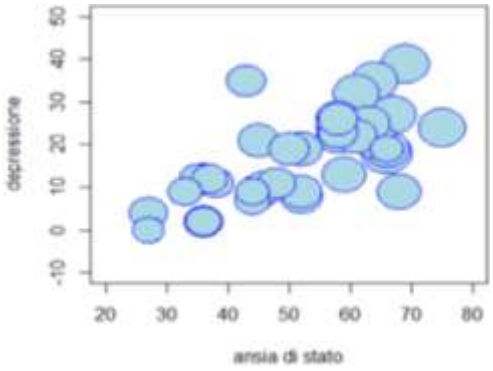
```
symbols(a$STAI_stato, a$BDI_II_depressione, circles
  = raggio, inches = .25, fg = "dark blue", bg =
  "light blue", xlab="ansia di stato",
  ylab="depressione")
text(x = 25, y = 40, labels = "raggio delle bolle:
  ansia di tratto", pos = 4)
```



Effettivamente, la **relazione tra ansia di stato e depressione è evidente e positiva;** inoltre, le **bolle più piccole si trovano in corrispondenza delle coordinate X_1X_2 inferiori**, e, al crescere della distribuzione bivariata, si fanno via via più grandi: **chi ha più depressione e più ansia di stato tende anche ad avere più ansia di tratto.**

Ancora più semplice è `PlotBubble(x, y, area= variabile in funzione della quale cambia il raggio delle bolle)` del solito `DescTools`.

```
PlotBubble(x=a$STAI_stato, y = $BDI_II_depressione,
 area=a$STAI_tratto, col = "light blue",
 border = "blue", xlab="ansia di stato",
 ylab="depressione")
```



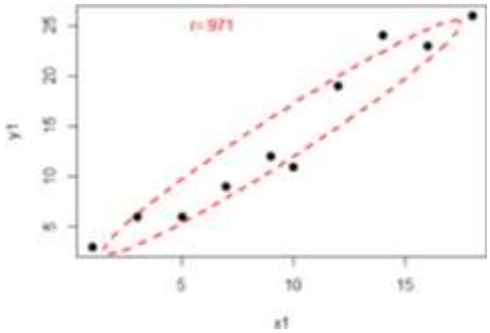
8.2 Quantificare e verificare ipotesi su una distribuzione bivariata con variabili continue

Nel caso di due variabili continue, H_0 è che le due variabili siano tra loro **linearmente indipendenti**, ovvero non abbiano **andamento comune** (se sono ordinali) o varianza in comune (comunanza o **covarianza**, se sono metriche): al variare dell'una, l'altra variabile varia in maniera linearmente **indipendente**. Invece, H_1 è che le due variabili siano tra loro **interdipendenti**, ovvero che abbiano andamento comune o covarianza:

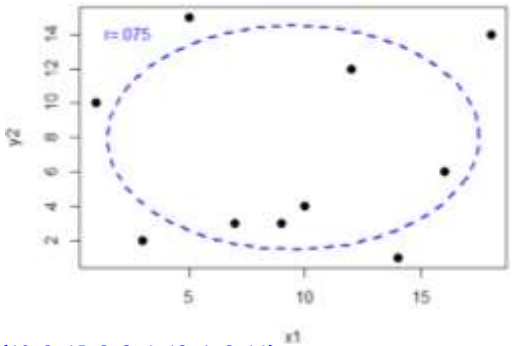
- $H_0 = \rho_{x1x2} = 0$
- $H_1 = \rho_{x1x2} \neq 0$

***Attenzione ai simboli**: poiché H_0 si riferisce al comportamento due variabili in popolazione, la correlazione è indicata con il simbolo greco ρ corrispondente alla lettera latina "r": **non** si tratta quindi del coefficiente di correlazione ordinale rho di Spearman, che vedremo nel §8.3.2.*

Abbiamo visto nello scatterplot che la disposizione dei punti informa sulla **direzione** della relazione; in realtà, informa anche sulla **intensità** della relazione, indicata **dall'ampiezza dell'ellisse** formata dai punti: tanto più l'ellisse è ristretta e molto allungata, tanto più alta è la correlazione; l'ellisse più ristretta di tutte, ovvero una linea, indica la massima correlazione (positiva o negativa); al contrario, un'ellisse poco allungata indica una scarsa correlazione, e la meno allungata di tutte, ovvero un cerchio, indica una correlazione lineare assente. **L'ampiezza di questa ellisse,** ovvero la **dispersione dei punti attorno all'immaginaria retta** che indicherebbe una correlazione **perfetta**, è quantificata dal **coefficiente *r* di Pearson**.

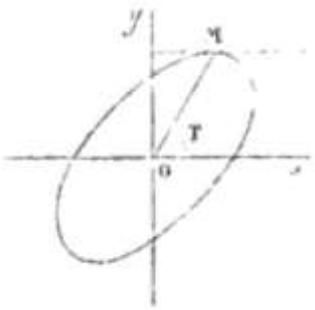



```
x1<-c(1,3,5,7,9,10,12,14,16,18)
y1<-c(3,6,6,9,12,11,19,24,23,26)
plot(pch=19, cex=1.2, x1, y1)
DrawEllipse[63](x = mean(x1), y=mean(y1), lwd=2,radius.x = 14,
radius.y = 1.5, rot = .971, border = "red", lty = 2, col = NA)
```




```
y2<-c(10,2,15,3,3,4,12,1,6,14)
plot(pch=19, cex=1.2, x1, y2)
DrawEllipse(x = mean(x1), y=mean(y2)+1,lwd=2, radius.x =
8, radius.y = 6.5, rot = .01, border = "blue", lty = 2,
col = NA)
```

[63] `DrawEllipse` è una funzione di `DescTools`; `x=` e `y=` indicano le coordinate del centro dell'ellisse, `radius.x` e `radius.y` rispettivamente la lunghezza del semiasse maggiore e del semiasse minore (in pollici), `rot` l'angolo di rotazione in radianti (§8.5).



*Questa è la prima ellisse (Bravais, 1844) associabile alla correlazione; nello stesso testo appaiono anche quella che Galton chiamerà retta di regressione, con relativa equazione, e il termine "corrélation" (pag. 9). In realtà, Bravais stava lavorando sulla dimostrazione dell'**indipendenza degli errori** associati alle variabili manifeste, tant'è che, sconfessando un suo precedente apprezzamento (1895), lo stesso Pearson affermerà (1920) che il vero merito deve essere attribuito a Galton. Ciononostante, il coefficiente r è spesso citato come **coefficiente di Bravais – Pearson**.*


ANALYSE MATHÉMATIQUE.

ERREURS DE SITUATION D'UN POINT.

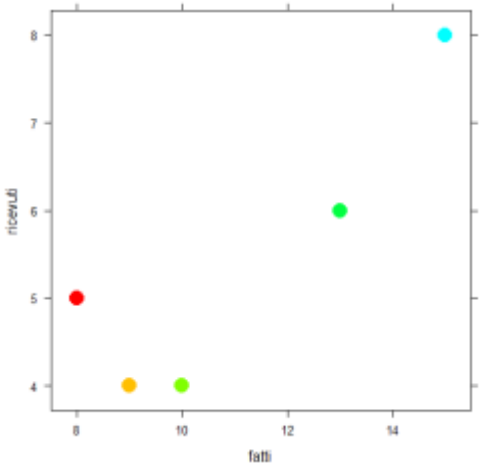
PAR A. BRAVAIS.

8.2.1 Variabili metriche: il coefficiente di correlazione di Pearson

Per sapere se due variabili sono correlate, ovvero se **covariano,** dobbiamo quantificare la loro **covarianza**. Abbiamo detto nel capitolo 3 che la **varianza** di una distribuzione rappresenta la **quantità media di variabilità dei dati attorno alla media**. Se due variabili X_1 e X_2 sono correlate, ci aspettiamo che il **modo** in cui i soggetti **deviano dalla media in X_1 sia coerente con il modo in cui gli stessi soggetti deviano dalla media in X_2**.

Per fare un esempio, proviamo a demolire il mito dello spirito natalizio: è vero che la generosità verso gli altri non è disinteressata, ma è direttamente proporzionale a quella ricevuta? Contattiamo cinque persone **a caso** e chiediamo quanti regali hanno fatto, e quanti ne hanno ricevuti, lo scorso Natale:

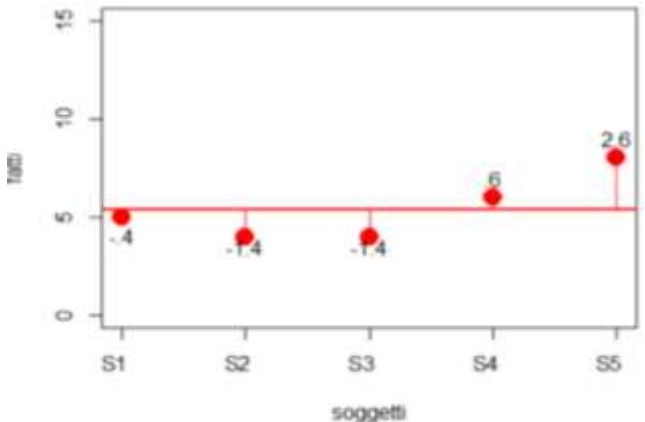
```
soggetti<-c(1,2,3,4,5)
regali_fatti<-c(5,4,4,6,8)
regali_ricevuti<-c(8,9,10,13,15)
mean(regali_fatti);sd(regali_fatti)
[1] 5.4
[1] 1.67332
mean(regali_ricevuti);sd(regali_ricevuti)
[1] 11
[1] 2.915476

xyplot(regali_fatti~regali_ricevuti, xlab="fatti",
   ylab="ricevuti", col= rainbow(15), pch=19, cex=1.5)
```



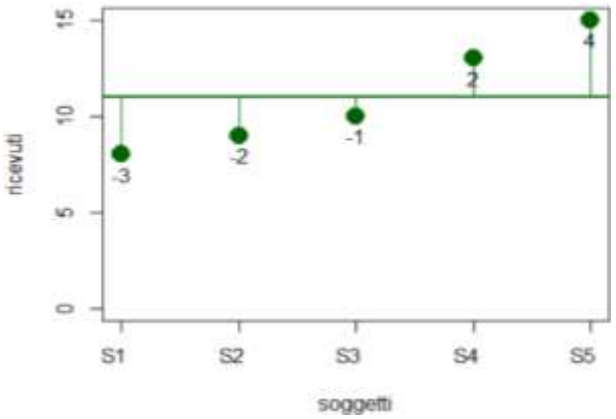
Diremmo che abbiamo trovato persone con amici più generosi di loro, e che la variabilità nella generosità altrui è molto più marcata di quella dimostrata dai nostri soggetti. Dal grafico, sembra di poter leggere una relazione positiva: chi fa pochi regali tende a riceverne pochi; chi ne fa molti, tende a riceverne molti.

Ora visualizziamo in un altro modo, più inusuale, lo stesso dato: mettiamo in ascissa i soggetti, e per ciascuno di loro indichiamo nel plot i regali fatti (in rosso) e quelli ricevuti (in verde). Nei grafici visualizziamo anche gli **scarti dalla media** di ogni soggetto con `segments` e `abline`.

```
(scarti_fatti<-regali_fatti-mean(regali_fatti))
 [1] -0.4 -1.4 -1.4  0.6  2.6
(scarti_ricevuti<-regali_ricevuti-mean(regali_ricevuti))
 [1] -3 -2 -1  2  4
```




```
plot(soggetti,regali_fatti, col="red", pch=19,
  cex=2, ylim=c(0,15), xlab="soggetti",
  ylab="ricevuti")
abline(h = mean(regali_fatti), lwd=2, col="red")
segments (soggetti, regali_fatti, soggetti,
  mean(regali_fatti), col="red", lwd=1.5)
mtext(text = "S",side = 1,at = c(.90, 1.90, 2.90,
  3.90, 4.90),line = 1)
text(x = c(1,2,3,3.8,4.8), y = c(4,3.5,3.5,7,8.5),
  labels = scarti_fatti, pos = 4)
```




```
plot(soggetti,regali_ricevuti, col="dark green",
  pch=19, cex=2, ylim=c(0,15), xlab="soggetti",
  ylab="ricevuti")
abline(h = mean(regali_ricevuti), lwd=2, col="dark
  green")
segments (soggetti, regali_ricevuti, soggetti,
  mean(regali_ricevuti), col="dark green", lwd=1.5)
mtext(text = "S",side = 1,at = c(.90, 1.90, 2.90,
  3.90, 4.90),line = 1)
text(x = c(1,2,3,3.8,4.8), y = c(7,8,9,13,14),
  labels = scarti_ricevuti, pos = 4)
```

Il modo in cui i soggetti **deviano** dalla propria media in ciascuna distribuzione è **coerente**: i primi tre sono sotto la media in entrambe le distribuzioni, gli ultimi due sono sopra la media in entrambe. Notate, comunque, che l'entità delle distanze entro ogni soggetto non è perfettamente proporzionale: ad esempio, il soggetto 1 è appena sotto nei regali fatti, ma molto sotto nei regali ricevuti. Quindi, il **pattern delle deviazioni comuni** sembra promettente rispetto all'esistenza di una relazione, ma occorre **sintetizzarlo e quantificarlo**. Sappiamo che non possiamo semplicemente sommare gli scarti semplici nelle due distribuzioni, dato che $\sum_{scarti} = 0$. Per la devianza di una distribuzione avevamo ovviato al problema elevando al quadrato gli scarti, ma nel caso della devianza comune a due distribuzioni abbiamo un'altra possibilità, ben più informativa: **moltiplichiamo ogni scarto della distribuzione X_{i1} per il corrispettivo scarto della distribuzione X_{i2}**, e successivamente sommiamo tutti questi prodotti.

Così, gli scarti **coerenti** (entrambi sotto la media o entrambi sopra la media) daranno origine a prodotti **positivi**; gli scarti **incoerenti** (sotto la media in X_1 e sopra la media in X_2, o viceversa) daranno origine a prodotti **negativi**. Perciò, se ci sarà una **preponderanza di scarti coerenti**, la **somma** dei prodotti sarà **positiva**; se ci sarà una **preponderanza di scarti incoerenti**, la **somma** dei prodotti sarà **negativa**. Se ci sarà **un'uguale proporzione di scarti coerenti e scarti incoerenti**, la somma dei prodotti **tenderà a zero**.

La somma dei prodotti così calcolati si chiama **codevianza:**

$$cod = \sum (x_{i1} - \bar{X}_1)(x_{i2} - \bar{X}_2)$$

Nel nostro esempio tutti gli scarti dei cinque soggetti sono coerenti, quindi avremo certamente una codevianza positiva:

```
(codevianza_regali<-sum(scarti_fatti*scarti_ricevuti))
[1] 17
```

Abbiamo già visto trattando il modello-media che la devianza (e così la codevianza) ha un difetto: più osservazioni aggiungiamo, più si gonfia. Questo fa sì che campioni con poche osservazioni tutte coerenti possono avere codevianze più piccole di quelle osservate in campioni con molte osservazioni, ma non tutte coerenti.

Nuovamente, **ponderiamo la codevianza per il numero di osservazioni meno 1**, ovvero per i df in popolazione, ottenendo la **covarianza:**

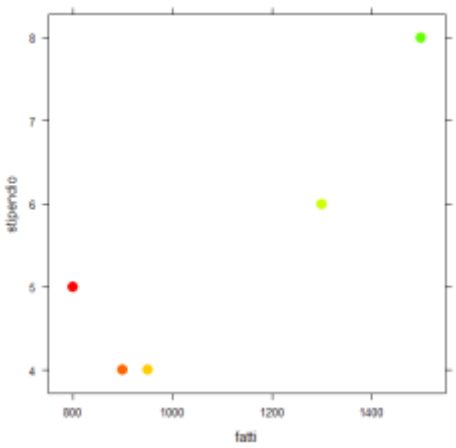
$$cov_{X_1X_2} = \frac{\sum (x_{i1} - \bar{X}_1)(x_{i2} - \bar{X}_2)}{N - 1}$$

```
(covarianza_regali<-codevianza_regali/(length(soggetti)-1))
[1] 4.25
```

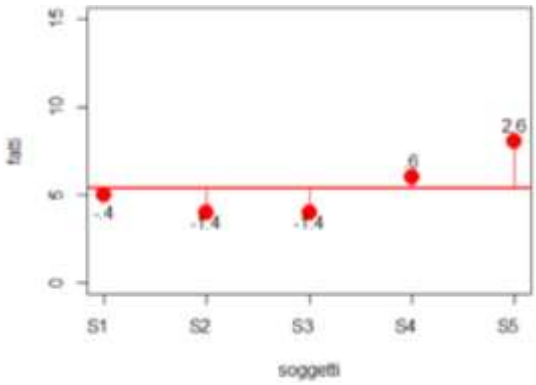
Ora sappiamo che la relazione tra regali fatti e ricevuti è positiva (più ne fai più ne ricevi, o più ne ricevi più ne fai), e quantificabile con una $covarianza = 4.25$, ma non sappiamo come interpretare cosa esprima il valore 4.25: la relazione tra le due variabili è trascurabile, debole, modesta, forte? La **covarianza dipende dall'unità di misura** utilizzata, il che pone due problemi: **non ha un parametro assoluto di riferimento** e rende **impossibile confrontare covarianze tratte da distribuzioni bivariate con unità di misura differenti**.

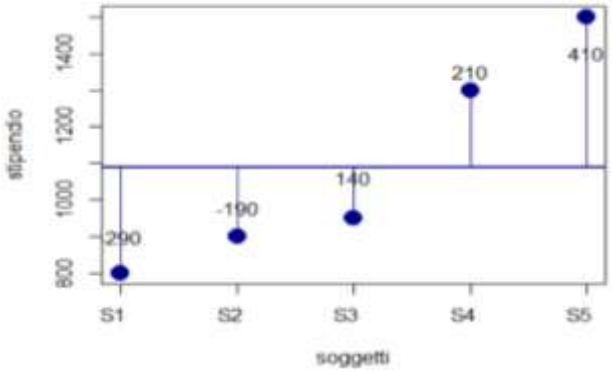
Per esempio, un altro ricercatore potrebbe affermare che il numero di regali fatti è molto più legato alle oggettive disponibilità economiche che ai regali ricevuti. Per dimostrarlo, chiede agli stessi **soggetti l'importo (approssimato) delle entrate mensili** in euro:

```
soggetti<-c(1,2,3,4,5)
stipendio<-c(800,900,950,1300,1500)
mean(stipendio);sd(stipendio)
[1] 1090
[1] 296.6479
xyplot(regali_fatti~stipendio, xlab="fatti",
   ylab="stipendio", col= rainbow(15), pch=19, cex=1.5)
```



Anche in questo caso, la relazione è positiva: chi fa pochi regali tende ad avere un basso stipendio; chi ne fa molti, ha uno stipendio più alto.





La distribuzione degli scarti nelle due variabili sembra molto coerente; quantifichiamola in **covarianza** con **`cov(x1, x2)`**:

```
cov(regali_fatti,stipendio)
[1] 442.5
```

La covarianza tra regali fatti e regali ricevuti era pari a **4.25**, quindi la relazione tra regali e stipendio sembrerebbe 100 volte più forte! Ma se avessimo espresso lo stipendio in **migliaia di euro**, avremmo avuto un'impressione opposta:

```
stipendio_migliaia<-c(.8,.9,.95,1.3,1.5)
mean(stipendio_migliaia);sd(stipendio_migliaia)
[1] 1.09
[1] 0.2966479
cov(regali_fatti, stipendio_migliaia)
[1] 0.4425
```

La soluzione è semplice: per confrontare "cose" prescindendo dalla loro unità di misura, è sufficiente **standardizzarle**, rapportandole così a un'unica unità di misura. L'unità di misura standard che abbiamo abbondantemente usato nel caso di una sola distribuzione è la deviazione standard: perciò, potremmo **standardizzare la covarianza, esprimendola in unità di deviazioni standard dalla media**, e risolvere il problema. Naturalmente, questa volta abbiamo **due deviazioni standard**, una per distribuzione; quindi, al denominatore della covarianza avremo s_{X_1} e s_{X_2}, sotto forma di prodotto $s_{X_1}s_{X_2}$.

Una distribuzione: punteggio standardizzato $z_{xi1} = \frac{x_{i1} - \bar{X}_1}{s_{x1}}$

Due distribuzioni: covarianza standardizzata $r_{(x_1,x_2)} = \frac{cov_{(x_1,x_2)}}{s_1 s_2}$

Il rapporto tra covarianza e prodotto delle due deviazioni standard, ovvero la covarianza standardizzata, è il famoso **coefficiente di correlazione r** *Errore. Il segnalibro non è definito.***di (Karl) Pearson**. Pearson non ha "inventato" la correlazione: ha tradotto in formula matematica la formulazione teorica di "correlazione bivariata normale" avanzata da Galton (1885 e 1888[64], che però non comprende il concetto di correlazione negativa; §8.2.2), il quale a sua volta riprende lavori precursori di Gauss (1823) e Bravais (1846).

Applichiamola a `regali_fatti` e `stipendio`:

```
cov(regali_fatti, stipendio)/sd((regali_fatti)*sd(stipendio))
[1] 0.8914417
```

Per la precisione, la formula originaria di Pearson (1895) era basata sul rapporto tra la codevianza e la media geometrica delle due varianze di X e Y:

$$r = \frac{\sum(x_{i1} - \bar{X}_1)(x_{21} - \bar{x}_2)}{\sqrt{(x_{i1} - \bar{X}_1)^2}\sqrt{(x_{21} - \bar{X}_2)^2}}$$

Al **numeratore** le variabili **X e Y sono centrate sulla rispettiva media** prima di calcolare la somma dei loro prodotti, mentre il **denominatore** corregge le scale delle variabili perché si esprimano con **uguale unità di misura**. Nel nostro caso:

```
numeratore<-sum((regali_fatti-mean(regali_fatti))*(stipendio-mean(stipendio)))
denominatore<-sqrt(sum((regali_fatti-mean(regali_fatti))^2)*sum((stipendio-mean(stipendio))^2))
numeratore/denominatore
[1] 0.8914417
```

Dalla formula precedente è facile capire che r può essere espresso (e ricavato) anche[65] come **media dei prodotti delle distribuzioni standardizzate z_{X_1} e z_{X_2}**: basta dividere il numeratore e il denominatore della formula di Pearson per il prodotto delle due deviazioni standard.

$$r = \frac{\sum z_{x1} z_{x2}}{N}$$

Dato che la **media** di una distribuzione (nel nostro caso bivariata) è il suo **primo momento[66]**, ecco perché il coefficiente r è chiamato **coefficiente prodotto – momento**.

Attenzione, però: per usare questa formula, le variabili vanno standardizzate usando la **deviazione standard campionaria**, ricavata dalla varianza **campionaria**, ovvero la devianza **divisa per N**, e non per i $df = N - 1$. Se invece si usa la devianza divisa per i df, il numeratore di r andrà diviso per $N - 1$. Volendo usare la funzione `scale`, che calcola la deviazione standard usando i gradi di libertà, dovremmo cambiare il denominatore di r in $N - 1$:

```
zfatti<-(scale(regali_fatti))
zstipendio<-(scale(stipendio))
sum(zfatti*zstipendio)/(5-1)
[1] 0.8914417
```

Altrimenti, se calcoliamo la deviazione standard dividendo la devianza per N, useremo N al denominatore di r:

```
zfatti2<-(regali_fatti-mean(regali_fatti))/sqrt(sum((regali_fatti-mean(regali_fatti))^2/5))
zricevuti2<-(stipendio-mean(stipendio))/sqrt(sum((stipendio-mean(stipendio))^2/5))
sum(zfatti2*zricevuti2)/5
[1] 0.8914417
```

[64] "*Two variable organs are said to be co-related when the variation of one is accompanied on the average by more or less variation of the other, and in the same direction*", pag. 135.

[65] Rodgers e Nicewander (1988) elencano ben tredici diversi modi di concettualizzare e ottenere *r* dai dati; ne vedremo solo alcuni, in questo capitolo e nei successivi.

[66] Sono definiti **momenti** diversi **indici statistici** di una distribuzione quantitativa aleatoria (continua o categoriale): la **media** è il momento centrale di **ordine 1**, la **varianza** è il momento di **ordine 2**, **l'asimmetria** il momento di **ordine 3**, la **curtosi** il momento di **ordine 4**…

Oltre ad aver risolto il problema dell'unità di misura, r risolve anche quello legato al parametro di riferimento assoluto: è dimostrabile[67] che la **covarianza**, in valore assoluto, al numeratore **non può essere maggiore del prodotto delle deviazioni standard** al denominatore. Perciò, il rapporto tra la covarianza e il prodotto delle deviazioni standard, ovvero r, è per forza **compreso tra -1 e +1**.

La perfetta correlazione positiva $r = 1$ indica **identità** tra i valori **standardizzati / centrati appaiati** (Caham, 1987): per questo motivo, r informa sulla **prossimità all'identità** (*closeness to identity*) **di z_x e z_y**. Quindi, r può essere interpretato come una misura di **goodness of fit delle variabili standardizzate alla retta di identità** tra le variabili (standardizzate) stesse, cioè una misura di **somiglianza tra i valori centrati**. Vediamo con un esempio: le variabili A e B hanno punteggi grezzi diversi, ma punteggi z identici, e la correlazione tra loro è perfetta. Costruiamo lo scatterplot di z_A e z_B e tracciamo la retta che esprime l'identità tra i punti in z_A e in z_B con `abline(a=0, b=1)`: la distribuzione bivariata si dispone perfettamente sulla retta di identità.

```
A<-c(49,40,48,41,45,54)
B<-c(39,30,38, 31,35,44)
zA<-round(scale(A),3)
zB<-round(scale(B),3)
```

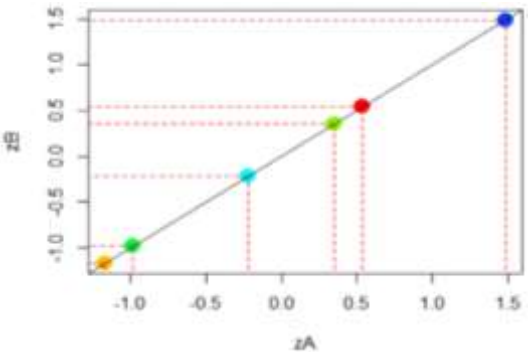
→

```
cbind(zA, zB)
       [,1]   [,2]
[1,]  0.538  0.538
[2,] -1.170 -1.170
[3,]  0.348  0.348
[4,] -0.981 -0.981
[5,] -0.221 -0.221
[6,]  1.487  1.487
```

→

```
cor(A, B)
[1] 1
```

```
plot(zA, zB, pch=19, cex= 1.5, col= rainbow(6)); abline(0,1)
segments(x0 = zA, y0= -2, y1 = zB, col= rainbow(6), lty= 2)
segments(x0 = -2, y0= zB, x1 = zA, col= rainbow(6), lty= 2)
```



Due variabili C e D, ben correlate ma i cui punteggi standardizzati non sono identici, hanno una correlazione inferiore a 1 e la distribuzione bivariata non si sovrappone perfettamente alla retta di identità tra z_C e z_D: la loro *closeness to identity* è inferiore.

```
C<-c(49,40,48,41,45,54)
D<-c(40,30,31,31,36,42)
zC<-round(scale(C),3)
zD<-round(scale(D),3)
```

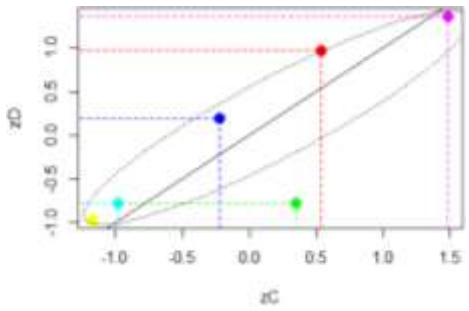
→

```
cbind(zC, zD)
       [,1]   [,2]
[1,]  0.538  0.973
[2,] -1.170 -0.973
[3,]  0.348 -0.778
[4,] -0.981 -0.778
[5,] -0.221  0.195
[6,]  1.487  1.362
```

→

```
cor(C,D)
[1] .827
```

```
plot(zC, zD, pch=19, cex= 1.5, col= rainbow(6)); abline(0,1)
segments(x0 = zC, y0= -2, y1 = zD, col= rainbow(6), lty= 2)
segments(x0 = -2, y0= zD, x1 = zC, col= rainbow(6), lty= 2)
DrawEllipse (x = mean(zC)+.2, y=mean(zD)+.2, lwd=1.5,radius.x = 1.85,
radius.y = .4, rot = .70, border = "black", lty = 3, col = NA)
```



r è quindi un **coefficiente di effect size**: convenzionalmente, se $r < |.2|$ la relazione è trascurabile, fino a $r = |.5|$ è debole-moderata, se $r > |.5|$ sempre più forte. Tuttavia, un'informazione molto più obiettiva e precisa sull'intensità della relazione si ottiene dal **coefficiente r al quadrato**, che prende il nome di **coefficiente di determinazione R^2**: **moltiplicato per 100**, indica la **percentuale di varianza comune / condivisa tra le due variabili**: ecco finalmente un'indicazione **esatta** di quanto forte sia la relazione tra le due variabili.

Concludiamo calcolando le altre correlazioni nel nostro esempio:

```
cov(regali_fatti, regali_ricevuti)/(sd(regali_fatti)*sd(regali_ricevuti))
[1] 0.8711651
cov(regali_fatti, stipendio_migliaia)/(sd(regali_fatti)*sd(stipendio_migliaia))
[1] 0.8914417
```

[67] Disuguaglianza di Cauchy-Schwarz, che stabilisce un limite superiore a un prodotto vettoriale: $(\sum_{i=1}^{n} x_{1i}^2)(\sum_{i=1}^{n} x_{2i}^2) \geq (\sum_{i=1}^{n} x_{1i}x_{2i})^2$, applicabile anche a variabili scarto come le nostre $[\sum_{i=1}^{n}(x_{1i} - \mu_{x1})\sum_{i=1}^{n}(x_{2i} - \mu_{x2})]^2 \geq \sum_{i=1}^{n}(x_{1i} - \mu_{x1})^2 \sum_{i=1}^{n}(x_{2i} - \mu_{x2})^2$.

Anche se di poco, la relazione tra stipendio e regali fatti è un po' più forte della relazione tra regali fatti e ricevuti. Più precisamente, regali fatti e ricevuti condividono il 76% della varianza, mentre regali fatti e stipendio condividono il 79.5% della varianza:

```
round(0.8711651^2,3); round(0.8914417^2,3)
[1] 0.759
[1] 0.795
```

In R, conoscere il coefficiente di correlazione bivariata è facilissimo. La funzione più semplice è **`cor(x1,x2,method="tipo di coefficiente di correlazione")`**. L'argomento `method=` specifica se si vuole il coefficiente di correlazione di Pearson, di Spearman o di Kendall (§8.3.2); l'opzione di default è "`pearson`", quindi possiamo ometterla per richiedere:

```
cor(regali_fatti,regali_ricevuti)
[1] 0.8711651
```

La funzione `cor` ci informa su direzione e intensità della relazione nel campione, ma non è affatto informativa su come sia stimata la **correlazione in popolazione**: ci serve il CI, e potrebbe non dare fastidio anche associare un *p*-value al coefficiente di correlazione per stimare la probabilità della relazione riscontrata nel campione, data per vera l'ipotesi nulla di assenza di relazione lineare in popolazione.

A questo scopo, dato che non segue una distribuzione di probabilità nota, r può essere espresso come un **quantile *z*** di una distribuzione normale standardizzata, ma più frequentemente si trasforma in un **quantile *t* di una distribuzione *t* con** $df = N - 2$.

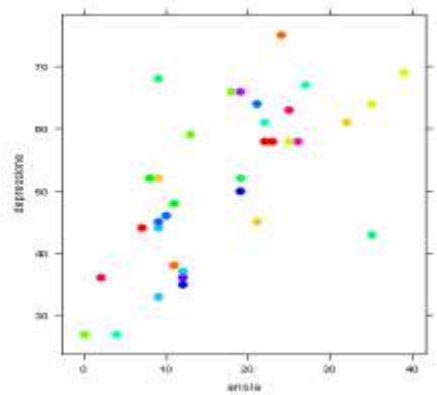
$$t_{N-2} = \frac{r\sqrt{N-2}}{\sqrt{1-r^2}}$$

La funzione di R che fornisce, oltre alla statistica campionaria, anche CI e $p - value$ del quantile t è **`cor.test`**`(x1, x2, method= "tipo di coefficiente di correlazione")`; oltre agli argomenti già visti in `cor`, `cor.test` consente di cambiare l'intervallo del CI (di default al 95%) con `conf.level=` , nonché di specificare la direzione di H, che è come sempre bidirezionale per default, con `alternative=` "`two.sided`" / "`greater`" / "`less`", dove `greater`: $H_1: r > 0$ e `less`: $H_1: r < 0$. Notate che possiamo indicare le variabili da correlare sia nella forma: `(dataframe$variabile1, dataframe$varabile2)`, sia usando la **formula**: **`(~variabile1 + variabile2, data= dataframe)`**.

Vediamo l'output di `cor.test` applicandolo ai dati veri del dataframe `attaccamento` [dovreste già averlo rinominato in **a**, e aver calcolato i due nuovi totali come richiesto a inizio capitolo].

È noto in letteratura che ansia e depressione siano dimensioni fortemente correlate, talvolta inestricabili. Vediamo se è così anche nel campione dei caregiver, mettendo in relazione ansia di stato (`a$STAI_stato`) e depressione (`a$BDI_II_depressione`): in entrambi i test, a punteggio maggiore corrisponde una sintomatologia più grave, perciò ci aspettiamo una relazione positiva. Esercitiamoci con `xyplot`: ricordate di caricare `lattice`.

```
xyplot(STAI_stato~BDI_II_depressione, data= a, pch=19,
   col=rainbow(15), xlab="ansia", ylab="depressione", cex=1.5)
cor.test(a$STAI_stato,a$BDI_II_depressione)
 Pearson's product-moment correlation
data:  a$STAI_stato and a$BDI_II_depressione
t = 5.6447, df = 38, p-value = 1.749e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4607521 0.8153205
sample estimates:
     cor
0.675332
```



Dallo scatterplot sembra abbastanza evidente una relazione positiva tra ansia e depressione nel campione: i caregiver con poca ansia tendono ad avere anche poca depressione, e viceversa. L'output di `cor.test` conferma che la relazione nel campione è positiva e di discreta entità ($r = .676$); il CI in popolazione non comprende il valore previsto da H_0: $\rho = 0$: quindi, la **relazione è reale** e non casuale anche in popolazione. Il $p - value$, in maniera ridondante, ribadisce che sarebbe un evento estremamente raro riscontrare in popolazione questo pattern di relazione nei dati, assumendo che H_0 sia vera. Però, il CI è piuttosto ampio: in popolazione la forza della relazione attesa varia da abbastanza debole (.461) a piuttosto forte (.815). Quindi, il nostro campione supporta la teoria che ansia e depressione siano dimensioni interconnesse, ma non ci consente molta precisione nel determinare la forza di questa connessione nella popolazione. Quantifichiamo la varianza comune ad ansia e depressione nel campione con il coefficiente R^2:

```
0.675332 ^2
[1] 0.4560733
```

Potremmo anche, con l'eleganza che ormai dovrebbe contraddistinguere i nostri compiti in R, costruire l'oggetto-test e usarne gli elementi della lista, come abbiamo fatto per `t.test` e `chisq.test`. L'elemento che ci serve è `$estimate`:

```
ansia_depressione<-cor.test(a$STAI_stato,a$BDI_II_depressione)
ansia_depressione$estimate^2
      cor
0.4560733
```

La costruzione del CI attorno a r, che R ci fornisce liberandoci dall'onere del calcolo, è più complicata di quanto potrebbe sembrare. Anche in questo caso per stimare il CI è necessario che i dati della distribuzione campionaria di r siano **simmetrici** rispetto al valore ρ in popolazione, ma, a differenza di quanto accade alle medie della DCM rispetto a μ, i valori della DC degli r sono **distribuiti normalmente** attorno a ρ **solo** quando $\boldsymbol{r}$ **tende a 0** ed i **campioni sono molto ampi** (tendenti a infinito). Quindi, pur conoscendo r e il suo errore standard, **non potremmo** procedere in analogia a quanto faremmo per una media: ~~$CI_{\rho} = r \pm t_{n-2,\alpha/2} \times SE_r$~~. Niente paura: sono stati proposti ben **quattro** metodi matematici e un bel po' di metodi grafici. R utilizza la trasformazione di r in z_r (metodo di **Fisher**: è una trasformazione logaritmica), ma possiamo non approfondire ulteriormente le sgradevolezze del calcolo[68].

Verificate la relazione tra ansia di stato e depressione, e tra ansia di stato e ansia di tratto: quali conclusioni potremmo trarne?

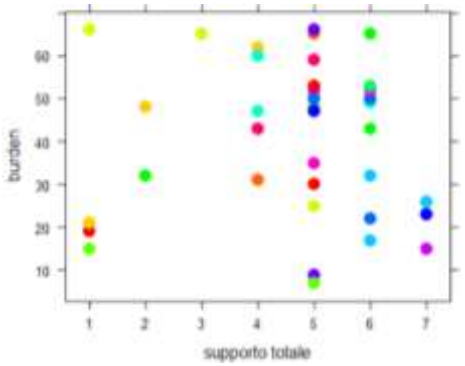
Calcolate il coefficiente della correlazione tra burden emotivo e burden fisico che abbiamo rappresentato graficamente. Quale conclusione ne dovremmo trarre?

Rifate la correlazione tra le due variabili dopo aver eliminato i due soggetti che avevamo definiti "anomali" leggendo il grafico. È cambiato qualcosa? Perché?

È abbastanza intuitivo che il carico dell'assistenza possa trarre beneficio dall'essere supportati dagli altri. Nella ricerca, il supporto sociale è stato valutato con la scala di Zimet: se l'intuizione fosse corretta, dovremmo trovare una relazione negativa tra burden e supporto (un minor burden corrisponde a un maggior supporto, e viceversa). Verifichiamo usando i totali delle due scale:

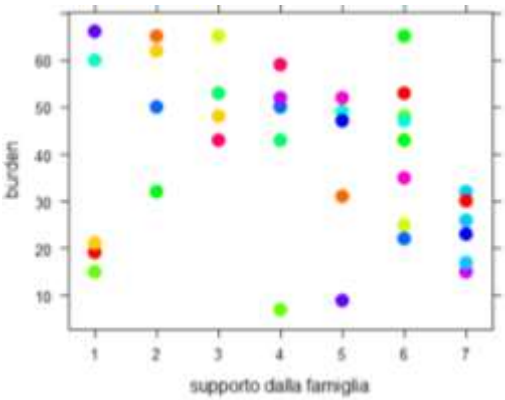
[68] per chi ci volesse provare: $z_r = .5 * \log \frac{1+r}{1-r}$; $SE_r = \frac{1}{\sqrt{N-3}}$, ove 3 è una costante; $CI_{\rho} = z_r \pm \alpha_{/2} \times SE_r$. Per esprimere i limiti del CI sulla stessa scala di *r* campionario, bisogna "tornare indietro" dal logaritmo con la funzione esponenziale: $UL = \frac{e^{2 \times UL_{log}} - 1}{e^{2 \times UL_{log}} + 1}$ e $LL = \frac{e^{2 \times LL_{log}} - 1}{e^{2 \times LL_{log}} + 1}$, dove *e* è la costante di Nepero= 2.718

```
 xyplot(burden_totale~Zimet_supporto_sociale_totale, data= a
  , pch=19, col=rainbow(15), ylab="burden", xlab="supporto t
  otale", cex=1.5)
(burden_supporto<-cor.test(~burden_totale + Zimet_supporto_s
 ociale_totale, data=a))
 Pearson's product-moment correlation
data:  a$burden_totale and a$Zimet_supporto_sociale_totale
t = -0.052532, df = 38, p-value = 0.9584
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3191835  0.3037943
sample estimates:
         cor
-0.008521465

burden_supporto$estimate^2
cor
7.261537e-05
```



Invece, sembra proprio che le due variabili siano **indipendenti** in popolazione: nel grafico non emerge alcuna chiara presentazione dei punti, né lineare né curvilinea. Il coefficiente è indiscutibilmente $r = 0$, e inevitabilmente il CI comprende valore previsto da H_0: $\rho = 0$. Forse aver usato i totali oscura relazioni più sottili tra le due dimensioni? Vediamo se è rilevante almeno il supporto dei familiari (`$zimet_supporto_famiglia`):

```
xyplot(burden_totale~Zimet_supporto_famiglia, data= a, pch
   =19, col=rainbow(15), ylab="burden", xlab="supporto dall
   a famiglia", cex=1.5)
(burden_famiglia<-cor.test(~burden_totale + Zimet_supporto
   _famiglia, data=a))
   Pearson's product-moment correlation

data:  burden_totale and Zimet_supporto_famiglia
t = -2.1764, df = 38, p-value = 0.03581
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5838747 -0.0238848
sample estimates:
       cor
-0.3329171

burden_famiglia$estimate^2
      cor
0.1108338
```



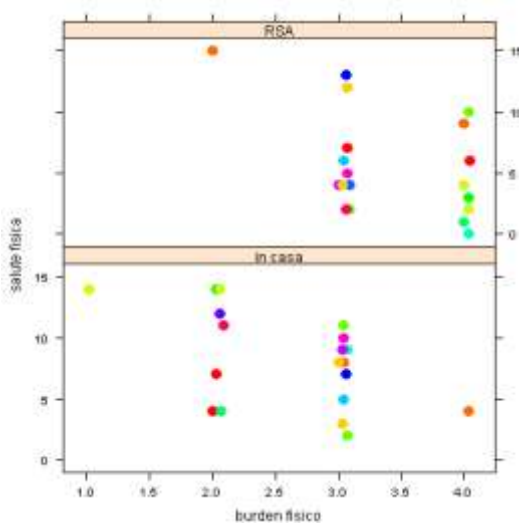
Ora nel grafico la relazione negativa è più facilmente individuabile, anche se i casi che non rispettano la relazione sono diversi. La relazione tra carico e supporto famigliare nel campione è negativa e di entità piuttosto debole: le due variabili condividono l'11.1% di varianza. In popolazione la relazione non è casuale: il CI non comprende (ma per poco) il valore previsto da $H_0 = \rho = 0$, anche se la sua ampiezza ostacola la precisione della stima, che varia da una relazione praticamente assente a una relazione discreta.

Verificate la relazione tra le altre dimensioni del supporto sociale e il burden complessivo: quali conclusioni potremmo trarne rispetto alle fonti di sostegno dei caregiver di pazienti con demenza?

Verifichiamo ora se la relazione tra il **carico fisico** dell'assistenza e la qualità della vita intesa come **soddisfazione per la propria salute fisic**a sia la stessa nei caregiver che assistono il proprio caro in casa e in coloro che l'hanno temporaneamente ricoverato in una RSA. La relazione in popolazione dovrebbe essere negativa, ma forse i due gruppi costituiscono in realtà due popolazioni diverse, in cui le relazioni tra burden fisico e salute seguono andamenti diversi.

Potremmo fare due **subset** ed eseguire le correlazioni in ciascuno, uno per volta; invece, impariamo a usare `xyplot` come un **coplot** (plot condizionale) e a usare l'argomento `subset=` nella funzione `cor.test`. Per rappresentare la distribuzione bivariata, in `xyplot` inseriremo la variabile in base alla quale devono essere presentati plot separati come **argomento condizionato** dopo la formula: **`|fattore`**. L'argomento `subset=` richiede che venga specificato il livello della variabile `factor` per cui intendiamo eseguire l'analisi. Nel nostro caso, la variabile filtro è `$domicilio_assistito` e i suoi due livelli sono "`in casa`" e "`RSA`".

```
xyplot(CBI_burden_fisico~QOL_salute_fisica|domicilio_assi
 stito, data= a, pch=19, cex=1.5, col=rainbow(15), xlab="
 burden fisico", ylab="salute fisica")

cor.test(~CBI_burden_fisico+QOL_salute_fisica, data= a,
 subset=domicilio_assistito=="in casa")
 Pearson's product-moment correlation

data:  CBI_burden_fisico and QOL_salute_fisica
t = -2.6085, df = 18, p-value = 0.01778
alternative hypothesis: true correlation is not equal to
 0
95 percent confidence interval:
-0.7844606 -0.1057448
sample estimates:
       cor
 -0.5237567
cor.test(~CBI_burden_fisico+QOL_salute_fisica, data= a,
 subset=domicilio_assistito=="RSA")
Pearson's product-moment correlation

data:  CBI_burden_fisico and QOL_salute_fisica
t = -1.7046, df = 18, p-value = 0.1055
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6998714  0.0834790
sample estimates:
      cor
-0.372814
```



Il coplot suggerisce che una diversità ci sia: mentre i punti dei soggetti RSA sono ammassati in una limitata area della distribuzione bivariata, il che suggerisce assenza di relazione, i punti dei soggetti che assistono il paziente in casa sono più dispersi, e, anche se debolmente, si intuisce l'emergere di una relazione negativa. I coefficienti di correlazione confermano. Nel gruppo RSA la relazione, di segno negativo, è **debole** e **non significativa** in popolazione: il CI è molto ampio e comprende 0; nell'altro gruppo, la relazione negativa, di entità **discreta**, è **significativa** in popolazione: anche se molto ampio, il CI non comprende 0. L'ampiezza dei due CI, come la stentata significatività raggiunta dal coefficiente del secondo gruppo nonostante la sua discreta entità, possono almeno in parte dipendere dalla **ridotta numerosità dei due gruppi**: quando consideravamo il campione nel suo complesso, avevamo $N = 40$ e $df = 38$; in questo caso abbiamo N_1 e $N_2 = 20$ e $df = 18$. Notate che il coefficiente prima calcolato tra supporto della famiglia e burden totale era pari a $r = |.333|$ con $p = .035$: era quindi più piccolo del coefficiente tra burden fisico e salute fisica per il campione RSA, e tuttavia era risultato significativo, a differenza di quest'ultimo e a ennesima dimostrazione della perdita di potenza dei test di significatività al diminuire di N.

Un dettaglio tecnico prima di proseguire: come **gestire valori mancanti NA** in una o entrambe le variabili da correlare? Naturalmente, se un soggetto ha un valore mancante in una delle misure oggetto di correlazione, quel soggetto viene escluso dall'analisi; il punto è come farlo capire a R. Con la funzione `cor.test` non c'è da preoccuparsi: di default sono considerate per l'analisi solo le osservazioni senza NA in entrambe le variabili, quindi non serve specificare nulla. Con `cor`, invece, se c'è almeno un NA in una delle due variabili in correlazione, l'output restituito sarà un NA: è necessario

specificare l'argomento **use = "complete.obs"** o **use = "pairwise.complete.obs"**: complete.obs è il criterio più crudele, che determina un'esclusione di tipo **listwise** in cui saranno esclusi dalla correlazione tutti quei casi che hanno almeno un valore mancante in **una qualsiasi** delle variabili del dataframe, anche se non è una delle variabili in analisi. pairwise.complete.obs usa, invece, un criterio di esclusione **pairwise**, più permissivo, che elimina dall'analisi i casi che hanno NA **solo** nelle variabili correlate.

8.2.2 Prerequisiti del coefficiente di Pearson: distribuzione normale bivariata

Abbiamo detto che Pearson riprende il lavoro di Galton per il suo coefficiente. In effetti, Galton concepisce la relazione tra due variabili come **rapporto delle loro due medie**: dati i suoi interessi eugenetici, era interessato a questioni come "qual è l'altezza media di figli di padri insolitamente alti, rispetto all'altezza media dei loro padri?" Negli sviluppi successivi, il rapporto viene sempre fatto considerando le **popolazioni**, e non i campioni, dato che è solo con campioni tendenti a infinito che il rapporto tra le medie dà valori identici al coefficiente r. Usiamo l'esempio di Rodgers e Nicewander (1988): X è il QI di una popolazione di madri, e Y il QI dei loro primogeniti. Standardizziamo le distribuzioni in modo che: $\mu_X = 0$ e $\mu_Y = 0$, $\sigma_X = 1$ e $\sigma_Y = 1$. Selezioniamo (arbitrariamente) un valore di X grande (X_C), e calcoliamo la media del QI delle madri con $QI > X_C$, che chiamiamo $\mu(X|X > X_C)$. Poi facciamo la media dei QI dei primogeniti di queste madri selezionate: $m(Y|Y > X_C)$. La relazione tra i QI di queste madri e dei loro primogeniti, in popolazione, è data da:

$$r = \frac{\mu(Y|X>X_C)-\mu_Y}{\mu(X|X>X_C)-\mu_X} = \frac{\mu(Y|X>X_C)}{\mu(X|X>X_C)}$$

Questa equazione, però, richiede l'assunzione della **normalità bivariata** XY: la distribuzione normale bivariata è il caso più semplice, limitato a due variabili, di distribuzione **normale multivariata**[69], che è una generalizzazione, da 2 a n dimensioni, della normale univariata quando si rilevano contemporaneamente più misure sul campione.

Esistono sia test inferenziali per l'ipotesi nulla che la distribuzione campionaria sia normale bivariata (multivariata), sia grafici ad hoc: ne vedremo rapidamente (il discorso è più complesso di come lo affronteremo) due di ciascun tipo, cioè il test di Mardia e il test di Royston, il χ^2 Q-Q plot e il perspective plot. I due test e i due grafici si gestiscono facilmente con **mvn** di **MVN**: **mvn(data=matrice, mvnTest="royston" oppure "mardia", multivariatePlot="qq" oppure "persp")**. L'argomento data= richiede un dataframe o una matrice composti dalle sole variabili che costituiscono la distribuzione in analisi (nel nostro caso, le due variabili da correlare); in mvnTest= scegliamo uno dei test inferenziali e in multivariatePlot= quale plot visualizzare.

Il **test di Mardia** calcola **l'asimmetria** e la **curtosi multivariate** di una distribuzione multivariata (nel caso delle correlazioni, bivariata): per ciascuna di esse, **mvn** fornisce il **valore di confronto**, che è dato da un quantile della distribuzione **chi quadrato** per l'asimmetria (per grandi campioni, il coefficiente di asimmetria multivariata g_{ip} segue una distribuzione casuale χ^2) e da un quantile della distribuzione z per la curtosi (il coefficiente di curtosi multivariata g_{2p} segue una distribuzione normale), oltre ai $p - value$ loro associati; dice anche esplicitamente se la distribuzione è (MVN= **YES**) o non è (MVN= **NO**) normale multivariata. L'output riporta anche le statistiche univariate di normalità (e le statistiche descrittive campionarie per ciascuna variabile. Di default è mostrato il **test univariato di Anderson-Darling** (Stephens, 1974), che stima la probabilità di estrarre una distribuzione da una distribuzione aleatoria attesa, nel nostro caso una distribuzione normale. Si può produrre il test di Shapiro – Wilks a noi noto specificando univariateTest="SW": opzione di default fino al penultimo aggiornamento, il test di Shapiro – Wilks è sconsigliato per più di 5000 casi o meno di 3 casi, condizioni per noi decisamente rare.

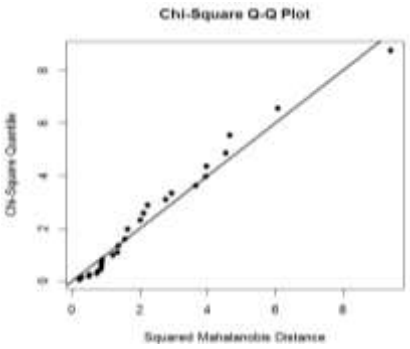
[69] Ce ne occuperemo nella regressione multipla

Il **test di Royston** rappresenta **l'estensione del test di Shapiro – Wilks** al confronto tra una distribuzione normale multivariata e la distribuzione campionaria multivariata. Nell'output, un valore $p < .05$ indica quindi una differenza non trascurabile tra la distribuzione campionaria multivariata e la distribuzione normale multivariata; un valore $p > .05$ suggerisce la sovrapposizione tra le due distribuzioni. Anche in questo caso, è esplicitata l'interpretazione da dare alla normalità multivariata (MVN= YES/NO) e sono fornite le statistiche univariate.

L'analogo multivariato del Q-Qplot univariato è il **χ^2 Q-Q plot**,Errore. Il segnalibro non è definito.che si interpreta esattamente come un Q-Q plot univariato: più i punti si dispongono lungo la retta di riferimento, maggiore è l'adesione della distribuzione bivariata alla bivariata normale. A differenza dei due test e del χ^2 Q-Q plot, il **perspective plot** Errore. Il segnalibro non è definito.si applica **solo a distribuzioni bivariate** e non multivariate: la figura rappresentata nel perspective plot è un'elegante forma tridimensionale, che, se la distribuzione fosse effettivamente normale bivariata, sarebbe una perfetta **campana tridimensionale**.

Per esempio, la distribuzione bivariata di burden fisico e salute fisica, che avevamo prima correlato, non è sovrapponibile a una normale bivariata; la distribuzione univariata del burden è affine alla normale, quella della salute fisica no:

```
fisico_salute<-data.frame(burden_fisico = attaccamento$CBI_burden_fisico, salute_fisica =
   attaccamento$QOL_salute_fisica)
mvn(data = fisico_salute, mvnTest = "royston", multivariatePlot
    = "qq")
$multivariateNormality
     Test        H      p value MVN
1 Royston 16.04348 0.0002707647  NO

$univariateNormality
             Test      Variable Statistic   p value Normality
1 Anderson-Darling burden_fisico    0.6681  0.0751       YES
2 Anderson-Darling salute_fisica    2.4677  <0.001       NO

$Descriptives
               n  Mean   Std.Dev Median  Min   Max   25th
    75th
burden_fisico 40 7.175 4.1872058   7.00 0.00 15.00 4.0000
    10.2500
salute_fisica 40 3.016 0.7619974   3.05 1.02  4.04 2.7725
    3.3175
                    Skew   Kurtosis
burden_fisico  0.2090242 -1.2083610
salute_fisica -0.3412332 -0.4646424
```

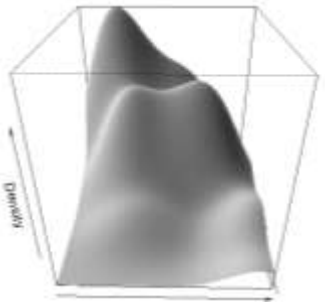


Indicando il test di Shapiro – Wilks come opzione per il test univariato, avremmo lo stesso risultato univariato:

```
mvn(data = fisico_salute, mvnTest = "royston", multivariatePlot = "qq", univariateTest= "SW")
$multivariateNormality
     Test        H      p value MVN
1 Royston 16.04348 0.0002707647  NO

$univariateNormality
          Test      Variable Statistic   p value Normality
1 Shapiro-Wilk burden_fisico    0.9507    0.0801    YES
2 Shapiro-Wilk salute_fisica    0.8610    0.0002    NO
```

Vediamo cosa risulterebbe con il test di Mardia e un perspective plot:

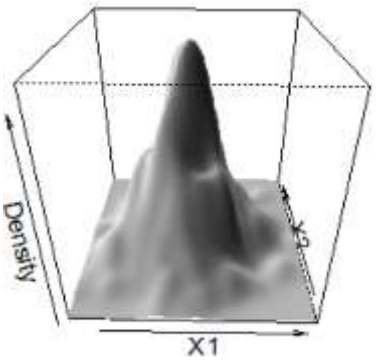
```
mvn(data =fisico_salute, mvnTest= "mardia",multivariatePlot = "persp")
$multivariateNormality
             Test         Statistic           p value Result
1 Mardia Skewness  1.44568866410965 0.836216502094745    YES
2 Mardia Kurtosis -1.19524350568992 0.231991905197849    YES
3             MVN              <NA>              <NA>    YES
```



La parte successiva dell'output ripete le descrittive e i test univariati; il grafico non rappresenta davvero una "elegante curva a campana".

Se volete vedere un perspective plot di una distribuzione bivariata ragionevolmente normale, provate a creare due variabili aleatorie da una distribuzione normale; sappiamo come si fa:

```
set.seed(123)
norma1<-rnorm(1000, mean = 10,sd = 2)
norma2<-rnorm(1000, mean = 5,sd = 1.5)
normale<-data.frame(X1=norma1, X2=norma2)
mvn(data = normale, mvnTest = "mardia", multivariatePlot = "persp")
$multivariateNormality
             Test          Statistic           p value Result
1 Mardia Skewness  0.918801672005227 0.921848052734316    YES
2 Mardia Kurtosis -0.605687764712657  0.54472211028202    YES
3             MVN               <NA>              <NA>    YES
$univariateNormality
             Test  Variable Statistic   p value Normality
1 Anderson-Darling    X1        0.2965    0.5920       YES
2 Anderson-Darling    X2        0.2984    0.5864       YES
```

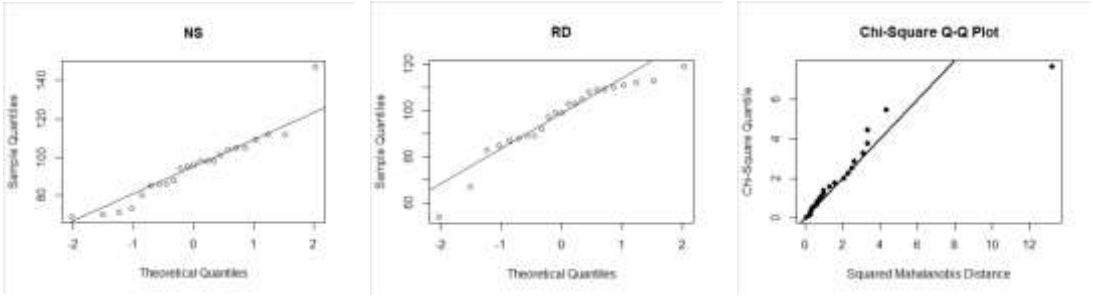


In questo esempio, da due variabili univariate normali è emersa una distribuzione bivariata normale, ma non è un risultato scontato, come avevamo anticipato. Lo verifichiamo usando due variabili di un dataframe usato per gli esami (d), e che quindi non sveleremo completamente: contiene due misure di personalità, Novelty Seeking e Reward Dependence del test TCI di Cloninger (§6.3), registrate su 23 casi divisi in controlli e pazienti:

```
d$Novelty_Seeking
 [1]  71  94  69 101  86  96  95  70  98 109  88  98  85 105  73  80 105 86 104 112 147 112  98
d$Reward_Dependence
 [1] 112  83 111 103  85 109  67  88  99 110 105 109  92 119  89 108  87 99  97 103  54 113  89
```

Uniamole in un dataframe e applichiamo il test di Royston: due distribuzioni univariate normali e una distribuzione bivariata non normale:

```
mag<-data.frame(NS=d$Novelty_Seeking, RD=d$Reward_Dependence)
mvn(mag,mvnTest = "royston", univariateTest = "AD", multivariatePlot= "qq")
$multivariateNormality
     Test        H    p value MVN
1 Royston 8.353407 0.01515666  NO

$univariateNormality
             Test  Variable Statistic   p value Normality
1 Anderson-Darling    NS        0.4488    0.2535     YES
2 Anderson-Darling    RD        0.6309    0.0878     YES
```

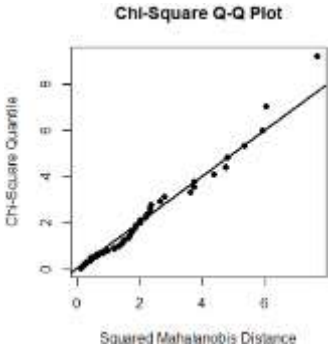


Attenzione a un equivoco, ahimè non raro agli esami: la normalità bivariata (e le sue rappresentazioni grafiche) costituiscono un requisito di applicabilità del test parametrico di correlazione, **ma** non sono affatto un modo per indicare che le due variabili siano correlate. Vediamo un esempio con due variabili aleatorie tratte da una distribuzione normale teorica:

```
set.seed(123)
x1<-rnorm(n = 50, mean = 0, sd = 1)
x2<-rnorm(n = 50, mean = 0, sd = 1)
```

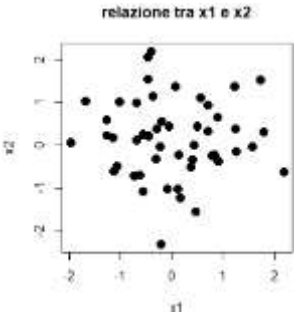
La loro distribuzione bivariata dovrebbe essere normale:

```
mvn(data = data.frame(x1,x2), mvnTest = "royston", multivariatePlot =
 "qq")
$multivariateNormality
     Test      H p value MVN
1 Royston 0.0255   0.987 YES

$univariateNormality
          Test  Variable Statistic   p value Normality
1 Shapiro-Wilk    x1         0.989     0.928    YES
2 Shapiro-Wilk    x2         0.991     0.962    YES
[omissis]
```



infatti, la distribuzione di X_1 e X_2 è normale bivariata. Ma le due variabili sono forse correlate?

```
cor(x1,x2)
[1] -0.0359
```



No, non lo sono affatto. Considerate che la correlazione si applica a distribuzioni appaiate (il punteggio in x_1 e x_2 del soggetto S_1, del soggetto S_2, ecc.), ma non è così per nella verifica della normalità delle distribuzioni bivariate.

8.3 Quantificare e verificare ipotesi su una distribuzione bivariata con variabili ordinali

In questo paragrafo ci occupiamo della quantificazione e della verifica di ipotesi nel caso di due variabili ordinali; i test presentati si adottano anche nel caso in cui una variabile sia ordinale e l'altra metrica, o nel caso di due variabili metriche con distribuzione bivariata non normale (in questo caso, sono più robusti, quindi preferibili al test di Pearson). Vedremo il test rho di Spearman e il test tau di Kendall: entrambi valutano i dati sotto condizione di $H_0: \rho_{X_1X_2} = 0$, ovvero assumendo che le variabili siano tra loro **monotonicamente indipendenti**, cioè non abbiano **andamento comune**.

8.3.1 Il test rho di Spearman

Spearman riprende, tra il 1904 e il 1906, i concetti alla base della correlazione di Pearson per proporre un **coefficiente di correlazione rho (ρ) basato sui ranghi**, successivamente ridiscusso e corretto più volte (tra gli altri, da Gosset): in effetti, il suo coefficiente consiste nel **calcolo del coefficiente r sui ranghi delle variabili**. Il coefficiente ρ varia quindi anch'esso da -1 a +1, passando per 0 (perfetta indipendenza).

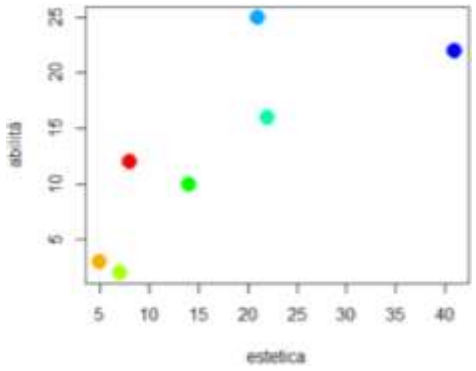
Non sorprenderà nessuno, probabilmente, che, nonostante questa connessione tra i due coefficienti, i rapporti personali e professionali tra i loro due autori, che lavorarono a lungo per lo stesso Dipartimento, siano stati improntati a una reciproca disistima (ad esempio, Pearson, 1904[70]: "*The formula invented by Mr Spearman […] is clearly wrong […] not only are his formulae, expecially for probable errors erroneous, but he quite misunderstands and misuses partial correlation coefficient*").

Il calcolo del coefficiente rho prevede la trasformazione dei dati in **ranghi**: la logica è che se le due variabili sono correlate, un soggetto che otterrà ranghi bassi in una variabile, otterrà ranghi bassi (o alti, se la correlazione è negativa) anche nell'altra variabile: anche in questo caso, quindi, la relazione prevede la ricerca di un qualche tipo di **concordanza** tra i dati.

[70] pag. 160. Pearson non si riferisce qui alla formula di rho, che comunque non si perita di criticare altrove.

Usiamo dati inventati, relativi a **sette** cagnolini (da A a G), con altissimo pedigree, che nel tour annuale di concorsi sono stati classificati sia rispetto agli standard estetici di razza ($\boldsymbol{X_1}$) sia rispetto a prove di affiatamento cane – padrone ($\boldsymbol{X_2}$). La loro allevatrice, un po' delusa dall'esito, pensa che i **giudizi dei giudici nelle due categorie di premio non siano realmente indipendenti**, ovvero che il giudizio sull'abilità sia legato all'aspetto estetico delle bestiole – e viceversa. Riportiamo i piazzamenti nelle due categorie ottenuti, nell'ultimo concorso, dalle sue creature:

```
cagnolino<-c("A","B","C","D","E","F","G")
x1_estetica<-c(8,5,7,14,22,21,41)
x2_abilita<-c(12,3,2,10,16,25,22)
rho<-data.frame(cagnolino, x1_estetica, x2_abilita)
rho
  cagnolino x1_estetica x2_abilita
1         A           8          12
2         B           5           3
3         C           7           2
4         D          14          10
5         E          22          16
6         F          21          25
7         G          41          22
```



Per prima cosa, **assegniamo separatamente i ranghi** a ciascuna distribuzione: per estetica, il meglio piazzato è B, al secondo posto C, ecc.; per abilità, il meglio piazzato è C, segue B, e così via. Usiamo `rank(variabile)` per creare le due distribuzioni `$ranghi_x1` e `$ranghi_x2`:

```
rho$ranghi_x1<-rank(x1_estetica)
rho$ranghi_x2<-rank(x2_abilita)
rho
  cagnolino x1_estetica x2_abilita ranghi_x1 ranghi_x2
1         A           8          12         3         4
2         B           5           3         1         2
3         C           7           2         2         1
4         D          14          10         4         3
5         E          22          16         6         5
6         F          21          25         5         7
7         G          41          22         7         6
```

Effettivamente, i cagnolini che occupano i primi posti in X_1 occupano i primi posti anche in X_2, e così naturalmente gli ultimi; se vi fosse una perfetta **concordanza positiva** tra i giudizi ricevuti, la differenza tra i ranghi dovrebbe essere pari a 0[71]:

```
rho$concordanza<-rho$ranghi_x1-rho$ranghi_x2
rho
  cagnolino x1_estetica x2_abilita ranghi_x1 ranghi_x2 concordanza
1         A           8          12         3         4          -1
2         B           5           3         1         2          -1
3         C           7           2         2         1           1
4         D          14          10         4         3           1
5         E          22          16         6         5           1
6         F          21          25         5         7          -2
7         G          41          22         7         6           1
```

Non ci sono $differenze = 0$, ma sono comunque piccole.

A questo punto, Spearman **applica il coefficiente di correlazione di Pearson alle** due distribuzioni di ranghi ottenute:

```
cor(rho$ranghi_x1, rho$ranghi_x2)
[1] 0.8214286
```

Infatti, se usiamo la funzione `cor` sulle distribuzioni dei dati grezzi cambiando l'argomento `method=` in "**`spearman`**":

[71] In effetti, la somma delle differenze (al quadrato) tra i ranghi è un vero e proprio test: **test di Hotelling – Pabst** (Hotelling e Pabst, 1936), con proprie tavole di significatività, oggi in disuso; rientra nella formula semplificata di rho: $\rho = 1 - \frac{6 \times \sum d_R^2}{N \times (N^2 - 1)} 6$.

```
cor(x1_estetica, x2_abilita, method = "spearman")
[1] 0.8214286
```

Se avessimo calcolato un coefficiente di Pearson sulle distribuzioni dei dati **grezzi**, invece:

```
cor(x1_estetica, x2_abilita, method = "pearson")
[1] 0.7924566
```

Quindi, la **relazione tra giudizio estetico e giudizio sull'affiatamento è positiva e di intensità più che discreta**: si direbbe che l'allevatrice abbia ragione, e che i giudici non abbiano saputo separare una valutazione dall'altra. Consideriamo, però, la **possibilità che il coefficiente campionario sia solo una fluttuazione casuale del vero valore atteso** in popolazione: H_0: $\rho = 0$, e associamo un $p - value$ al coefficiente ρ con **`cor.test(x1,x2, method="spearman)`**.

```
cor.test(x1_estetica, x2_abilita, method = "spearman")
        Spearman's rank correlation rho
data:  x1_estetica and x2_abilita
S = 10, p-value = 0.03413
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.8214286
```

Il $p - value$ associato alla correlazione è inferiore alla soglia $\alpha = .05$: cade quindi nella zona di rifiuto di H_0. In popolazione, probabilmente i due giudizi sono davvero correlati positivamente. Notiamo che nell'output non è presente il CI: per calcolarlo, è possibile usare una procedura di ricampionamento (**bootstrap**).

8.3.2 il test tau di Kendall

Trent'anni dopo Spearman (1938, 1948) sir Kendall propone il nuovo **coefficiente di correlazione tau** (**τ**): si utilizza per la **stessa H_0,** con lo **stesso tipo di dati** e con lo **stesso range di variazione da -1 a +1** del coefficiente ρ, ma può essere più indicato per **campioni molto piccoli** e con molti ranghi uguali (***ties***).

Anche i primi passaggi per il calcolo del coefficiente τ sono uguali a quelli per ρ: si assegnano i ranghi ai dati, iniziando da quelli della variabile X_1 e successivamente a quelli di X_2.

Riprendiamo i nostri cagnolini per valutare se il giudizio sulla relazione tra estetica e abilità cambia usando un diverso coefficiente; in fondo, il campione è davvero piccolo ($N < 10$), e quindi il test tau è più coerente con il tipo di distribuzione bivariata rispetto al test rho.

Avevamo già assegnato separatamente i ranghi a X_1 e X_2: riprendiamoli, ma questa volta **ordiniamo i soggetti in base ai ranghi ottenuti in X_1, secondo un andamento crescente**: vi ricordate la funzione `order` che abbiamo usato nel §3.2? L'argomento `decreasing=FALSE` di default è quello che ci serve. Eliminiamo la concordanza tra i ranghi calcolata per il rho di Spearman, che non serve più.

```
tau<-rho[order(rho$ranghi_x1), -6]
tau
```

cagnolino	x1_estetica	x2_abilita	ranghi_x1	ranghi_x2
2	B	5	3	1	2
3	C	7	2	2	1
1	A	8	12	3	4
4	D	14	10	4	3
6	F	21	25	5	7
5	E	22	16	6	5
7	G	41	22	7	6

Ora concentriamoci su cosa questo ordinamento ha prodotto in `ranghi_x2`. In ranghi_x_1, ordinata in senso crescente, **tutte le coppie di ranghi seguono ovviamente un ordine naturale concordante**, per cui $X_{R_{1.1}} < X_{R_{1.2}}, X_{R_{1.1}} < X_{R_{1.3}} \dots X_{R_{N-1}} < X_{R_N}$. Cosa succede **all'ordine naturale dei ranghi di X_2**, assegnati alle osservazioni disposte secondo le posizioni in X_1? Se la relazione tra X_1 e X_2 fosse **positiva**, anche i ranghi di X_2 dovrebbero risultare in **ordine crescente, concordante** con l'ordine di X_1; se la relazione tra X_1 e X_2 fosse **negativa**, i ranghi della variabile X_2 dovrebbero risultare in **ordine decrescente, discordante** con l'ordine di X_1. Se la relazione fosse $= \mathbf{0}$, l'ordine dei ranghi della variabile X_2 sarà **casuale**: il numero di ranghi concordanti $(+1)$ e discordanti (-1) dall'ordine naturale tenderà ad essere uguale, **con somma algebrica = 0**.

Contiamo, allora, il numero di casi concordanti con l'ordine (cui assegniamo +1) e quello dei casi discordanti con l'ordine (cui assegniamo -1) per ciascuno dei ranghi di X_2:

```
tau$cagnolino; tau$ranghi_x2
[1] B C A D F E G
[1] 2 1 4 3 7 5 6
```

R_B= 2: il rango 2 è seguito da R_C=1, che è discordante dall'ordine → -1; è seguito anche da cinque ranghi >2, dunque concordanti → +5; complessivamente R_B = **+4**

R_C= 1: il rango 1 è seguito da cinque ranghi >1, concordanti → **+5**

R_A= 4: il rango 4 è seguito da R_{D1}= 3, discordante dall'ordine → -1; è seguito anche da tre $R > 4$, quindi concordanti → +3; complessivamente R_A = **+2**

R_D= 3: è seguito da tre ranghi >3, quindi concordanti → **+3**

R_F= 7: ahimè, è seguito solo da due ranghi <7, perciò discordanti → **-2**

R_E= 5: è seguito da un rango >5 e quindi concordante → **+1**

R_G= 6: l'ultimo rango non partecipa al calcolo

La somma delle **concordanze** è pari a:

```
(concordanze<-sum(4,5,2,3,-2,1))
[1] 13
```

Naturalmente, accoppiando i sette ranghi ($N = 7$) a due a due ($r = 2$) in tutte le combinazioni possibili, potrei ottenere tredici concordanze solo per caso. E qual è la probabilità di ottenere **solo** per caso queste 13 concordanze? Se il $p - value$ associato fosse molto piccolo, ci troveremmo di fronte a un evento molto raro, per cui il dato non darebbe sostegno all'ipotesi nulla che le due variabili estetica e abilità non siano correlate. Al contrario, se il $p - value$ fosse abbastanza alto, i dati confermerebbero l'ipotesi nulla di indipendenza delle due variabili. Per prima cosa, dovremo calcolare tutte le possibili combinazioni a due a due dei sette ranghi, e rapportare a questo massimo teorico di combinazioni quelle empiricamente ottenute (13). Potete fare riferimento all'Appendice I per la spiegazione del calcolo delle combinazioni, oppure fidarvi[72], e scoprire che:

$$ {}_nC_r = \left[\frac{N}{r}\right] = \frac{N!}{r!\,(N-r)!} $$ → `factorial(7)/(factorial(2)*factorial(7-2))` `[1] 21`

Il coefficiente tau di Kendall è il risultato del **rapporto tra le concordanze riscontrate nel campione (N=13) e il numero massimo di combinazioni** dei dati (=21):

$$ \tau = \frac{concordanze\ ottenute}{concordanze\ possibili} $$

`13/21` → `[1] 0.6190476`

[72] Fidatevi pure, ma non confondetevi! La "r" della formula delle combinazioni **non è il coefficiente *r* di Pearson**, ma la notazione ("**errupla**") che indica il numero di elementi presi in ogni combinazione: a due a due (r=2), a tre a tre (r=3), eccetera.

Per **campioni piccoli**, i valori critici, corrispondenti a diverse soglie α per la significatività di τ, sono riportati in apposite tabelle:

	α				
N	**.05**	.025	**.01**	.005	*Monodirezionale*
	.1	**.05**	.02	**.01**	*Bidirezionale*
4	1.0				
5	.8	.8	1.0		
6	.733	.867	.867	1.0	
7	.619	**.714**	.870	.810	
8	.571	.643	.714	.786	
9	.500	.556	.667	.722	
10	.467	.511	.600	.644	
		eccetera			

Per $N = 7$, H_1 bidirezionale e $\alpha = .05$, abbiamo un coefficiente $\tau_{critico} = .714$: poiché il nostro $\tau_{ottenuto} = .\mathbf{619} < \tau_{critico}$, **accettiamo H_0**.

Per **campioni grandi**, si usa la normale Z; comunque, basterà chiedere il $p - value$ a R, che sa come si fa:

```
cor.test(x1_estetica, x2_abilita, method = "kendall")
        Kendall's rank correlation tau
data:  x1_estetica and x2_abilita
T = 17, p-value = 0.06905
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.6190476
```

Ed ecco una **discrepanza sulla decisione da prendere rispetto ad** H_0: rho ci aveva consigliato di accettare H_1, tau propende, anche se decisamente a soglia, per H_0. La discordanza tra i due coefficienti non parametrici non è insolita: soprattutto con campioni molto piccoli, come questo, ρ e τ vanno poco d'accordo, mentre i **valori dei coefficienti r e ρ tendono sempre a convergere**, tanto che per N ampi sono sostanzialmente identici.

```
cor(x1_estetica, x2_abilita, method = "p"); cor(x1_estetica, x2_abilita, method = "s");
 cor(x1_estetica, x2_abilita, method = "k")
[1] 0.7924566
[1] 0.8214286
[1] 0.6190476
```

Attenzione, però: anche se r e ρ con grandi campioni sono quasi identici, il loro **quadrato** non dà le stesse informazioni: il coefficiente di determinazione R^2 esprime effettivamente la proporzione di varianza condivisa, ma il coefficiente $\boldsymbol{\rho^2}$ è la **proporzione di varianza nei ranghi condivisa**. Il povero τ^2, invece, la cui natura è completamente diversa da r e rho, non ci dice nulla sulla proporzione di variabilità condivisa, nemmeno su quella dei ranghi.

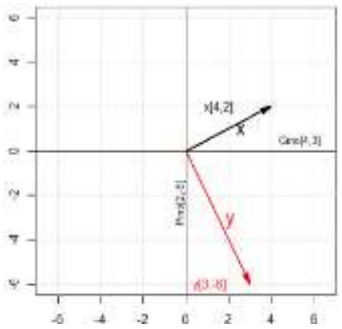
8.4 Matrici di correlazioni

Non è mai una buona cosa, anzi, è perlopiù una pessima idea, fare correlazioni "a pioggia" sulle variabili della ricerca, per il solo motivo che ci sono e senza un'idea della rete delle relazioni concettuali alle spalle delle loro operazionalizzazioni. Statisticamente parlando, questo procedimento aumenta la probabilità di commettere un errore di I tipo ben oltre la massima soglia prevista di .05 (§8.4.2); teoricamente parlando, si corre il rischio di interpretare correlazioni – spazzatura, oppure correlazioni spurie (§8.4.3, §8.4.4). Prendiamo il discorso un po' alla lontana, approfittandone per approfondire le correlazioni lineari, e vediamo perché.

8.4.1 Le correlazioni lineari come rappresentazioni geometriche

I **coefficienti di correlazione *r*** possono essere rappresentati geometricamente come l'**angolo tra** o come la **distanza tra le variabili, rappresentate come punti all'interno di uno spazio cartesiano** chiamato ***person space*** o ***subject space***: gli **assi** di questo spazio sono i **soggetti** e le **coordinate** di ogni punto / variabile nello spazio sono definite dai **punteggi** ottenuti dai soggetti. In uno spazio bidimensionale ($N = 2$) o tridimensionale (tre soggetti: $N = 3$) la rappresentazione è comprensibile all'occhio umano, ma diventa impossibile da visualizzare nell'iper-spazio definito da un maggior numero di soggetti / assi (chi mai raccoglierebbe campioni di sole tre persone per correlare variabili?). L'impossibilità della visualizzazione non coincide affatto, però, con l'impossibilità matematica: la rappresentazione delle variabili come punti definiti dalle coordinate in ogni soggetto è pratica corrente per concettualizzare **multiple** correlazioni tra variabili, in **matrici di correlazioni**; si usa, ad esempio, nelle tecniche di analisi fattoriale. Se avete bisogno di una rinfrescata alle definizioni fondamentali della trigonometria, date un'occhiata all'Appendice VI.

Cominciamo a vedere le **correlazioni come angoli** tra le variabili, in concreto. Sottoponiamo due test, X e Y, a due soggetti: Gino ottiene $X = 4$ e $Y = 3$, Pino invece $X = 2$ e $Y = -6$. Quindi, la variabile / il vettore X è $\boldsymbol{X[4 - Gino, 2 - Pino]}$ e la variabile Y è $\boldsymbol{Y[3 - Gino, -6\ Pino]}$. Il *person space* è rappresentabile da due assi (asse Gino e asse Pino) al cui interno tracciamo i vettori X e Y[73], più correttamente definibili come **vettori posizione**, con partenza dall'origine degli assi e destinazione definita dalle coordinate XY di ogni vettore.



Quanto è **simile l'andamento** delle due variabili?

Un primo indicatore della somiglianza tra i due vettori grezzi è il loro **prodotto interno** (PI; *dot product*), che è semplicemente la **somma dei prodotti x_i e y_i**:

$$PI_{x,y} = \sum(x_i y_i)$$

↓

È una stima molto "rozza": è tanto maggiore quanto l'andamento dei vettori appaiati è simile, ma **non ha un tetto** inferiore e superiore. Quando il PI viene centrato e diviso per N (o $N - 1$), abbiamo imparato a conoscerlo sotto il nome di **covarianza**.

$$cov_{x,y} = \frac{\sum(x_i - \bar{X})(y_i - \bar{Y})}{N - 1}$$

Se non avete timore di un po' di algebra matriciale, è facile calcolarlo: basta moltiplicare ogni elemento della riga *i* del primo vettore per il corrispettivo elemento del secondo vettore: se il prodotto interno è $= 0$, l'angolo tra i vettori è $= 90°$ $\rightarrow$ le variabili grezze sono **ortogonali**.

Nel nostro caso: $PI = \sum \begin{bmatrix} 4 \\ 2 \end{bmatrix} \times \begin{bmatrix} 3 \\ -6 \end{bmatrix} = (4 \times 3) + (2 \times -6) = 12 - 12 = \mathbf{0}$ $\rightarrow$ le variabili sono effettivamente ortogonali.

Per circoscrivere il PI a un range $0 - 1$ (se i valori di X e Y sono positivi), lo dividiamo per il prodotto delle **lunghezze** dei due vettori. La **lunghezza del vettore è** la **distanza euclidea dall'origine** (**norma euclidea** o $L2\ norm$). In generale,

[73] Il grafico è ottenuto con le funzioni di base che conoscete, tranne i vettori che sono tracciati con la funzione `vectors(variabile)` di `matlib`. La rotazione a 90° dell'etichetta "Pino" è data dall'argomento `srt=90` nella funzione `text`. Se volete crearne uno, ecco lo script:

```
x=c(4,2)
y=c(3,-6)
xlim <- c(-6,6)
ylim <- c(-6,6)
par(mar=c(3,3,1,1)+.1)
plot(xlim, ylim, type="n", xlab="X", ylab="Y", asp=1)
grid()
abline(v=0)
abline(h=0)
vectors(y, labels="Y", pos.lab=4, frac.lab=1, col="red")
vectors(x, labels="X", pos.lab=4, frac.lab=1)
text(x = .5, y = 2, labels = "x[4,2]", pos=4)
text(x = 0, y = -6, labels = "y[3,-6]", pos=4, col="red")
text(x = 2, y = .5, labels = "Gino[4,3]", pos=4, cex=.8)
text(x = -.5, y = -2, labels = "Pino[2,-6]", pos=1, cex=.8, srt=90)
```

una distanza euclidea tra due vettori è la **radice quadrata della somma delle differenze al quadrato tra i vettori:** $d(x,y) = \sqrt{\sum(x_i - y_i)^2}$. Poiché, in questo caso, uno degli estremi del vettore è l'origine degli assi, cioè 0, la formula diviene: $d(x,0) = \sqrt{\sum(x_i - 0)^2} = \sqrt{\sum x_i^2}$.

Questa stima della similarità dei vettori si chiama **somiglianza del coseno** (***cosine similarity***), e si interpreta come il **coseno dell'angolo** tra i due vettori. La sua formula dovrebbe cominciare a sembrarvi familiare:

$$cosine\ sim. = \frac{\sum(x_i y_i)}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

↓

Già. C'è **un solo passo** tra la *cosine similarity* e il coefficiente r, ovvero la centratura di entrambi i vettori sulla propria media.

$$r = \frac{\sum(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum(x_i - \bar{X})^2}\sqrt{\sum(y_i - \bar{Y})^2}}$$

Il coefficiente di Pearson, quindi, altro non è che la *cosine similarity* dei vettori centrati, ovvero il **coseno dell'angolo tra i due vettori centrati**, variante in un range da -1 ($\alpha = 180°$, $r = -1$) a +1 ($\alpha = 0$, $r = +1$).

Attenti alle parentesi:

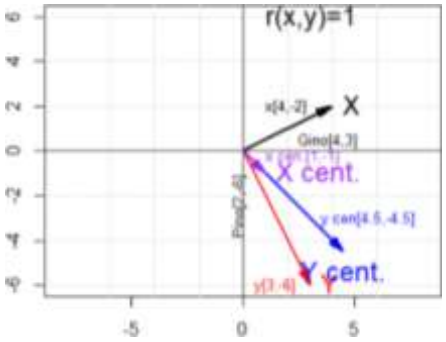
```
sum((x-mean(x))*(y-mean(y))) / (sqrt(sum((x-mean(x))^2))*sqrt(sum((y-mean(y))^2)))
[1] 1
```

La relazione tra X e Y è una perfetta correlazione positiva.

*Se avete bisogno di un ripasso **molto elementare** su seno, coseno, et cetera, date un'occhiata all'Appendice 1*

Vediamo il coseno dei vettori centrati `x_cen=x-mean(x)` e `y_cen=y-mean(y)`: **l'angolo tra i vettori centrati è = 0** (come approfondiremo successivamente, i due vettori sono linearmente dipendenti) e il **coseno di un angolo = 0° è = 1**.

```
cos(0*pi/180)
[1] 1
```



Perciò, se la *cosine similarity* esprime la somiglianza / prossimità / relazione tra i vettori grezzi, r esprime la somiglianza / prossimità / relazione tra i vettori centrati sulla media. Mentre la *cosine similarity* non è invariante rispetto a trasformazioni lineari (se aggiungiamo una costante a X e Y, la *cosine similarity* cambia), la correlazione di Pearson è invariante sia rispetto ai cambiamenti di scala sia rispetto alla trasformazione lineare.

Torneremo nel §8.5 ad usare gli angoli tra i vettori. Vediamo ora come le **correlazioni** possono essere rappresentate dalle **distanze tra le variabili**: facciamo un esempio leggermente più realistico e in uno spazio tridimensionale: somministriamo a tre soggetti tre scale di personalità (Amicalità - A, Apertura mentale - M, Coscienziosità - C), espresse in punti T:

```
A<-c(49,40,48)
M<-c(44,38,45)
C<-c(38,40,46)
round(cor(A,M),3); round(cor(A,C),3); round(cor(M,C),3)
[1] 0.973
[1] 0.179
[1] 0.402
```

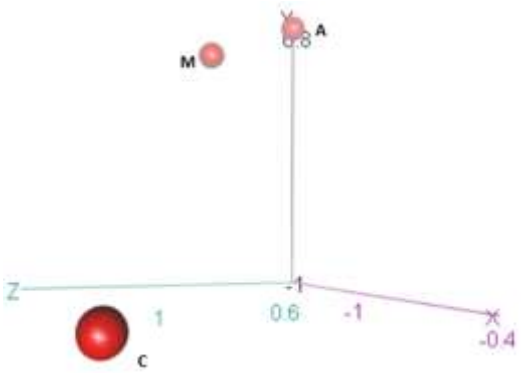
La correlazione tra Amicalità e Apertura Mentale è fortissima, quella tra Amicalità e Coscienziosità è inesistente, quella tra Apertura mentale e Coscienziosità piuttosto debole.

Per creare il *person space*, **standardizziamo** le variabili. Purtroppo, per soddisfare l'equazione che vedremo tra poco, per creare il denominatore dovremo **dividere la varianza per N** (***sample method***) e non per $N-1$: perciò, invece di usare la comoda funzione `scale(variabile)`, dovremo procedere passo passo, facendo strage di parentesi:

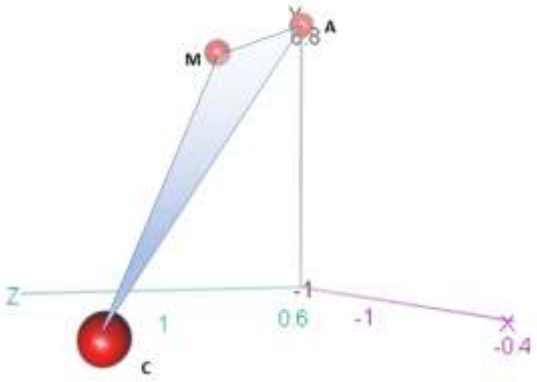
```
za<-(A-mean(A))/sqrt((sum((A-mean(A))^2))/3)
zm<-(M-mean(M))/sqrt((sum((M-mean(M))^2))/3)
zc<-(C-mean(C))/sqrt((sum((C-mean(C))^2))/3)
```

Aggiungiamo l'informazione sui soggetti (s_1, s_2, s_3), che costituiranno i tre assi dello spazio (Y, X, Z), e visualizziamo: facciamoci aiutare da `scatter3d`[74] di `car` e da `rgl` per costruire il *person space* tridimensionale.

```
sogg<-c("s1_Y","s2_X","s3_Z")
dati<-data.frame(sogg=sogg, A=za, M=zm, C=zc)
dati
 sogg                 A             M             C
 1 s1_Y           0.828         0.539        -0.981
 2 s2_X          -1.407        -1.402        -0.392
 3 s3_Z           0.579         0.863         1.373
                Coordinate di A  Coordinate di B  Coordinate di C
```



Uniamo le tre variabili fra loro e ripassiamo un po' di geometria elementare.



Ogni lato del poligono (in questo esempio tridimensionale, un triangolo) al cui vertice si trovano le variabili è una **distanza euclidea** (**lunghezza** del segmento che separa due punti). Già abbiamo visto (*cosine similarity*) che è data dalla **radice quadrata della somma delle differenze al quadrato:** $d(x,y) = \sqrt{\sum(x_i - y_i)^2}$.

Nel caso delle relazioni tra variabili, la distanza al quadrato $\sum(x_i - y_i)^2$ è **inversamente proporzionale al coefficiente r**; mettendo tutto sotto radice, la **distanza correlativa** $d(x,y)$ è uguale a:

$$dis.corr = \sqrt{2N(1 - r_{XY})}$$

Quindi, nel grafico la distanza maggiore ($\overline{AC}$) corrisponde alla correlazione minore ($r_{AC} = .179$) e la distanza minore ($\overline{AM}$) alla correlazione più forte ($r_{AM} = .973$), con la distanza $\overline{MC}$ intermedia tra le due ($r_{MC} = .402$). In effetti, se $r = 1$, la distanza correlativa è $d = 0$ (i punti sarebbero sovrapposti); se $r = 0$, allora $d = \sqrt{2n}$; se $r = -1$, la distanza è maggiore: $d = 2\sqrt{n}$. Torniamo, nuovamente, al concetto di **prossimità all'identità** di z_X e z_Y (Falk e Well, 1997) di cui abbiamo parlato nel §8.2.1.

Mettiamo alla prova l'equazione:

$d(x,y) = \sqrt{\sum(x_i - y_i)^2}$	`sqrt(sum((za - zm)^2))` `[1] 0.4043579`	`sqrt(sum((za - zc)^2))` `[1] 2.220084`	`sqrt(sum((zm - zc)^2))` `[1] 1.894507`
$d(x,y) = \sqrt{2n(1 - r_{xy})}$	`sqrt(2*3*(1-cor(za,zm)))` `[1] 0.4043579`	`sqrt(2*3*(1-cor(za,zc)))` `[1] 2.220084`	`sqrt(2*3*(1-cor(zm,zc)))` `[1] 1.894507`

[74] Lo ritroveremo nella regressione multipla; per ora, se volete replicarlo, lo script è:
```
scatter3d(x = c(-1.407, -1.402, -.392),y = c(.828,.539,-.981),z = c(.579,.863,1.373), xlab="X", ylab="Y", zlab="Z", point.col="red", axis.scales = TRUE, surface = FALSE)
```

Ora, vi ricordate cosa dice **l'inuguaglianza dei triangoli retti**? Afferma che la somma di due lati qualsiasi di un triangolo deve essere maggiore o uguale alla lunghezza del terzo lato (nei triangoli retti, è una conseguenza del teorema di Pitagora). Quindi, questa **piccola serie di correlazioni** (come matrici di correlazioni ben più grandi) presenta **dei vincoli**, dato che, note due correlazioni, il valore della terza non ne è indipendente. Questa **mancata indipendenza delle correlazioni riferite a uno stesso set di dati ci porta al fondamentale problema del *family-wise error rate***, che affronteremo nel prossimo paragrafo (e anche successivamente, nell'Analisi della varianza).

8.4.2 Family-wise error rate

Il ***family-wise error rate*** è **l'incremento della probabilità di commettere un errore di I tipo in una qualsiasi famiglia di test** quando H_0 è vera in ciascun caso: la "famiglia" di test è il **set di test condotti sullo stesso dataset e che rispondono alla medesima domanda di ricerca**.

Prendiamo come esempio quattro variabili: A, B, C, D. Il pattern delle correlazioni complessive è pari a sei correlazioni bivariate: AB, AC, AD, BC, BD, CD. Per ciascuna di esse, la probabilità di non commettere un errore di I tipo, seguendo le convenzioni usuali, è uguale a .95, **ma** la **probabilità *overall* di non incorrere in un errore di I tipo facendo le sei correlazioni è uguale al prodotto delle singole probabilità** (principio del prodotto), ovvero:

```
.95*.95*.95*.95*.95*.95
[1] 0.7350919
```

Di conseguenza, la probabilità di commettere un errore di I tipo nell'intero set di analisi è oltre cinque volte maggiore del massimo ritenuto accettabile per una singola correlazione:

```
1-.735019
[1] 0.264981
```

Vedremo diversi suggerimenti per correggere il family-wise error nell'analisi della varianza per una X a più di due livelli (capitolo 12); per la correlazione, intanto, possiamo usare la funzioe `rcorr.adjust(matrice)` di `RcmdrMisc`, che fornisce la matrice di correlazioni (di Pearson o di Spearman) con i $p - value$ corretti in funzione del numero di correlazioni nell'analisi, secondo il **metodo di Holm** (1979), la cui logica affronteremo nel Capitolo 12.

Vediamo la matrice di correlazione delle cinque **variabili relative alle modalità di attaccamento** nel dataframe `attaccamento`. Creiamo la matrice in un modo po' diverso; aggiungiamo un cambiamento dei nomi delle variabili, che nel dataframe sono piuttosto lunghi, per agevolare la lettura della matrice di correlazione:

```
asq<-data.frame(fiducioso=a$ASQ_attaccamento_fiducia, evitante=a$ASQ_attaccamento_evitante,  tim
 oroso= a$ASQ_attaccamento_timoroso, distanziante= a$ASQ_attaccamento_distanziante, ambivalente=
 a$ASQ_attaccamento_ansioso_ambivalente)
asq<-as.matrix(asq,40,5)
```

Possiamo **ottenere una matrice composta da soli coefficienti**, **senza il $p - value$**, usando in maniera creativa la funzione `cor`: il suo argomento diventa una **matrice / un dataframe composto dalle sole variabili da correlare**. Arrotondando i decimali per facilitare la lettura, avremo

```
round(cor(asq),3)
             fiducioso evitante timoroso distanziante ambivalente
fiducioso        1.000   -0.192   -0.342       -0.215      -0.031
evitante        -0.192    1.000    0.369        0.621       0.091
timoroso        -0.342    0.369    1.000        0.433       0.382
distanziante    -0.215    0.621    0.433        1.000      -0.021
ambivalente     -0.031    0.091    0.382       -0.021       1.000
```

Può essere anche comodo **ordinare la matrice**, mettendo al primo posto (prima riga) la variabile che presenta la maggior parte di correlazioni rilevanti con le altre, e via via in fondo le variabili con meno correlazioni: salviamo la matrice come oggetto e usiamo la già nota funzione `order`.

```
matr<-round(cor(asq),3)
ord <- order(matr[1, ])
(matr_ord <- matr[ord, ord])
```

```
             timoroso distanziante evitante ambivalente fiducioso
timoroso        1.000        0.433    0.369       0.382    -0.342
distanziante    0.433        1.000    0.621      -0.021    -0.215
evitante        0.369        0.621    1.000       0.091    -0.192
ambivalente     0.382       -0.021    0.091       1.000    -0.031
fiducioso      -0.342       -0.215   -0.192      -0.031     1.000
```

Noi, però, dobbiamo vedere anche i $p - value$. La funzione **`rcorr(matrice, type)`** di **`Hmisc`** lavora su **matrici composte dalle sole variabili da correlare**, come quella appena prodotta: `type=` consente di fare correlazioni parametriche o non parametriche (`"pearson"`, di default, o `"spearman"`). Nell'output è restituita la **matrice delle correlazioni** (con decimali arrotondati a due) più N, e, separatamente, la **matrice delle significatività**, **non corrette** per il family-wise error rate. Vediamolo:

```
rcorr(asq, type = "pearson")
```

```
             fiducioso evitante timoroso distanziante ambivalente
fiducioso         1.00    -0.19    -0.34        -0.21       -0.03
evitante         -0.19     1.00     0.37         0.62        0.09
timoroso         -0.34     0.37     1.00         0.43        0.38
distanziante     -0.21     0.62     0.43         1.00       -0.02
ambivalente      -0.03     0.09     0.38        -0.02        1.00

n= 40


P               fiducioso  evitante  timoroso  distanziante  ambivalente
fiducioso                  0.2346    0.0306    0.1836        0.8476
evitante        0.2346               0.0191    0.0000        0.5773
timoroso        0.0306     0.0191              0.0053        0.0149
distanziante    0.1836     0.0000    0.0053                  0.8971
ambivalente     0.8476     0.5773    0.0149    0.8971
```

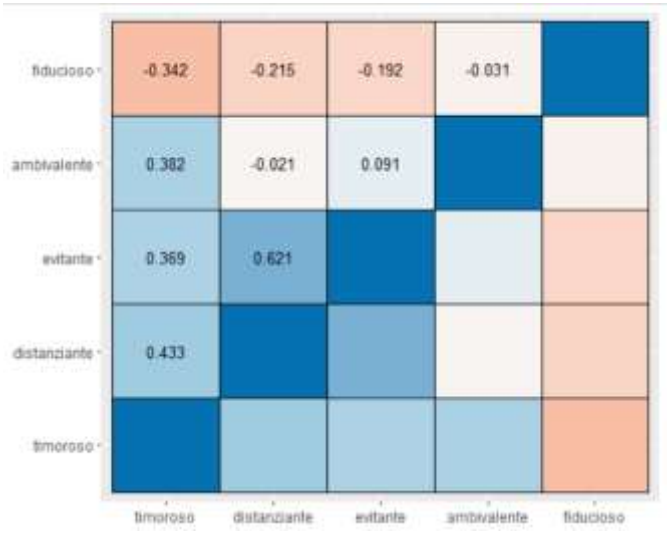
Nella prima matrice dell'output sono contenuti i coefficienti di correlazione. La **diagonale della matrice esprime la correlazione di una variabile con se stessa** (ovviamente $r = 1$) e taglia la matrice stessa **in due metà speculari e ridondanti**: le correlazioni a coppie si possono leggere per riga o per colonna. L'attaccamento fiducioso ha correlazioni negative, come atteso, con le forme di attaccamento meno adattive, ma sono tutte deboli ($-.34$), trascurabili ($-.21$, $-.19$) o proprio inesistenti ($-.03$). L'attaccamento evitante ha una discreta correlazione positiva con quello distanziante ($.62$) e una debole correlazione quello timoroso ($.37$), ma è indipendente dall'attaccamento ambivalente. L'attaccamento timoroso ha correlazioni deboli con tutte le altre dimensioni: negativa con il fiducioso, positiva con le altre ($.37$, $.43$, $.38$). Infine, l'attaccamento ambivalente sembra indipendente dalle altre forme di attaccamento, con la parziale eccezione della debole relazione con quello timoroso, già illustrata.

La seconda matrice dell'output contiene solo i $p - value$ **non corretti** associati a ciascuno dei coefficienti r della matrice precedente: solo le correlazioni con $r > .30$ cadono nella regione di rifiuto di H_0 e non sono quindi casuali, anche se restano interpretativamente deboli, con la sola eccezione della relazione tra attaccamento evitante e distanziante: positiva, di discreta entità e significativa.

Se la grafica vi aiuta nell'interpretazione, inserite il **dataframe** (**non la matrice)** delle variabili da correlare in `sjp.corr(data= dataframe delle variabili da correlare)` di `sjPlot`. Di default, sono calcolati e mostrati

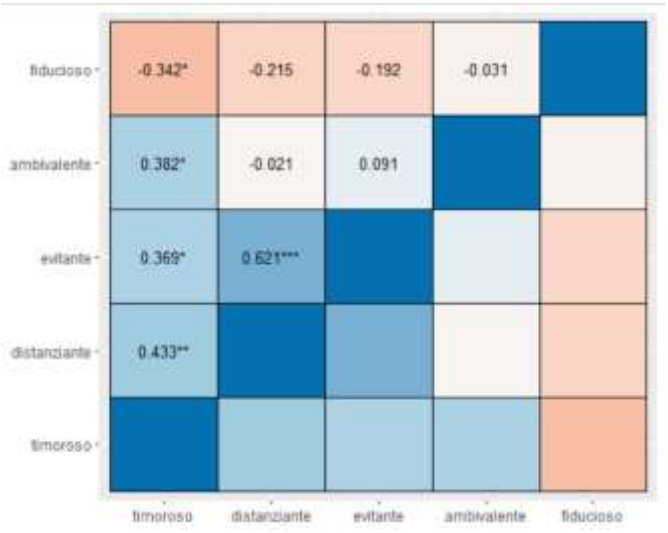
coefficienti di Pearson (`corr.method="pearson / spearman / kendall"`, `show.values=TRUE`) e la loro significatività usando gli asterischi (`show.p= TRUE`): * se il $p-value$ è < 0.05, ** se $< .01$, *** se $< .001$, nessun asterisco se il coefficiente non è significativo. I $p-value$, naturalmente, non sono corretti; quindi, sarebbe opportuno usare questa funzione solo per visualizzare i coefficienti, senza asterischi (`show.p=FALSE`). Le correlazioni sono ordinate per intensità (`sort.corr= TRUE`), e colorate da rosso (negative) a blu (positive), passando per il bianco (prossime a zero).

```
asq<-data.frame(fiducioso=a$ASQ_attaccamento_fiducia, evitante=a$ASQ_attaccamento_evitante,
   timoroso= a$ASQ_attaccamento_timoroso, distanziante= a$ASQ_attaccamento_distanziante,
   ambivalente= a$ASQ_attaccamento_ansioso_ambivalente)
```



Senza *p*-value non corretti

```
sjp.corr(data = asq, show.p = FALSE)
```



Con *p*-value non corretti

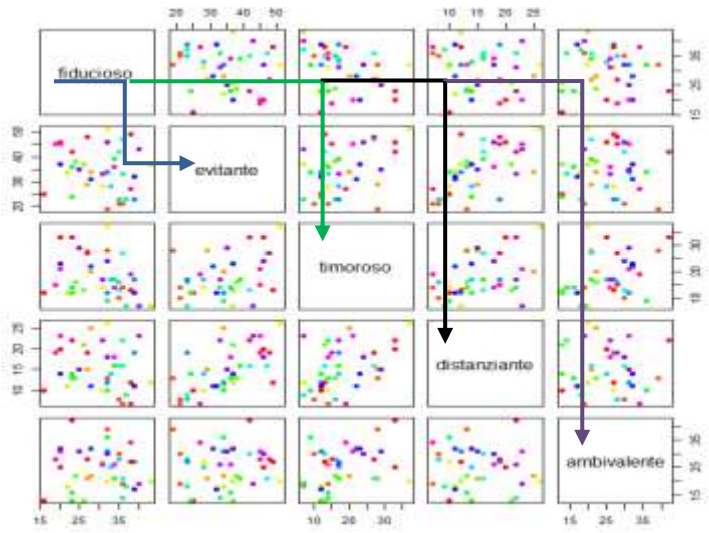
```
sjp.corr(data = asq)
```

Ora vediamo come cambiano i $p-value$ corretti per il *family-wise error rate* con **`rcorr.adjust(matrice)`**:

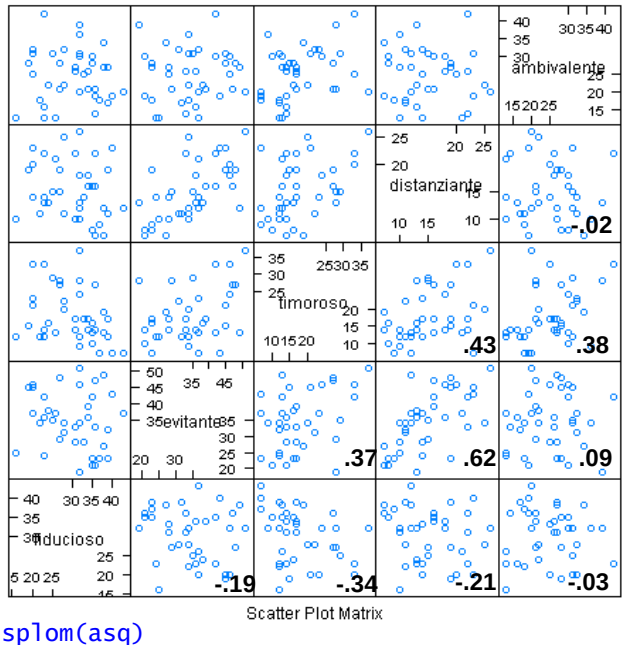
```
rcorr.adjust(asq)
 Pearson correlations:
             fiducioso evitante timoroso distanziante ambivalente
fiducioso       1.0000  -0.1923  -0.3423      -0.2146     -0.0314
evitante       -0.1923   1.0000   0.3691       0.6211      0.0908
timoroso       -0.3423   0.3691   1.0000       0.4330      0.3823
distanziante   -0.2146   0.6211   0.4330       1.0000     -0.0211
ambivalente    -0.0314   0.0908   0.3823      -0.0211      1.0000

Number of observations: 40

Pairwise two-sided p-values:
            fiducioso evitante timoroso distanziante ambivalente
fiducioso              0.2346   0.0306   0.1836       0.8476
evitante    0.2346             0.0191   <.0001       0.5773
timoroso    0.0306    0.0191            0.0053       0.0149
distanziante 0.1836   <.0001   0.0053                 0.8971
ambivalente  0.8476   0.5773   0.0149   0.8971

Adjusted p-values (Holm's method)
            fiducioso evitante timoroso distanziante ambivalente
fiducioso              0.9384   0.1837   0.9182       1.0000
evitante    0.9384             0.1336   0.0002       1.0000
timoroso    0.1837    0.1336            0.0473       0.1194
distanziante 0.9182   0.0002   0.0473                 1.0000
ambivalente  1.0000   1.0000   0.1194   1.0000
```

Tutti i $p-value$ **corretti si alzano**, riducendo la probabilità di commettere un errore di I tipo. Le correlazioni Distanziante-Timoroso e Distanziante-Evitante, più forti, restano significative anche corrette, mentre quelle Timoroso-Fiducioso, Timoroso-Evitante e Ambivalente-Timoroso, più deboli, perdono la significatività.

Per visualizzare tutti gli scatterplot delle correlazioni bivariate della matrice, potete usare **`pairs(matrice)`**, oppure **`splom(matrice o dataframe)`** di `lattice`: entrambe lavorano su oggetti di classe `matrix` o `data.frame` e creano una vera e propria matrice di correlazione, in cui i grafici sostituiscono i coefficienti *r*, anche se, per complicare un po' le cose, tracciano la diagonale che divide la tabella nel senso opposto ☺



`pairs(asq, col=rainbow(15), pch=19, cex=1)`



`splom(asq)`

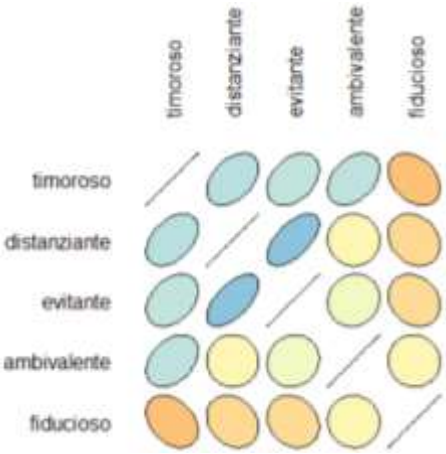
Infine, soprattutto se avete molte variabili da correlare e necessità di un colpo d'occhio essenziale, potete usare `plotcorr(matrice, colori)` di `ellipse`: come suggerisce il nome del package, produce una **matrice di ellissi**, tanto più strette quanto più la correlazione è forte, fino a diventare cerchi per correlazioni tendenti a zero. L'**orientamento** dell'ellisse indica una correlazione **positiva** ↗ o **negativa** ↘. Usare una palette di colori da `RColorBrewer` (`brewer.pal`) rafforza l'impatto dell'informazione: colori divergenti per correlazioni positive e negative, tanto più saturi quanto più *r* tende a |1|. Altrimenti, il grafico si presenta uniformemente grigio.

Per creare la tavolozza dei colori, scegliamo uno dei set di colori **divergenti** di `RColorBrewer`, specificando il numero di sfumature (`n=`) e il tipo di palette ("`RdYlBu`"; per un elenco delle tavolozze, usate l'`help` del package).

```
colori <- brewer.pal(n = 5, "RdYlBu")
colori <- colorRampPalette(colori)(100)
```

E ora il plot: lo applichiamo alla matrice di correlazioni ordinata `matr_ord`, creata qualche pagina fa:

```
plotcorr(corr=matr_ord , col=colori[matr_ord*50+50],
 mar=c(1,1,1,1))
```

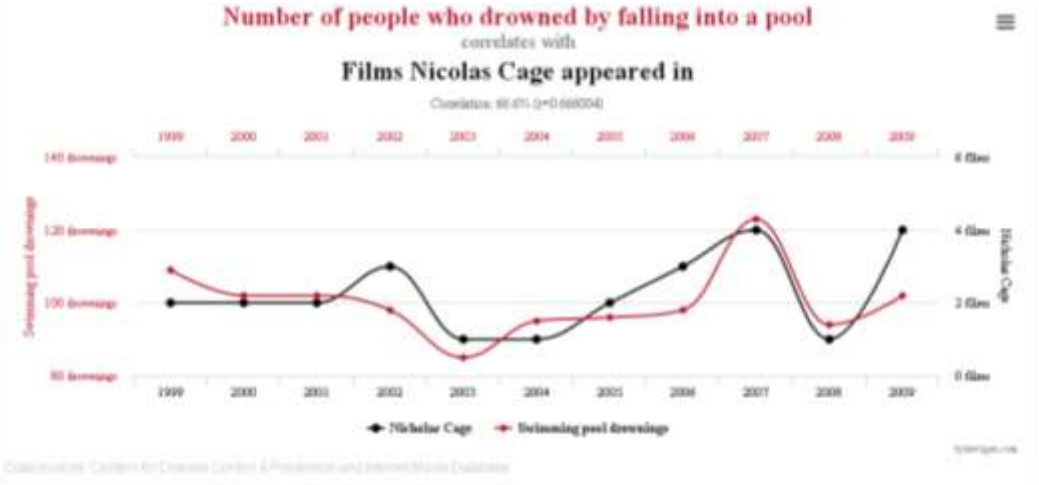


La variabile che presenta più correlazioni rilevanti è Timoroso, positivamente correlata alle altre tranne Fiducioso, con cui ha una discreta correlazione negativa; la correlazione (negativa) più forte è tra Evitante e Distanziante. Le variabili meno associate alle altre sono Ambivalente e Fiducioso.

8.4.3 Crud factor

È noto da tempo il fenomeno che Meehl (1990, pag. 204) chiama senza riguardi "**crud factor**", ovvero fattore sudiciume, spazzatura: "*In the social sciences and arguably in the biological sciences, '**everything correlates to some extent***

***with everything else.*"** Se volete interrompere la noia della statistica con due risate, date un'occhiata a questo (serio) sito: https://tylervigen.com/spurious-correlations, da cui è tratto il seguente grafico di correlazione (vera!)



L'origine della reciproca disistima tra Pearson e Spearman, cui si è accennato a inizio capitolo, è un probabile esempio di *crud factor.* Nelle sue ricerche su eugenetica ed ereditarietà, Pearson aveva raccolto moltissimi dati, tratti dai giudizi di maestri di circa 2000 scuole, in diadi di fratelli, sorelle e coppie fratello-sorella; era interessato a variabili fisiche (salute, colore di occhi e capelli, struttura dei capelli, capacità atletiche, altezza, dimensioni della testa) e psicologiche (assertività, vivacità, popolarità, introspezione, coscienziosità, indole, calligrafia). Le correlazioni medie tra le misure nelle tre diadi erano risultate molto simili, tanto che Pearson scrisse di essere "obbligato, proprio letteralmente obbligato, a trarre la conclusione generale che le caratteristiche fisiche e psichiche nell'uomo sono ereditate, a grandi linee, nella stessa maniera e con la stessa intensità" (1903, pag. 156[75]).

TABLE III.
Inheritance of the Physical Characters.
School Observations on Children.

Character.	Correlation.		
	Brothers.	Sisters.	Brother and Sister.
Health	·52	·51	·57
Eye Colour	·54	·52	·53
Hair ,, ...	·62	·57	·55
Hair Curliness	·50	·52	·52
Cephalic Index	·49	·54	·43
Head Length	·50	·43	·46
Head Breadth...	·59	·62	·54
Head Height	·55	·52	·49
Mean	·54	·53	·51
Athletic Power	·72	·75	·49

TABLE IV.
Inheritance of the Mental Characteristics.
School Observations on Children.

Character.	Correlation.		
	Brothers.	Sisters.	Brother and Sister.
Vivacity	·47	·43	·49
Assertiveness	·53	·44	·52
Introspection	·59	·47	·63
Popularity	·50	·57	·49
Conscientiousness	·59	·64	·63
Temper	·51	·49	·51
Ability	·46	·47	·44
Handwriting	·53	·56	·48
Mean	·52	·51	·52

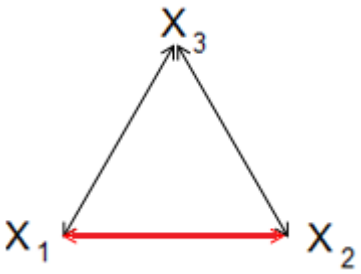
Spearman (1904[76]) scrisse che queste misure erano probabilmente afflitte da "**deviazioni sistematiche**", che i giudizi dei maestri di scuola erano tutt'altro che esenti da errori ed erano sottoposti al grave bias della mancanza di indipendenza, che era scientificamente scorretto considerare "*irrelevant"*, come li definiva Pearson, i molti eventi post natali. Spearman conclude che: "i coefficienti di correlazione [*di Pearson]* devono essere considerati come irrimediabilmente distorti" (pag. 99) e quindi "è difficile evitare la conclusione che le notevole coincidenza annunciata tra l'eredità fisica e mentale può difficilmente essere più che una **semplice coincidenza accidentale**" (pag. 98).

[75] On the laws of inheritance in man: II. On the inheritance of the mental and moral characters in man, and its comparison with inheritance of the physical characters. *Biometrika, 3*(2/3), 131-190.
[76] The proof and measurement of association between two things. *American Journal of Psychology, 15*(1), 72-101.

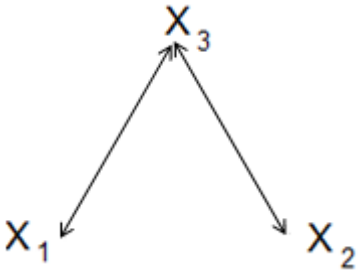
Nonostante Spearman si sia affrettato a precisare che l'oggetto della sua critica non erano le tecniche di correlazione in sé, quanto piuttosto il trarne conclusione affrettate (*ibidem*, pag. 99), Pearson non la prese bene.

8.4.4 Correlazioni parziali o di ordine uno

La mancanza di indipendenza criticata da Spearman si ritrova in molti campi della ricerca correlazionale. Uno dei motivi per diffidare delle relazioni bivariate di ordine zero è che la relazione apparente tra X_1 e X_2, anche se di entità discreta e apprezzabile oltre che significativa, potrebbe essere in realtà **mediata** dalla relazione che entrambe hanno con una terza variabile X_3: eliminando questa variabile (parzializzandone l'effetto), la relazione tra X_1 e X_2 scompare. Relazioni fittizie di questo tipo si definiscono **spurie**, e sono verificate da particolari correlazioni definite **parziali**.



Relazione diretta tra X_1 e X_2



Relazione spuria tra X_1 e X_2

La correlazione **parziale** (o di **ordine uno**) verifica la relazione esistente tra X_1 e X_2, una volta **controllato** (parzializzato) **l'effetto della relazione tra** X_3 ed **entrambe** $\boldsymbol{X_1}$ **e** $\boldsymbol{X_2}$[77]. Nella sua formula si inserisce in genere il coefficiente r di Pearson, ma può essere applicata anche al coefficiente rho:

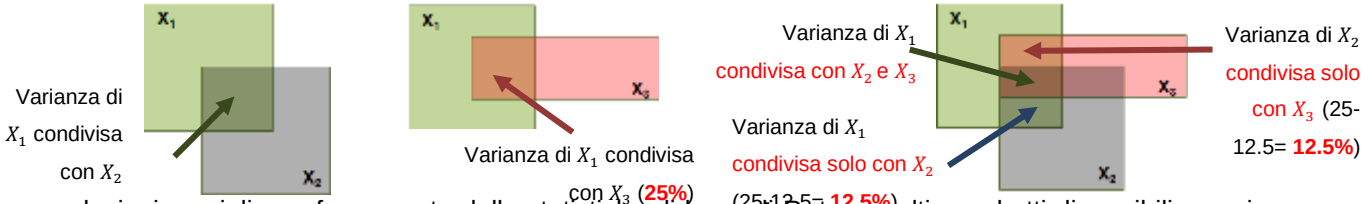
$$r_{12.3} = \frac{r_{12} - 3r_{13}r_{23}}{\sqrt{(1 - r_{13}^2)(1 - r_{23}^2)}}$$

Usiamo un esempio romantico. Diciamo che la relazione sentimentale tra Giulietta (X_1) e Romeo (X_2) è operazionalizzabile e quantificabile con un coefficiente di correlazione $r = .750$: la relazione tra loro (**di ordine zero**) sembra positiva e di più che discreta entità. Tuttavia, la coppia vive con la suocera (X_3): qual **è l'effetto della relazione che la suocera ha con entrambi sulla relazione tra Giulietta e Romeo?** Possiamo verificarlo **traslocando la suocera** (ovvero, **parzializzando l'effetto di X_3**) e verificando nuovamente la relazione nella coppietta. Si aprono così tre scenari:

a) la relazione è quantificabile con un coefficiente $r = .750$, quindi è **immutata**: la relazione di Giulietta e Romeo con la suocera non aveva alcun effetto sulla relazione diretta tra loro;
b) la relazione è quantificabile con un coefficiente $r = .900$, quindi è **aumentata**: la suocera aveva un effetto **inibitorio** sulla relazione diretta tra Giulietta e Romeo;
c) la relazione è quantificabile con un coefficiente $r = .225$, quindi è **diminuita**: la presenza della suocera X_3 ha **simulato** una relazione diretta tra Giulietta e Romeo in realtà inesistente, o inferiore a quanto rilevato nella correlazione di ordine zero (insomma, i due stavano insieme solo per far contenta la suocera).

In termini meno romantici, questo significa che parte della varianza di X_1 spiegata dalla varianza di X_1 ($covarianza_{X_1X_2}$) **non è unica**: può essere, infatti, spiegata da una terza variabile X_3; la **correlazione tra due variabili in cui gli effetti di una terza variabile sono tenuti costanti** è la nostra correlazione **parziale**.

[77] Le correlazioni **semi-parziali** controllano l'effetto della relazione tra X_3 e **una** tra X_1 o X_2.



Le correlazioni parziali non fanno parte delle statistiche di base di R, tra i molti pacchetti disponibili, possiamo usare **ppcor** che contiene **pcor.test(X1, X2, X3, method=)**; in **method=** specifichiamo **"pearson"** (default), **"spearman"** o **"kendall"**. Oltre al coefficiente di correlazione parziale (e semi-parziale, che non è nel nostro programma), la funzione fornisce anche una statistica ricavata dal coefficiente e il relativo $p - value$, basato su una distribuzione t (Kim, 2015).

Usiamo come esempio il dataframe **disforia**, che contiene, tra le molte variabili, misure di ansia di stato, ansia di tratto e depressione, costrutti notoriamente intrecciati tra loro, rilevate in soggetti clinici e non clinici. Le correlazioni di ordine zero tra ansia di stato, ansia di tratto e depressione sono positive e di discreta entità:

```
cor.test(disforia$BDI, disforia$STAI_stato)
   Pearson's product-moment correlation
data:  disforia$BDI and disforia$STAI_stato
t = 11.233, df = 152, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5767839 0.7516029
sample estimates:
      cor
0.6735044
```

```
cor.test(disforia$BDI, disforia$STAI_tratto)
   Pearson's product-moment correlation
data:  disforia$BDI and disforia$STAI_tratto
t = 13.197, df = 152, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6473988 0.7968076
sample estimates:
      cor
0.7307371
```

Ora verifichiamo le correlazioni parziali. Per prima cosa, dobbiamo selezionare solo i soggetti con dati completi, perché pcor.test non ammette NA:

```
bdi<-disforia$BDI[is.na(disforia$BDI)==FALSE]
stai_stato<-disforia$STAI_stato[is.na(disforia$STAI_stato)==FALSE]
stai_tratto<-disforia$STAI_tratto[is.na(disforia$STAI_tratto)==FALSE]
```

Cominciamo con la correlazione parziale tra depressione e ansia di stato, controllando l'effetto dell'ansia di tratto; possiamo mettere alla prova la formula:

```
(cor(bdi, stai_stato) - (cor(bdi,stai_tratto) * cor(stai_stato, stai_tratto))) / sqrt((1-(cor(bdi,
stai_tratto)^2)) * (1-(cor(stai_stato, stai_tratto)^2)))
[1] 0.1692222
```

ma sarà molto più veloce la funzione:

```
pcor.test(bdi, stai_stato,stai_tratto)
     estimate    p.value  statistic   n  gp   Method
[1] 0.1692222  0.03590179   2.11684  155   1  pearson
```

Una volta controllato l'effetto dell'ansia di tratto, la **correlazione tra ansia di stato e depressione sparisce** (l'entità è trascurabile, la varianza condivisa è $< 3\%: r = .17^2$), e la significatività si mantiene solo a causa del gran numero di

osservazioni. Quindi, questa correlazione parziale deve essere considerata una misura più realistica della relazione tra le due variabili, la cui apparente relazione diretta è probabilmente mediata in gran misura dalla predisposizione all'ansia. Vediamo cosa succede alla relazione tra depressione e ansia di tratto, controllando l'effetto dell'ansia di stato:

```
pcor.test(bdi, stai_tratto,stai_stato)
     Estimate      p.value  statistic   n  gp   Method
[1] 0.4159374 8.108886e-08  5.638947  155   1  pearson
```

L'entità della correlazione diminuisce, ma **resta comunque apprezzabile**: quindi, anche se l'ansia di stato contribuisce a "gonfiare" la relazione diretta tra depressione e ansia di tratto, la parzializzazione del suo effetto ha un impatto decisamente minore rispetto a quello verificato nella precedente correlazione parziale.

*All'inizio del capitolo abbiamo disegnato il bubbleplot della relazione tra ansia di stato e depressione, in funzione del punteggio di ansia di tratto, nel dataframe `attaccamento`: calcolate le correlazioni parziali tra queste variabili, e verificate cosa accade: quali inferenze **cliniche** possiamo trarre sui costrutti di ansia e depressione, alla luce di quanto avrete ottenuto?*

8.5 Linearmente indipendenti, ortogonali o non correlati?

Concludiamo il capitolo sulle correlazioni con una precisazione, non solo terminologica, su concetti che troveremo ripetutamente nei prossimi capitoli.

Può capitare di parlare di coppie di variabili come "linearmente indipendenti" o "ortogonali" o "non correlate" usando queste tre espressioni come se fossero sinonimi. In effetti, però, **indipendenza lineare**, **ortogonalità** e **correlazione = 0** sono concetti sottilmente diversi e facilmente confusi: in realtà, distribuzioni ortogonali possono essere indifferentemente correlate o non correlate (Rogers, Nicewander e Toothacker, 1984).

La prima macroscopica differenza fra i termini riguarda le distribuzioni cui si riferiscono:

- l'indipendenza lineare e l'ortogonalità delle distribuzioni X e Y riguardano i dati grezzi,
- la correlazione si riferisce alle distribuzioni X_C e Y_C centrate (sulla media) della distribuzione.

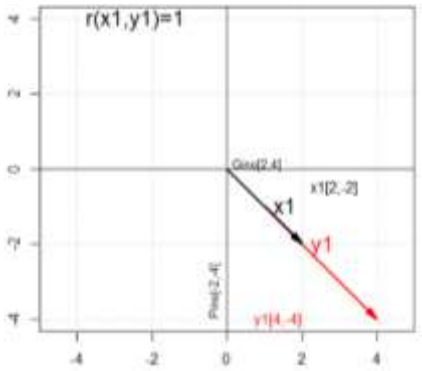
Aiutiamoci con la grafica; dato che faremo riferimento al *person space*, useremo il termine **vettore** per intendere la distribuzione. Rappresenteremo grafici in uno spazio **bidimensionale**, cioè distribuzioni bivariate ottenute da **due soli soggetti**, per gli evidenti limiti percettivi umani: va da sé che la matematica sottostante si applica a *person space* n-dimensionali.

Le **variabili linearmente indipendenti** sono quei **vettori che non cadono sulla stessa linea**: **non** esiste una costante moltiplicativa λ (lambda: un valore scalare, ovvero un qualsiasi $numero\ reale \neq 0$) che espande, contrae o riflette un vettore nell'altro. L'equazione che corrisponde a questa proprietà è: $\lambda X - Y \neq 0$ (in pratica, l'unica combinazione lineare che darebbe un vettore nullo $V(0,0)$ consisterebbe nel moltiplicare i vettori per $\lambda = 0$). All'opposto, le variabili **linearmente dipendenti** cadono sulla **stessa linea**: un vettore è una **funzione lineare** dell'altro: $X = \lambda Y$, ovvero $\lambda X - Y = 0$.[78]

[78] Individuare "a occhio" la costante moltiplicativa si può fare con due vettori, che è il nostro caso; se entriamo in uno spazio n-dimensionale, l'unico modo per definire la dipendenza lineare è risolvere con un sistema di equazioni $\lambda_1V_1+\lambda_2V_2+\lambda_3V_3+\ldots\lambda_nV_n=0$, ma non dovremo preoccuparcene. Tre o più vettori linearmente dipendenti si definiscono **complanari** (stanno sullo stesso piano).

Per esempio, riprendiamo i test X_1 e Y_1 del § 8.4.1 e somministriamoli ancora ai due soggetti: Gino ottiene $X_1 = 2$ e $Y_1 = 4$, Pino invece $X_1 = -2$ e $Y_1 = -4$. Quindi, la variabile / il vettore X_1 è $\boldsymbol{X_1[2,-2]}$ e la variabile Y_1 è $\boldsymbol{X_1[4,-4]}$. Rappresentiamole (per lo script, §48.4.1).

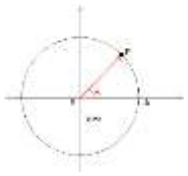
Le due variabili sono **linearmente dipendenti**: moltiplicando i punteggi di X_1 per 2 (**costante moltiplicativa**) si ottengono i punteggi di Y_1, cioè $\lambda X - Y = 0$.



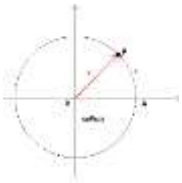

```
x1=c(2,-2); y1=c(4,-4)
x1*2
[1] 4 -4
(2*x1)-y1
[1] 0 0
```

Due vettori linearmente dipendenti sono perfettamente **correlati**. Nel caso di una correlazione **positiva** perfetta, come in questo esempio, il coseno del loro angolo (0°) è $\boldsymbol{cos = 1}$:

```
cor(X1,Y1)
[1] 1
cos(0*pi/180)
[1] 1
```

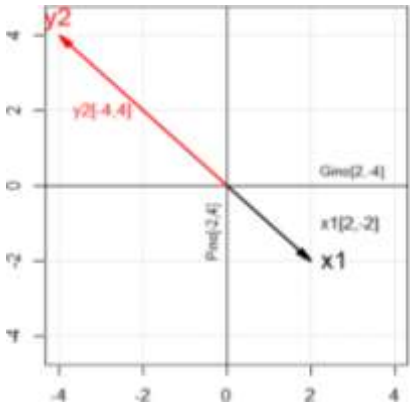


*[Per esprimere l'ampiezza degli angoli, R usa i radianti, secondo il sistema internazionale di misura. Un radiante (rad) è il rapporto tra la **lunghezza dell'arco** orientato di circonferenza centrata in O dell'angolo (**l**) e la lunghezza del raggio di questa circonferenza (r). infatti, la posizione di un punto p rispetto al sistema XY può essere espressa o come angolo tra raggio della circonferenza e X, o come rapporto tra* l *e raggio r. Per la conversione grado→radiante, si moltiplica il grado per π/180; per quella radiante→ grado, si moltiplica il radiante per 180/π].*

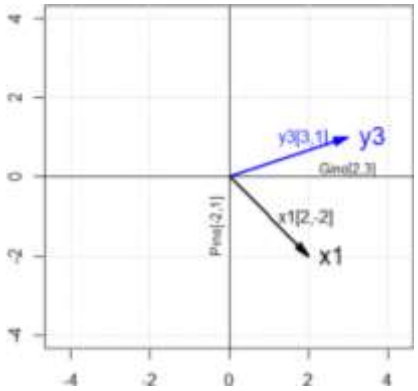


Nel caso di una correlazione **negativa perfetta**, il coseno di due vettori linearmente dipendenti (180°) **è** $\boldsymbol{cos = -1}$.

Per la stessa $X_1[2,-2]$ poniamo una $Y_2[-4,4]$: i vettori X_1 e Y_2 sono linearmente dipendenti (stessa linea, $\boldsymbol{\lambda = -2}$) e perfettamente correlati: stessa direzione, verso **opposto**:



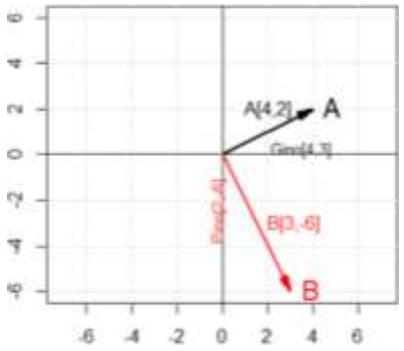

```
x1=c(2,-2); y2=c(-4,4)
x1*-2
[1] -4 4
(-2*x1)-y2
[1] 0 0
cor(x1, y2)
[1] -1
cos(180*pi/180)
[1] -1
```



Nella variabile Y_3, invece, Gino ottiene $Y_3 = 3$ e Pino $Y_3 = 1$. Quindi, la variabile / vettore Y_3 è $\boldsymbol{Y_3[3,1]}$. Rappresentiamo la distribuzione bivariata $\boldsymbol{X_1[2,-2]}$ e $\boldsymbol{Y_3[3,1]}$. Le due variabili **sono linearmente indipendenti**, dato che non giacciono sulla stessa linea: **non è possibile trovare una costante** che trasformi X_1 in Y_3.

Le **variabili ortogonali** sono un **caso particolare** di variabili **linearmente indipendenti**: NON cadono sulla stessa linea e **formano un angolo retto l'una rispetto all'altra** (in uno spazio *n*-dimensionale), cioè il loro coseno è $\boldsymbol{cos = 0}$. Quindi, tutte le variabili ortogonali sono linearmente indipendenti, ma non tutte le variabili linearmente indipendenti sono ortogonali.

Riprendiamo l'esempio del §8.4.1: Nelle due variabili A e B, Gino ottiene $A = 4$ e $B = 3$, Pino invece ottiene $A = 2$ e $B = -6$. Quindi, la variabile / vettore A è $\boldsymbol{A[4,2]}$ e la variabile B è $\boldsymbol{B[3,-6]}$.



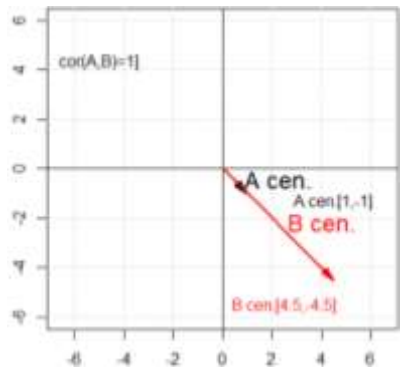
A e B sono linearmente indipendenti **e** formano un angolo a 90° → il loro coseno è $cos = 0$.

Abbiamo già visto che il **prodotto interno** di questi vettori: quando è $= 0$, i vettori sono ortogonali.

Nel nostro caso, Infatti: $\begin{bmatrix}4\\2\end{bmatrix} \times \begin{bmatrix}3\\-6\end{bmatrix} = (4 \times 3) + (2 \times -6) = 12 - 12 = \mathbf{0}$

Due variabili sono **non correlate** quando, **una volta che sono state centrate attorno alla media,** i loro vettori sono **perpendicolari**; ne consegue che due variabili sono tanto più perfettamente **correlate** quanto più, una volta centrate, il **loro angolo tende a 0** (se la correlazione è positiva; a 180° se negativa).

Continuiamo a usare le variabili A e B; invece di usarle grezze, centriamo ciascuna attorno alla sua media e raffiguriamone i vettori:




```
A=c(4,2); B=c(3,-6)
A_cen=A-mean(A); B_cen=B-mean(B)
A_cen; B_cen
[1]  1 -1
[1]  4.5 -4.5
```

Le due variabili grezze sono perfettamente correlate, e l'angolo tra i vettori centrati è $= 0$.

Quindi, **una volta centrati, i vettori non sono più ortogonali**; il loro prodotto interno, infatti, non è $= 0$, ma: $\begin{bmatrix}1\\-1\end{bmatrix} \times \begin{bmatrix}4.5\\-4.5\end{bmatrix} = (1 \times 4.5) + (-1 \times -4.5) = \mathbf{9}$

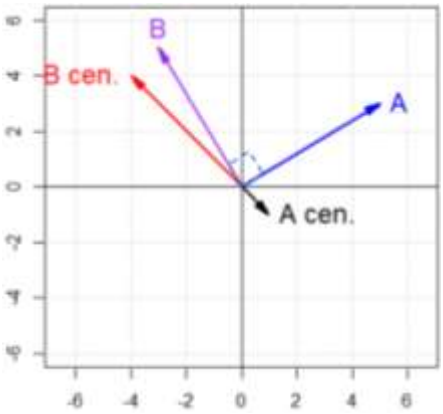
Sapere che due variabili sono correlate / non sono correlate non dà informazioni sulla disposizione nello spazio dei rispettivi vettori grezzi: **centrare una variabile potrà cambiare** (e lo farà spesso e volentieri) l'angolo tra i vettori.

Generalizziamo con un altro esempio, usando le variabili $A[5,3]$ e $B[-3,5]$:

```
A=c(5,3); B=c(-3,5)
A_cen=A-mean(A); B_cen=B-mean(B)
```

A e B sono ortogonali; $\begin{bmatrix}5\\3\end{bmatrix} \times \begin{bmatrix}-3\\5\end{bmatrix} = (5 \times -3) + (3 \times 5) = -15 + 15 = \mathbf{0}$

```
A_cen; B_cen
[1]  1 -1
[1]  -4 4
```



Una volta centrate, l'angolo tra A e B non è più $= 90°$, ma $= 180°$:

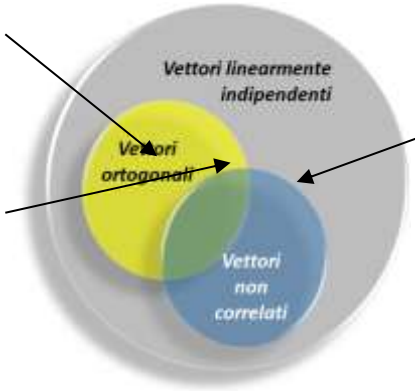
$\begin{bmatrix}1\\-4\end{bmatrix} \times \begin{bmatrix}-1\\4\end{bmatrix} = (1 \times -4) + (-1 \times 4) = -4 - 4 = \mathbf{-8}$

```
cor(A,B)
[1] -1
cos(180*pi/180)
[1] -1
```

Le due variabili mostrano una perfetta correlazione negativa.

1. Possiamo, quindi, verificare che quando le due variabili sono **linearmente indipendenti**: ue variabili grezze perpendicolari possono diventare oblique, una volta centrate: sono **ortogonali e correlate**.

2. Due variabili possono essere **ortogonali e non correlate**, se centrarle non cambia l'angolo tra i loro vettori.



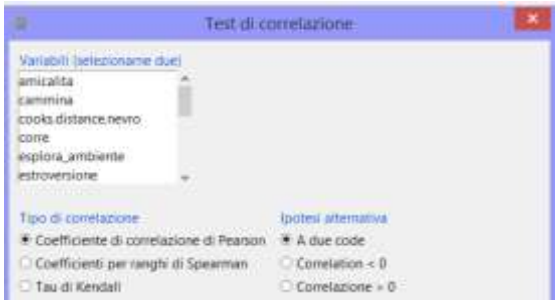
3. due variabili grezze oblique (non perpendicolari) possono diventare perpendicolari, una volta centrate: **non** sono **ortogonali e non** sono **correlate**.

Modificata da Rodgers, Nicewander e Toothaker, 1984

8.6 La correlazione lineare con `RCommander`

Fare una correlazione è così semplice che `RCommander` non aggiunge un gran vantaggio.

Comunque, una volta importato il dataframe i test di correlazione si trovano nel menu `Statistiche` → `Informazioni riassuntive` → `test di correlazione`:



Una volta selezionate le due variabili, si sceglie il tipo di coefficiente desiderato e la direzione di H_1.

Se avete molte variabili da correlare a coppie, il menu `Statistiche` → `Informazioni riassuntive` → `matrice di correlazione` produce la matrice di correlazione e relativi p-value, usando la funzione `rcorr` che abbiamo già visto; effettivamente, questo vi risparmia di produrre la matrice con le sole variabili da correlare, dato che ci pensa `RCommander`:

Selezionate "$p - values$ a coppie", se ne volete i valori. L'utilizzo delle osservazioni complete o delle osservazioni complete a coppie gestisce l'approccio listwise o pairwise ai dati mancanti.

Il grafico a dispersione è prodotto dal menu `Grafici` → `Grafico a dispersione`. La produzione di una matrice di grafici a dispersione per rappresentare più correlazioni contemporaneamente è `gestita da Grafici` → `matrice di grafici a dispersione` (corrisponde alle funzioni `pairs` o `splom`).

Capitolo 9
Regressione lineare semplice

In questo capitolo useremo il dataframe ***perseverazioni*** *pubblicato su Elly; prima di proseguire con la lettura, eseguite:*
1. aprite il dataframe e leggetene la descrizione nel file .txt "descrizione perseverazioni";
2. rinominatelo come ***p****.*

Dobbiamo l'evoluzione moderna dell'analisi di regressione a Galton.
L'analisi della regressione lineare semplice consente di ricavare dai dati campionari un **modello statistico** che **predica** i valori della variabile Y (***criterio*** *o* ***dipendente***) a partire dai valori di un'altra variabile X (***predittore*** o ***indipendente***): noto il valore in X_i del soggetto i, e nota la relazione che lega X e Y nella popolazione di appartenenza del soggetto i, il **modello di regressione** consente di **stimare**, con una certa quota di **errore** casuale, il valore in Y_i del soggetto. Ovvero: se sappiamo quanto è alto (x_i) Gino, che appartiene a una popolazione di giovani adulti, e conosciamo empiricamente quale sia la funzione che lega altezza e peso nei giovani adulti, potremo predire, con una approssimazione che cercheremo di rendere minima, quale sia il peso (y_i) di Gino.
La **relazione di predizione** trattata dall'analisi di regressione è stata tradizionalmente intesa come **relazione causale** $\boldsymbol{Y \sim X}$ (Y in funzione di X: il variare di X causa / determina, ovvero influenza, il variare di Y-effetto). In realtà, la relazione causale è solo **uno** dei tipi di relazione più genericamente predittiva. Semplificando molto i cinque principi di J. S. Mill (1843[79]), una relazione causale esiste se:

1) la causa è diversa dall'effetto: se non fossero entità distinte, la loro relazione sarebbe tautologica;
2) la causa ha preceduto l'effetto: è necessario verificare un precedente temporale nella relazione;
3) la causa è correlata all'effetto: la covarianza tra causa ed effetto deve essere dimostrata, anche se si privilegia una lettura più probabilistica che rigorosamente deterministica: la causa **aumenta la probabilità di verificarsi** (la **verosimiglianza**) dell'effetto, anche se non ne determina incondizionatamente la comparsa;
4) non si possono trovare spiegazioni alternative plausibili per l'effetto che siano diverse dalla causa: questa è evidentemente la condizione più complessa da dimostrare, a causa della miriade di covariate che possono influenzare (o simulare!) la relazione tra causa ed effetto.

Queste condizioni si possono verificare in **esperimenti**, in cui il ricercatore **manipola** la presunta causa e osserva l'effetto, valuta se la variazione nella causa è legata alla variazione nell'effetto, usa diversi metodi prima e durante l'esperimento per ridurre la plausibilità di spiegazioni alternative per l'effetto, nonché metodi statistici dopo l'esperimento per isolare l'effetto delle variabili che non può controllare. **Non si possono invece verificare in disegni correlazionali**, in cui il ricercatore non ha il controllo delle variabili, che si limita a rilevare. In questi casi, anche se genetica e/o buon senso possono orientare l'interpretazione della direzione della causalità tra X e Y, la regressione può al più qualificare e quantificare una **relazione di predittività** tra le due variabili, ma può individuare la causalità dell'una sull'altra.
Quando sia X sia Y sono variabili metriche **continue**, la relazione più **semplice** che possiamo ipotizzare tra esse è quella **lineare**: al variare di un'unità in X, $\boldsymbol{Y}$ **varia di una quantità costante per tutti i valori di** $\boldsymbol{X}$**.** Non è naturalmente l'unico tipo di relazione di regressione possibile tra due variabili (vedremo la **regressione logistica** nel capitolo 14) e

[79] *A system of Logic*, Vol.1: metodo diretto di concordanza, metodo della differenza, metodo della concordanza e della differenza, metodo del residuo, metodo della variazione concomitante.

certamente non sarà sufficiente una sola X a spiegare la gran parte del cambiamento di Y: se aggiungiamo altre X al modello, entriamo nel campo della **regressione lineare multipla** (capitolo 11).
Nella regressione semplice, $\boldsymbol{H_0}$ è che in popolazione X e Y siano **indipendenti** → tra X e Y **non esiste** una relazione di predittività: conoscendo il valore x_i, **non** possiamo predire quale sarà il corrispettivo valore x_i; al variare dell'una, l'altra variabile varia in maniera del tutto indipendente. $\boldsymbol{H_1}$ è che in popolazione X e Y siano **dipendenti** → tra X e Y esiste una **relazione di predittività** → al variare dell'una (predittore X), l'altra (criterio Y) varia in maniera predicibile: **conoscendo il valore x_i, possiamo predire quale sarà il corrispettivo valore y_i, con un margine di errore minimo (il più piccolo possibile).**

9.1 I parametri del modello

Abbiamo già visto le equazioni:

- $\boldsymbol{dato\ reale_i = (modello) + errore_i}$: l'osservazione empirica è data dal modello statistico più la quota di errore del modello per quella osservazione;
- $\boldsymbol{devianza = \sum(dato - modello)^2}$: il fit del modello è valutato dalla quantità di scarti del modello dai dati reali, cioè dalla somma dei suoi errori.

Nei capitoli precedenti, il modello statistico cui ci riferivamo, per descrivere la tendenza centrale di una distribuzione, era la media; qui, il modello statistico per descrivere la relazione bivariata tra X e Y è il **modello lineare generale**:

$$\boldsymbol{dato\ reale_{Y_i} = (modello\ lineare) + errore_{Y_i}}$$

Secondo il modello lineare, il modo migliore per rappresentare sinteticamente e in maniera affidabile la relazione lineare tra due distribuzioni X e Y è una **retta**: cerchiamo quindi di **adattare** un **modello lineare** alla distribuzione bivariata, ovvero di **costruire una retta** che **sintetizzi al meglio** la relazione dei dati: questa **retta conterrà tutti i dati previsti dal modello lineare per la distribuzione Y a partire dai corrispondenti valori della distribuzione X**. La **funzione** di **regressione** che esprime matematicamente la relazione tra X e Y è la base del **modello lineare generale (general linear model – GLM, o linear model, LM,** che occuperà le nostre lezioni (e i vostri incubi) ogni volta che postuleremo l'esistenza di una relazione tra (almeno) una variabile indipendente X e (almeno) una variabile dipendente Y.
Il termine dell'equazione "*(modello lineare)*" va sostituito con "cose che definiscono il tipo di retta che dobbiamo adattare ai dati": qualsiasi retta viene definita, ovvero costruita, da due **parametri** chiamati coefficienti di regressione:

- b_1 [β_1 *se si riferisce alla popolazione*] è il **coefficiente angolare**: esprime la **pendenza** (**slope**) della retta e indica la **quantità unitaria di cui cambia Y al variare di una unità di X**;
- b_0 [β_0 *se si riferisce alla popolazione*] è l'**intercetta** della retta di regressione, ovvero il **valore atteso di Y quando $X = 0$** (graficamente, corrisponde al punto in cui la rotta tocca l'ordinata).

L'equazione del modello lineare è quindi:

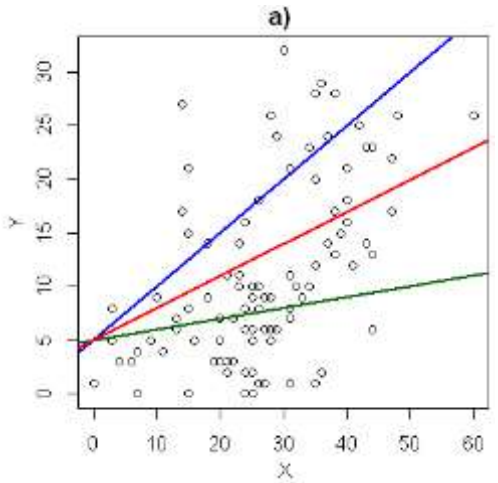
$$\boldsymbol{dato\ reale_{Y_i} = (modello\ lineare) + errore_{Y_i}}$$

$$\boldsymbol{dato\ reale_{Y_i} = (\beta_0 + \beta_1 X_i) + errore_{Y_i}}$$

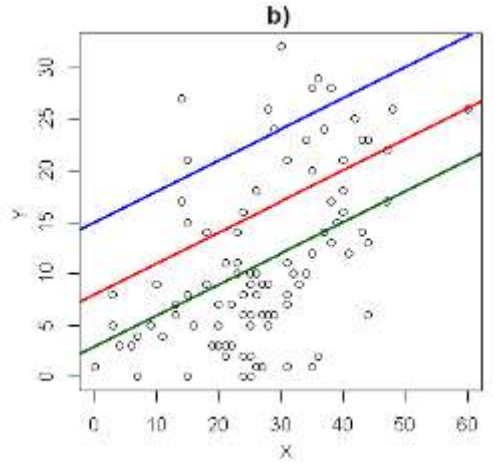
, in cui X_i è il dato campionario in X dell'osservazione i, ed $errore_{Y_i}$ è la distanza tra il valore in Y di i predetto dal modello ($\hat{y}_I$) e quello realmente osservato (y_i). I **valori predetti** dal modello si chiamano genericamente "**y hat**": siccome gli statistici sono gente spiritosa, l'accento circonflesso su $\hat{y}_i$ che distingue un valore predetto dal corrispettivo y_i rilevato nel campione sarebbe, infatti, un "cappellino" (*hat*).

La storia dell'analisi di regressione e del metodo con cui sono calcolati i parametri del modello lineare è tracciata nell'Appendice III, compresi i dati originali di Galton.

I due parametri b_0 e b_1 del modello lineare sono **indipendenti** l'uno dell'altro, e devono essere calcolati dai dati. Nelle figure sottostanti, potete osservare esempi a) di rette con uguale intercetta e diverso coefficiente angolare e b) con diversa intercetta e uguale coefficiente angolare. È rappresentata la relazione "depressione (BDI) in funzione del rimuginio depressivo (PTQ)". Per sovrapporre le rette al grafico si usa **abline(a=, b=)**, già vista: ora possiamo svelare che l' "**ab**" della funzione sono rispettivamente l'argomento intercetta (**a=**) e coefficiente angolare (**b=**) della retta che sarà tracciata nel grafico. Possiamo tracciare un infinito numero di rette variando i due parametri: si tratterà di **individuare la migliore combinazione di parametri**, cioè quella che **determina il modello lineare migliore**.




```
plot(p$PTQ_totale, p$BDI_totale, xlab="X",
  ylab="Y", main="a")
abline(a=5,b=.1, col="dark green", lwd=2)
abline(a=5,b=.5, col="blue", lwd=2)
abline(a=5,b=.3, col="red", lwd=2)
```

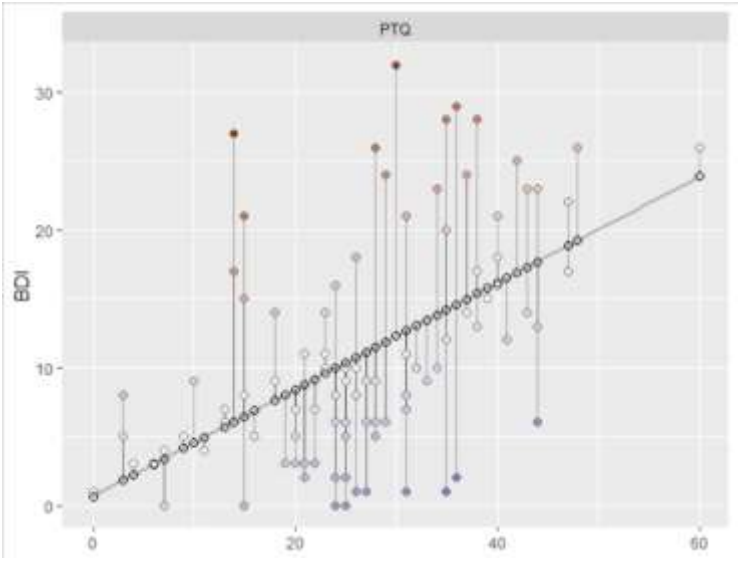



```
plot(p$PTQ_totale, p$BDI_totale, xlab="X",
  ylab="Y", main="a")
abline(a=3,b=.3, col="dark green", lwd=2)
abline(a=8,b=.3, col="red", lwd=2)
abline(a=15,b=.3, col="blue", lwd=2)
```

Ed ecco la retta che **rappresenta la migliore soluzione**, secondo il metodo dei minimi quadrati che ora vedremo in dettaglio, **per rappresentare la relazione BDI~PTQ**. Anche in questo modello ottimale, la **distanza** che separa ogni y_i osservato dal rispettivo $\hat{y}_i$ sulla retta, che rappresenta lo **sbaglio** commesso dal modello lineare–retta nella previsione, non è trascurabile per parecchi y_1.

Il grafico è stato ottenuto con `plot_residuals(fit= modello)` di `sjPlot`; l'argomento `geom.size=` gestisce la dimensione degli elementi del grafico (di default, `geom.size=2`); `show.pred=TRUE` (default) mostra i valori predetti dal modello, giacenti sulla retta; `show.resid=TRUE` (default) mostra gli errori del modello (i dati osservati); `show.lines=TRUE` (default) mostra la connessione tra valori predetti sulla retta e il rispettivo errore:

```
plot_residuals(fit = lm(p$BDI_totale ~ p$PTQ_totale), geom.size = 4)
```



Formalizziamo. L'errore $e_i = y_i - \hat{y}_i$ rappresenta lo **scarto verticale** del valore campionario, secondo l'equazione del modello, quindi:

$$y_i = (\beta_0 + \beta_1 X_i) + e_i \rightarrow \boldsymbol{e_{yi} = y_i - (\beta_0 + \beta_1 X_i)}$$

Infatti, abbiamo già visto nel modello–media che l'adattamento di un modello ai dati è stimato dalla differenza tra il modello e i dati realmente osservati: nella regressione, queste **distanze** o **scarti** o **errori** tra dati osservati e dati predetti dal modello sono chiamati **residui**, perché rappresentano tutto quello che il modello – retta **non è in grado di spiegare** dei dati reali. Avremo quindi residui / scarti positivi (sopra la retta-modello) e residui / scarti negativi (sotto la retta-modello): la **loro somma** (al quadrato, per superare il problema del reciproco annullamento) **rappresenta la distanza complessiva**, ovvero la **devianza residua – SS_R (residual sum of squares)**, tra i dati predetti dal modello e i dati reali. La SS_R è un indice di **goodness of fit** del modello: tanto **più piccola è**, **tanto migliore è il modello**.
Perciò, adotteremo come criterio per la costruzione del modello lineare / retta quello per cui esso deve rappresentare la migliore soluzione possibile nel **ridurre** la **distanza (residuo) tra ogni osservazione y e il corrispettivo punto stimato** $\hat{y}$ giacente sulla retta, ovvero deve **ridurre al massimo** l'**errore**.

Per calcolare β_0 e β_1 di una retta che rispetti il criterio della massima riduzione della quota d'errore, ci sono due metodi:

- ✓ il metodo dei **minimi prodotti** (*least products*): non fa parte del nostro programma, ma tenetelo presente; si applica se non sono rispettati i requisiti di applicabilità del metodo dei minimi quadrati (è un metodo **robusto**);
- ✓ Il **metodo dei minimi quadrati** (***ordinary least squares*** - **OLS**): è quello che useremo. Il principio dei minimi quadrati afferma che la **somma delle differenze al quadrato delle osservazioni dalla media è un minimo**, ovvero è **più piccola di qualsiasi altra somma delle differenze al quadrato da qualsiasi altro punto di riferimento**. L'enunciazione del principio è correntemente attribuita a Gauss (1809), anche se Legendre ne ha (abbastanza inutilmente) rivendicato la priorità (1805). Usare la media per combinare una serie di osservazioni indipendenti era una tecnica usata già dal XVI secolo: Gauss si è occupato del problema di selezionare, tra diversi modi alternativi per combinare dati, quello che producesse la minore incertezza possibile rispetto al "valore vero", quello in popolazione.

Nel metodo OLS, i parametri del modello si calcolano come segue:

- Il **coefficiente angolare** è dato dal **rapporto tra codevianza** delle distribuzioni XY e **devianza della distribuzione** X:

$$b_1 = \frac{\sum(y_i - \bar{Y})(x_i - \bar{X})}{\sum(x_i - \bar{X})^2}$$

- L'**intercetta**, calcolata dopo il coefficiente angolare, richiede anche il calcolo del **baricentro** della distribuzione bivariata, ovvero delle **medie di X e Y**:

$$b_0 = \bar{Y} - b_1\bar{X}$$

La formula di b_1 dovrebbe richiamare alla memoria il coefficiente di correlazione r, dato dal rapporto tra covarianza al numeratore e prodotto delle due deviazioni standard Il **coefficiente angolare** è dato dal **rapporto tra codevianza** delle distribuzioni XY e **devianza della distribuzione** X:

$$r = \frac{\sum(x_{i1} - \bar{X}_1)(x_{i2} - \bar{X}_2)}{\sqrt{\frac{\sum(x_{i1}-\bar{X}_1)^2}{N-1}}\sqrt{\frac{\sum(x_{i2}-\bar{X}_2)^2}{N-1}}}$$

Come in r, il numeratore di b_1 **esprime la direzione della relazione** (positiva o negativa).
In effetti, la **regressione lineare semplice risponde alle stesse domande della correlazione, ma con alcune differenze**:

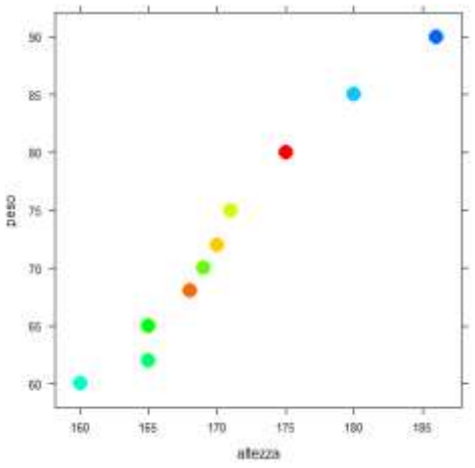
a) mentre la correlazione standardizza X_1 e X_2, e quindi i coefficienti relativi a diverse variabili sono confrontabili, la regressione mantiene l'unità di misura dei dati: perciò è difficile interpretare il confronto fra analisi condotte con variabili con diversa unità di misura. Tuttavia, si possono **standardizzare anche i coefficienti di regressione**, come vedremo, superando questa difficoltà;

b) il coefficiente di **correlazione non fornisce lo slope** della retta, ovvero non esprime la variazione unitaria in Y: **indica invece il grado di dispersione dei dati** attorno alla retta, cioè la loro variabilità, e la direzione della relazione. Il coefficiente angolare fornisce sia la quantità di variazione unitaria in Y sia la direzione di questa variazione.

Per capire i passaggi del calcolo usiamo una piccola distribuzione bivariata, sfruttando il banale esempio della **relazione tra altezza e peso**: è senso comune che tanto più una persona è alta, tanto più è pesante. **L'altezza** (metrica, continua) è quindi la variabile X, mentre il **peso** (metrico, continuo) è la variabile Y del nostro modello. La domanda è: **esiste una relazione lineare** tra altezza e peso, tale per cui, conoscendo l'altezza di una persona **siamo in grado di predirne**, con un **mimino margine di errore**, il corrispettivo peso?
Con una correlazione lineare bivariata saremmo in grado di rispondere alla prima parte della domanda (esiste una relazione lineare?), ma **non alla seconda** (possiamo predire il peso di una persona conoscendone l'altezza, sbagliando il meno possibile?).

Per scoprire quale sia il modello / funzione / equazione che descrive al meglio la relazione tra altezza e peso, reclutiamo dieci adulti ambosessi e di ciascuno rileviamo altezza e peso:

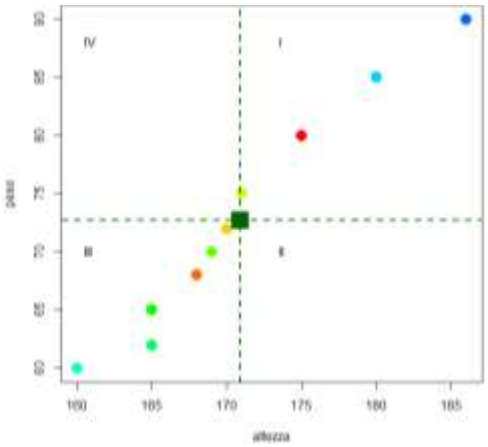
```
altezza<-c(175,168,170,171,169,165,165,160,180,186)
peso<-c(80,68,72,75,70,65,62,60,85,90)
yxyplot(peso~altezza, col=rainbow(15), pch=19, cex=2,
xlab="altezza", ylab="peso")
cor(altezza,peso)
[1] 0.9849472
cor(altezza,peso)^2
[1] 0.970121
```



La relazione è **lineare**, **positiva, fortissima**: il coefficiente di determinazione R^2 dice che le due variabili hanno in comune il **97%** della varianza.

Calcoliamo i parametri della relazione tra altezza e peso. Cominciamo con le **medie** delle due distribuzioni e tracciamole sul grafico: il loro punto d'incontro rappresenta il **baricentro della distribuzione**, e la **retta di regressione che descrive al meglio la loro relazione passerà per quel punto**. Usiamo la funzione `points(coordinata X, coordinata Y)` per inserire il baricentro, che visualizziamo come un quadrato (`pch=15`).

```
mean(altezza); mean(peso)
[1] 170.9
[1] 72.7
plot(altezza,peso,    col=rainbow(15),    pch=19,    cex=2,
  xlab="altezza", ylab="peso")
abline(h=72.7, v=170.9, col="dark green", lty=2, lwd=2)
points(mean(altezza), mean(peso), pch=15, cex=3, col="dark
  green")
text(x= c(173, 173, 160, 160),y= c(88, 70, 70, 88),labels=
  c("I", "II", "III", "IV"),pos = 4)
```



Tutte le coppie xy si distribuiscono nei quadranti I e III, che indicano una **concordanza positiva**: soggetti con peso > $\text{media}_{\text{peso}}$ e altezza > $\text{media}_{\text{altezza}}$ (I), oppure con peso < $\text{media}_{\text{peso}}$ e altezza < $\text{media}_{\text{altezza}}$ (III).

Calcoliamo la **codevianza** della relazione tra peso e altezza, cioè la somma del prodotto degli scarti di X e di Y:

```
scarti_x<-altezza-mean(altezza)
scarti_y<-peso-mean(peso)
(codevianza<-sum(scarti_x*scarti_y))
[1] 669.7
```

Poi calcoliamo la devianza di X, ovvero la somma degli scarti di X al quadrato:

```
(devianza_X<-sum(scarti_x^2))
[1] 528.9
```

E infine il coefficiente angolare, ovvero il rapporto tra codevianza e devianza:

```
(b1<-codevianza/devianza_X)
[1] 1.266213
```

La relazione tra peso e altezza è **positiva**: per **ogni cm in più** in altezza (misura unitaria in X), **si aumenta di 1.27Kg** (entità di variazione in Y).

Ora possiamo calcolare l'intercetta b_0, ovvero la media di Y meno il prodotto di b_1 per la media di X:

```
(b0<-mean(peso)-b1*mean(altezza))
[1] -143.6958
```

Come in molti casi, il valore interpretativo dell'intercetta è limitato: nel nostro esempio, **per una $\boldsymbol{altezza = 0}$** (caso evidentemente impossibile nella realtà), il **peso previsto sarebbe negativo** (caso evidentemente altrettanto impossibile nella realtà). Un modo per rendere l'intercetta un aiuto reale alla descrizione dei dati è quello di **centrare la distribuzione X**: come abbiamo visto nelle trasformazioni lineari (capitolo 4), centrare una variabile consiste nel **trasformare** i suoi punteggi grezzi in **punteggi di deviazione da un punto fisso**, di solito la media. Quando centriamo la variabile sulla **grand mean**, sottraendo la media dell'intera distribuzione da ogni punteggio grezzo, centriamo la variabile attorno a 0. In questo modo b_0 "cambia" il suo significato: ora diviene il valore di Y **quando il predittore assume un valore medio**. Quindi, se tutti i predittori sono centrati attorno alla media, allora b_0 **è il valore assunto da**

***Y* quando tutti gli *X* corrispondono al valore medio**. Torneremo, con qualche esempio, sulla centratura della distribuzione nella regressione multipla (capitolo 11).

Con R, conoscere i due parametri b_0 e b_1 con il metodo dei minimi quadrati è facilissimo: si usa la funzione `lm(y ~ x)`. Come facilmente intuibile, `lm` sta per *linear model*; attenzione all'ordine con cui costruite la sintassi: **prima *Y*, poi *X***. Siccome il modello creato con `lm` ci servirà ancora, creiamolo come oggetto e visualizziamolo:

```
modello<-lm(peso~altezza)
modello
Call:
lm(formula = peso ~ altezza)

Coefficients:
(Intercept)      altezza
   -143.696        1.266
```

Poiché b_1 esprime l'effetto di X su Y, nell'output è chiamato con il nome di X.

Scopriamo una vecchia conoscenza calcolando il **coefficiente angolare standardizzato β_1**: se b_1 non standardizzato esprime la variazione unitaria di Y al variare di una unità in X, il coefficiente angolare standardizzato β_1 **esprime la variazione in unità di deviazione standard**.

$$\beta_1 = b_1 \frac{s_x}{s_y}$$

Nella regressione semplice non ha una grande applicazione, in genere; ci sarà invece utilissimo nella regressione multipla.

Comunque, nella nostra relazione bivariata abbiamo:

```
beta1<-b1*(sd(altezza)/sd(peso)))
```

o anche:

```
modello$coefficients[2]*(sd(altezza)/sd(peso))
  altezza
0.9849472
```

Il coefficiente β_1, nella regressione semplice, corrisponde al **coefficiente di correlazione di Pearson** tra X e Y:

```
cor(altezza, peso)
[1] 0.9849472
```

Nel nostro esempio, al variare di una deviazione standard in X, il peso cresce di quasi una deviazione standard.

Nella regressione semplice[80], il coefficiente di Pearson (non negativo) appare inaspettato anche in altri modi: per esempio, corrisponde alla **media geometrica dei b_1 dei modelli $Y \sim X$ e $X \sim Y$**.

$$r_{xy} = \sqrt{b1_{yx} b1_{xy}}$$

In effetti, r, R^2 e b_1 sono legati dall'equazione: $r_{xy}^2 = b1_{yx} b1_{xy}$, e quindi: $r_{xy} = \sqrt{b1_{yx} b1_{xy}}$ (Rodgers e Nicewander, 1988). Verifichiamo con `Gmean` di `DescTools`:

```
altezza_peso<-lm(altezza~peso)
peso_altezza<-lm(peso~altezza)
Gmean(x = c(altezza_peso$coefficients[2], peso_altezza$coefficients[2]))
[1] 0.9849472
```

Ora: se il modello che abbiamo creato fosse perfetto, ovvero privo di errori / residui, le distanze tra ogni y_i (peso realmente rilevato) e il corrispettivo $\hat{y}_i$ giacente sulla retta sarebbero $= 0$: conoscendo l'altezza x_i di un soggetto, potremmo predire esattamente il suo peso y_i, cioè il peso sarebbe determinato esclusivamente dall'altezza della

[80] L'estensione della formula alla regressione multipla, con almeno due predittori, richiede concetti (al momento) avulsi dal nostro discorso; per i curiosi, data la **matrice** dei b_1, le radici quadrate degli autovalori del prodotto di queste matrici sono le correlazioni canoniche dei due set di variabili, che si riducono a un semplice *r* nel caso di una Y e una X.

persona. infatti, se nel modello lineare avessimo $\boldsymbol{e_i} = 0$, avremo $\boldsymbol{y_i} = \hat{y}_i$. Perciò, conoscendo i parametri della relazione e il valore in x_i, potremmo avere le coordinate di **tutti** i valori $\hat{y}_i$, per ogni x_i:

$y_i = (\beta_0 + \beta_1 x_i) + 0 = \hat{y}_i \rightarrow \hat{y}_i = \beta_0 + \beta_1 x_i.$

Creiamo il vettore dei valori predetti dal modello peso~altezza secondo la formula:

$$\hat{y}_i = \beta_0 + \beta_1 x_1$$

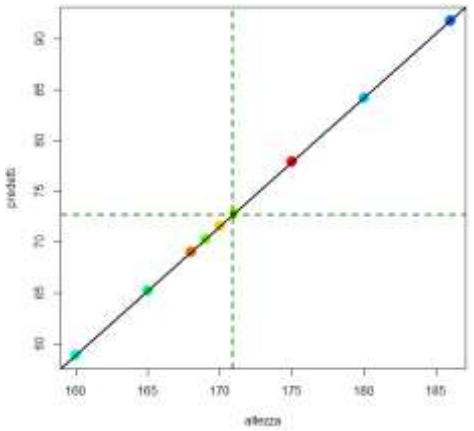
```
predetti<-b0+(b1*altezza)
```

e inseriamo i valori predetti nello scatterplot. Sovrapponiamo ai valori predetti la retta di regressione con `abline`: invece di indicare `a=-143.696` e `b=1.266` come b_0 e b_1 del modello, inseriamo direttamente il nome dell'oggetto – modello: la funzione capirà cosa vogliamo da lei.

```
plot(altezza, predetti, col=rainbow(15), pch=19, cex=2)
abline(modello, lwd=2)
```

Aggiungiamo anche il baricentro della distribuzione, e verifichiamo che cade sulla retta di regressione:

```
points(mean(altezza), mean(peso), col="dark green",
pch=15)
abline(h=72.7, v=170.9, col="dark green", lty=2, lwd=2))
```



Il vettore dei valori predetti può essere richiesto in R, dato che i valori predetti sono un elemento del modello lineare che abbiamo applicato a modello, con **`predict(modello)`** o **`fitted(modello)`** o **`modello$fitted.values`**:

```
predict(modello)
       1        2        3        4        5        6        7        8        9       10
77.89147 69.02798 71.56041 72.82662 70.29420 65.22934 65.22934 58.89828 84.22254 91.81981
fitted(modello)
       1        2        3        4        5        6        7        8        9       10
77.89147 69.02798 71.56041 72.82662 70.29420 65.22934 65.22934 58.89828 84.22254 91.81981
modello$fitted.values
       1        2        3        4        5        6        7        8        9       10
77.89147 69.02798 71.56041 72.82662 70.29420 65.22934 65.22934 58.89828 84.22254 91.81981
```

Attenzione: in realtà `predict` e `fitted` danno **informazioni diverse nei modelli non lineari** (e `predict` è più informativa); torneremo su questa differenza nella regressione logistica (capitolo 14), e dovremo tenerne conto; per ora, possiamo soprassedere.

Possiamo prevedere il valore di $\hat{y}_i$, in base al modello lineare, anche a partire da x_i che non erano stati osservati nel set di dati da cui è stato ricavato il modello, **purché tali x_i rientrino nel range della distribuzione X** utilizzata: questa operazione si chiama **interpolazione**.

Nel nostro esempio, possiamo prevedere in base a `modello` quale sia il peso di un soggetto alto 178 cm: il valore 178 non è presente tra i dati di X, ma è compreso nel suo range:

```
sort(altezza)
 [1] 160 165 165 168 169 170 171 175 180 186
(y178<-b0+b1*178)
[1] 81.69011
```

Un soggetto alto 178 cm dovrebbe pesare, con un certo margine di errore, circa 81.6 Kg. Vediamo quale dovrebbe essere il peso previsto, secondo i parametri di `modello`, per un soggetto alto 100 cm, valore **non compreso** in X:

```
(y100<-b0+b1*100)
[1] -17.07449
```

Questa seconda predizione è un **non senso**: non si può avere un reale peso negativo. Questo può capitare **quando si predicono valori $\hat{y}_i$ da x_i non compresi nel range** della distribuzione X che ha costruito il modello (**estrapolazione**): è un procedimento da **evitare**, per motivi che ora dovrebbero essere evidenti. Nel nostro caso, potremmo spiegare il valore ottenuto ipotizzando che la legge di accrescimento ponderale per soggetti alti un metro, presumibilmente bambini, sia diversa da quella che lo regola nel mondo degli adulti: quindi, il **modello lineare ricavato dagli adulti è inadeguato a rappresentarla**.

È anche possibile ricavare il ***CI* attorno a ciascuno dei valori predetti**: $CI_{\hat{y}_k} = \hat{y}_k \pm t_{(N-2,\alpha/2)} \times SE_{\hat{y}_k}$. Riprenderemo i CI dei valori predetti e il modo con cui si calcola lo SE dei valori predetti nel §9.3. Per ora, useremo semplicemente **`predict(modello, interval="confidence")`**, che calcola di default il CI al 95%. Li visualizziamo arrotondati a due decimali:

```
CI_predetti<-round(predict(modello, interval = "confidence"),2)
cbind(altezza, peso, CI_predetti)
```

```
   altezza peso   fit   lwr   upr
1      175   80 77.89 76.38 79.40
2      168   68 69.03 67.61 70.45
3      170   72 71.56 70.23 72.89
4      171   75 72.83 71.51 74.14
5      169   70 70.29 68.93 71.66
6      165   65 65.23 63.53 66.93
7      165   62 65.23 63.53 66.93
8      160   60 58.90 56.52 61.27
9      180   85 84.22 82.11 86.33
10     186   90 91.82 88.78 94.86
```

Il peso realmente osservato per un uomo alto 175 cm è 80 Kg: il corrispettivo peso predetto è pari a 77.9 Kg, e in popolazione ci aspettiamo, con il 95% di verosimiglianza, un range di predizione tra 76.4 e 79.4 Kg. Potete notare **come le previsioni sono più corrette e i *CI* più piccoli per i valori *Y* più prossimi alla media della distribuzione *Y*** (72.7 Kg): è un artefatto del metodo dei minimi quadrati, che produce stime migliori per i valori centrali della distribuzione Y. Possiamo verificarlo:

```
CI<-as.data.frame(cbind(altezza, peso, CI_predetti))
CI$ampiezza<-CI$upr-CI$lwr
CI<- CI[order(CI$peso),]
```

```
   altezza peso   fit   lwr   upr ampiezza
8      160   60 58.90 56.52 61.27     4.75
7      165   62 65.23 63.53 66.93     3.39
6      165   65 65.23 63.53 66.93     3.39
2      168   68 69.03 67.61 70.45     2.84
5      169   70 70.29 68.93 71.66     2.72
3      170   72 71.56 70.23 72.89     2.66 ← media Ypeso= 72.7
4      171   75 72.83 71.51 74.14     2.64
1      175   80 77.89 76.38 79.40     3.03
9      180   85 84.22 82.11 86.33     4.22
10     186   90 91.82 88.78 94.86     6.07
```

Ora **confrontiamo il modello – retta con i dati veri**, ovvero con i punti che identificano le **vere** coordinate di y_i in corrispondenza di x_i: andiamo, cioè, a **verificare la quantità di errore - residui** contenuta nel modello lineare.

Calcoliamo il **vettore dei residui**: i residui sono un elemento dell'oggetto **modello: `residuals`**`Errore. Il segnalibro non è definito.`**`(modello)`** o **`modello$residuals`**:

```
round(modello$residuals,3)
```

```
     1      2      3      4      5      6      7      8      9     10
 2.109 -1.028  0.440  2.173 -0.294 -0.229 -3.229  1.102  0.777 -1.820
```

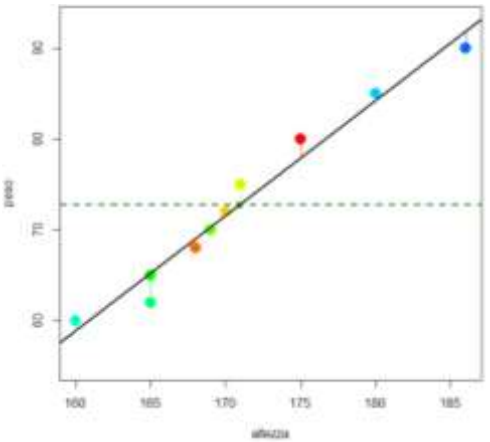
```
round(residuals(modello),3)
```

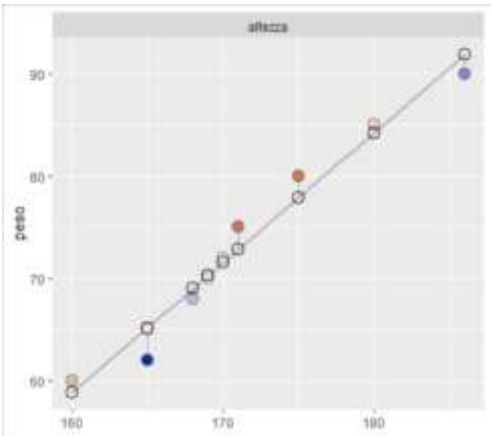
```
     1      2      3      4      5      6      7      8      9     10
 2.109 -1.028  0.440  2.173 -0.294 -0.229 -3.229  1.102  0.777 -1.820
```

Aggiungiamo il vettore dei residui al vettore dei pesi osservati e dei pesi predetti; costruiamo anche il grafico dei residui, aggiungendovi la linea che rappresenta la media di Y:

```
residui<-residuals(modello)
```

```
round(cbind(peso,
predetti, residui),2)
 peso predetti residui
   80    77.89    2.11
   68    69.03   -1.03
   72    71.56    0.44
   75    72.83    2.17
   70    70.29   -0.29
   65    65.23   -0.23
   62    65.23   -3.23
   60    58.90    1.10
   85    84.22    0.78
   90    91.82   -1.82
```





```
plot(altezza, peso, col=rainbow(15), pch=19, cex=2, ylim=c(55,
  93))
abline(modello, lwd=2)
points(mean(altezza), mean(peso), col="dark green", pch=15)
abline(h=72.7, col="dark green", lty=2, lwd=2)
segments(altezza, peso, altezza, predetti, col=rainbow(15))
```

```
plot_residuals(fit =
lm(peso~altezza), geom.size = 4)
```

Ora abbiamo tutti gli elementi che compongono l'equazione di regressione:

```
cbind(peso, altezza, residui)
   peso altezza    residui
1    80     175  2.1085271
2    68     168 -1.0279826
3    72     170  0.4395916
4    75     171  2.1733787
5    70     169 -0.2941955
6    65     165 -0.2293439
7    62     165 -3.2293439
8    60     160  1.1017206
9    85     180  0.7774627
10   90     186 -1.8198147
```

…più l'effetto del predittore, cioè la variazione unitaria Y per ogni variazione unitaria in X…

Il valore Y del soggetto i appartenente al livello j di X… → $y_{ij} = (b_0 + b_1 X_j) + e_{ij}$ ← *…più la quota di errore del modello riferito al soggetto ij*

…è dato dal valore in Y quando X =0 — *moltiplicato per il valore in X_j del soggetto…*

Il peso del soggetto 9 $y_9 = 85$ è dato dal valore in Y per $X = 0$ ($b_0 = -143.695784$) più l'effetto del predittore $X_{altezza}$ ($b_1 = 1.266213$) moltiplicato per l'altezza del soggetto 9 ($X_9 = 180$), più la quota di errore commessa dal modello per il soggetto 9 ($residuo_9 = .7774627$):

```
-143.695784+(1.266213*180)+.7774627
[1] 85.00002
```

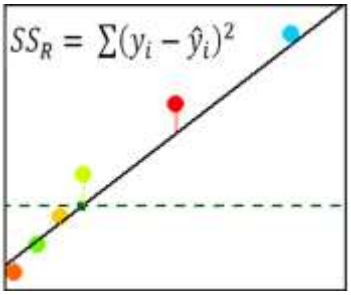
Il peso del soggetto 2 $y_2 = 68$ è dato dal valore in Y per $X = 0$ ($b_0 = -143.695784$) più l'effetto del predittore $X_{altezza}$ ($b_1 = 1.2666213$) moltiplicato per l'altezza del soggetto 2 ($X_2 = 168$), più la quota di errore commessa dal modello per il soggetto 2 ($residuo_2 = -1.0279826$):

```
-143.695784+(1.266213*168)+(-1.0279826)
[1] 68.00002
```

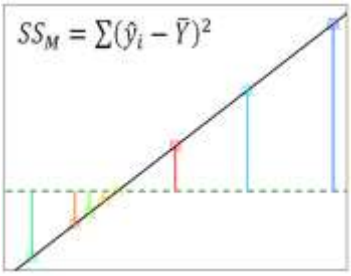
Eccetera.

9.2 Fit e significatività del modello

Nel plot precedente abbiamo tracciato anche la media di Y, perché anch'essa concorre a definire le tre diverse **distanze** che compaiono nel grafico:



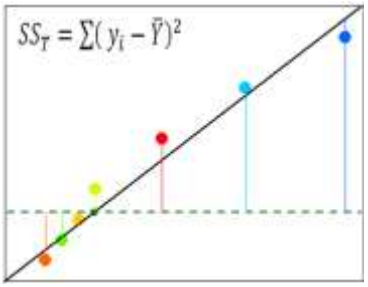
1) la **distanza tra ogni y osservato e il suo corrispettivo valore predetto $\hat{y}_i$** sulla retta esprime l'errore compiuto dal modello – retta nell'indovinare il vero valore di y. Già sappiamo che la somma di tutte queste distanze esprime la **devianza d'errore o residua del modello**: tanto più grande è questo valore, tanto peggiore è il fit del modello.



2) la **distanza tra ogni valore predetto $\hat{y}_i$ e la media di Y** esprime quanto il modello–retta si discosta dalla media di Y nel prevedere il vero valore di Y. Se non ci fosse alcuna relazione tra altezza - X e peso - Y, il modo più affidabile per conoscere il vero peso di una persona y_i sarebbe solo basarsi sulla media del peso nel campione. *Ad esempio, se non sapessi quanto è alto Gino, ma sapessi che in media gli uomini pesano 76kg, il modo per rendere più probabile la vincita alla scommessa sul peso di Gino sarebbe puntare sul peso medio della sua popolazione di appartenenza.*

Questa situazione sarebbe graficamente rappresentata da una retta di regressione perfettamente sovrapposta alla media di Y: per tutti i valori di altezza in X, il valore predetto sarebbe sempre e solo la media di Y. Quindi, più la retta di regressione si discosta dalla media (quindi più è grande la somma delle distanze tra ogni valore predetto e la media), più forte è l'effetto di X.

La somma di queste distanze si definisce **devianza del modello o devianza di regressione**.



3) la **somma delle distanze tra ogni valore y_i e la media di Y** esprime la variabilità totale della distribuzione Y: la **devianza totale** corrisponde alla somma della devianza d'errore e della devianza del modello..

Abbiamo creato il vettore dei residui e il vettore dei valori predetti del modello. Possiamo usarli per calcolare queste tre devianze:

```
(devianza_residui<-sum(residui^2))
[1] 26.11722
(devianza_modello<-sum((predetti-
mean(peso))^2))
[1] 847.9828
(devianza_residui<-sum(residuals(modello)^2)=
 [1] 26.11722
(devianza_modello<-sum((predict(modello)-
mean(peso))^2))
 [1] 847.9828
(devianza_totale<-devianza_modello+devianza_residui)
[1] 874.1
```

Il **rapporto tra la devianza del modello e la devianza totale è il coefficiente di determinazione R^2 del modello lineare**: infatti, tanto è maggiore la variabilità di Y attribuibile a X, rispetto a tutta la variabilità complessiva di Y, tanto migliore sarà la predizione dei valori Y conoscendo i valori di X. Il valore di questo rapporto, in realtà, lo conosciamo già, dato che:

```
devianza_modello/devianza_totale
[1] 0.970121
cor(altezza, peso)^2
[1] 0.970121
```

Il **coefficiente di determinazione R^2**, che avevamo calcolato come quadrato del coefficiente di correlazione bivariata, nella regressione semplice[81] corrisponde al **rapporto** tra **devianza del modello e devianza totale**. Può essere calcolato anche come **correlazione al quadrato tra il vettore degli y_i e dei valori predetti dal modello**. Nel nostro modello, il **97.1% della variabilità del peso è attribuibile all'effetto dell'altezza**.

Se un rapporto tra devianze ci dà R^2, cosa ci dirà un rapporto tra devianze ponderate per i gradi di libertà, ovvero un **rapporto tra varianze - MS**? Dividendo ciascuna devianza per i propri gradi di libertà, otteniamo tre **varianze**:

- ✓ $df_{modello=}$ **$numero\ di\ parametri - 1$** : nella regressione semplice, dato che abbiamo una b_0 e un solo b_1, i $df_{mdoello}$saranno sempre $2 - 1 = 1$; ne consegue che nella regressione semplice $SS_{modello} = MS_{modello}$;
- ✓ $df_{errore=}$ **$N - numero\ b_1 - 1$**: nella regressione semplice, dato che abbiamo un solo b_1, i df_{errore} saranno sempre $= N - 2$;
- ✓ $df_{totale=}$ **$N - 1$**

Dato che le devianze vengono divise ciascuna per una diversa quantità, per le varianze non varrà la proprietà additiva che avevamo notato nelle devianze: $SS_{tot} = SS_M + SS_R$, ma $MS_{tot} \neq MS_M + MS_R$. Nel nostro esempio:

```
(varianza_modello<-devianza_modello/1)
[1] 847.9828
(varianza_residui<-devianza_residui/8)
[1] 3.264653
(varianza_totale<-devianza_totale/9)
[1] 97.12222
```

In realtà, ignoreremo la varianza totale, perché è il **rapporto tra la varianza del modello e la varianza di errore** a essere interessante: il **rapporto F** (proposto da Snedecor nel 1934, ma da lui così battezzato in onore di Fisher) indica di **quante volte la variabilità di Y attribuibile a X (MS_M) è maggiore della variabilità di Y determinata dall'errore (MS_{RM}).**

$$F_{[1;N-2]} = \frac{MS_M}{MS_R}$$

Chiaramente, se il rapporto fosse esattamente $F = 1$ ($MS_M = MS_R$) o addirittura $F < 1$ ($MS_M < MS_R$), la capacità predittiva di X sarebbe inesistente: la relazione lineare tra X e Y non sarebbe significativa. Ma **di quanto il rapporto F dovrebbe essere superiore a 1 (**$MS_M > MS_R$) perché la relazione $Y \sim X$ possa essere considerata significativa? Poiché il rapporto F si distribuisce come un quantile di una distribuzione F (capitolo 5), i cui gradi di libertà sono rispettivamente quelli di SS_M e SS_R, è **possibile attribuire un $p - value$ al rapporto F ottenuto o uno più estremo, sotto condizione di H_0**. Se il $p - value$ è inferiore alla soglia α prescelta, rifiutiamo H_0: la relazione è significativa e la quantità di varianza di Y spiegata da X è significativamente maggiore della quantità di varianza di Y spiegata da tutto ciò che non è X (errore di misura, altre variabili non inserite nell'equazione).

Il rapporto F che esprime di quante volte è maggiore (o minore) la quantità di varianza del peso spiegata dall'altezza rispetto a quella non spiegata dall'altezza (dall'esercizio fisico, dalle calorie giornaliere, eccetera) è:

```
(F<-varianza_modello/varianza_residui)
[1] 259.7467
```

[81] Nella regressione multipla avremo il coefficiente di determinazione **multiplo**, che non corrisponderà più al coefficiente di correlazione al quadrato tra X e Y.

La quantità di variabilità del peso predetta dall'altezza è 259.7 volte maggiore di quella determinata da altri fattori. Calcolare il $p - value$ per F di questa entità è praticamente inutile, ma facciamolo lo stesso:

```
pf(q = 259.7467,df1 = 1, df2 = 8, lower.tail = FALSE)
[1] 2.205866e-07
```

La probabilità di ottenere un rapporto $F = 259.7$ o superiore, sotto condizione di ipotesi nulla, è decisamente un evento molto, molto raro (0.00000022): la relazione tra peso e altezza **non dovrebbe essere casuale**.

R ci dispensa dal calcolo di devianze, varianze, F e relativi $p - value$, e anche da quello del coefficiente R^2: la funzione `summary(lm(y~x))`, o, se il modello è stato creato come oggetto di classe `lm`, `summary(modello)` ci dice tutto. Se intendete applicare la regressione **solo a una parte dei soggetti**, si può specificare la variabile filtro con l'argomento `subset= variabile filtro== "criterio"`, come abbiamo visto nella correlazione.

Spezziamo l'output per commentarlo:

```
summary(lm(peso~altezza))
Call:
lm(formula = peso ~ altezza)
Residuals:
    Min      1Q  Median      3Q     Max
-3.2293 -0.8445  0.1051  1.0207  2.1734
    ↑                            ↑
```

Sovrastima: $y < \hat{y}$ *sottostima:* $y > \hat{y}$

```
-----
```

La prima sezione sintetizza gli errori del modello: gli errori nella previsione del peso in base all'altezza vanno da una **sovrastima** (**peso meno predetti**) massima di 3.2 kg a una sottostima (peso meno predetti) massima di 2.2 Kg; l'errore mediano è di circa un etto: non molto. Avevamo già visto il vettore dei residui:

```
round(sort(residui),2)
    7    10     2     5     6     3     9     8     1     4
-3.23 -1.82 -1.03 -0.29 -0.23  0.44  0.78  1.10  2.11  2.17

-----
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -143.69578   13.43899  -10.69 5.14e-06
altezza        1.26621    0.07857   16.12 2.21e-07
-----
```

In questa sezione sono descritti i parametri del modello, l'intercetta (prima riga) e il coefficiente angolare (seconda riga). Sotto la colonna `Estimate` sono indicati i loro valori; le tre colonne seguenti ci occuperanno subito aver dato un'occhiata all'ultima sezione:

```
-----
Residual standard error: 1.807 on 8 degrees of freedom
Multiple R-squared:  0.9701,     Adjusted R-squared:  0.9664
F-statistic: 259.7 on 1 and 8 DF,  p-value: 2.206e-07
-----
```

Qui sono riportate le statistiche riferite al fit del modello (R^2) e alla sua significatività (rapporto F e relativo $p - value$). Il ***residual standard error*** è l'errore standard della distribuzione dei residui del modello: potremmo considerarlo un indice negativo di fit (tanto più grande è, tanto maggiore è l'errore contenuto nel modello), ma non avendo un range di variazione ed essendo dipendente dall'unità di misura delle variabili, non ha un gran valore interpretativo per il singolo modello. Lo ritroveremo nella regressione multipla, quando parleremo del confronto tra modelli riferiti a una stessa Y.

```
sqrt(varianza_residui)
[1] 1.806835
```

L'output di R definisce "*multiple*" il coefficiente R^2 sia nella regressione semplice (in cui non è multiplo) sia in quella multipla: nella regressione semplice, esprime la quantità di varianza di Y spiegata dall'unico predittore X, mentre in quella multipla esprime la quantità di varianza di Y spiegata da **tutti** i predittori inseriti nel modello.

Il **coefficiente R^2 *adjusted*** è una **stima** della reale quota di variabilità di Y spiegata da X **in popolazione**: avremo **sempre $R_{adj} < R^2$**. La differenza tra i due coefficienti è un indicatore della **capacità predittiva del modello lineare se generalizzata alla popolazione**: se è piccola, il modello tiene; se è grande, no.

In `lm`, R usa la formula di Wherry (Yin e Fan, 2001) per R^2_{adj}; ce ne sono diverse altre.

$$R^2_{adj} = 1 - (1 - R^2)\frac{N-1}{df_{errore}}$$

Torneremo a occuparci di R^2_{adj}R$^2_{adj}$, del suo rapporto con R^2 e dei vantaggi che offre nella definizione dei fit di modelli a confronto nella regressione multipla.

Ora concludiamo la sezione dell'output dedicata ai coefficienti:

```
-----
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -143.69578   13.43899  -10.69 5.14e-06
altezza        1.26621    0.07857   16.12 2.21e-07
-----
```

L'**errore standard** di b_0 (13.44) e b_1 (.078) serve a due cose.

Come i più intuitivi avranno intuito, in primis serve a calcolare il CI in popolazione per b_0 (che di solito non interessa a nessuno) e b_1: $CI_{b_0} = b_0 \pm z\alpha_{/2} \times SE_{b_0}$ e $CI_{b_1} = b_1 \pm z\alpha_{/2} \times SE_{b_1}$. Ad esempio, con l'abituale probabilità del 95% (e quindi $z_{\alpha/2} = |\mathbf{1.96}|$) avremo:

```
UL_b1<-b1+1.96*.07857
LL_b1<-b1-1.96*.07857
round(LL_b1,1); round(UL_b1, 1)
[1] 1.1
[1] 1.4
```

In popolazione, la reale variazione del peso, per ogni cm di altezza in più, varia da 1.1kg a 1.4kg: il CI è piccolo, la precisione della stima del modello è buona.

Per i precisini: gli SE dei parametri si calcolano così: $SE_{b0} = \sqrt{MS_R\left(\frac{1}{N} + \frac{\bar{X}^2}{\sum(x_i - \bar{X})}\right)}$ $SE_{b1} = \sqrt{\frac{MS_R}{\sum(x_i - \bar{X})^2}}$

, ma per tracciare i CI noi useremo la comoda funzione `confint(modello, level= .95)`, che usa i quantili t per la definizione della verosimiglianza:

```
confint(modello)
                  2.5 %      97.5 %
(Intercept) -174.686153 -112.705415
altezza        1.085041    1.447385
```

il quantile t e il rispettivo $p - value$ che compaiono nelle righe di b_0 e b_1 si riferiscono a un t-test che risponde alle ipotesi nulle:

- ✓ $H_0: \beta_0 = 0$; quando $X = 0$, $Y = 0$. Non è un'ipotesi cui si sia spesso interessati.
- ✓ $\boldsymbol{H_0: \beta_1 = 0}$; per una variazione unitaria di X, la variazione di Y è $= 0$, ovvero non c'è: le due variabili non hanno una relazione lineare. Se una retta di regressione lineare ha **un'inclinazione $= 0$**, allora **al variare di X i valori Y non cambiano in maniera prevedibile e si distribuiscono casualmente attorno alla media di Y** (che rimane l'unico modello in grado di rappresentare la distribuzione Y).

La logica del t-test per β_0 e β_1 è ancora un **rapporto** tra l'effetto o segnale (differenza tra il **parametro** empirico e il **parametro del modello previsto da** H_0, ovvero **0**), e il "rumore di fondo" (*noise*), cioè l'errore standard del parametro: ecco **la seconda cosa** cui serve **l'ES**.

$$t_{\beta 0, N-2} = \frac{b_0 - \beta_0}{SE_{b0}}$$

$$t_{\beta 1, N-2} = \frac{b_1 - \beta_1}{SE_{b1}}$$

Questo rapporto t si distribuisce come un quantile t in una distribuzione con $df = N - 2$, per ipotesi monodirezionali e bidirezionali. A parte $\beta_{H0} = 0$, questo t-test si può usare per confrontare qualsiasi altro valore atteso β_0 o β_1 con quelli

descritti dal modello lineare empirico; per esempio, si possono confrontare b_1 ottenuti in campioni differenti mettendo in relazione le stesse X e Y. Nel nostro esempio:

```
t_b0<-(-143.69578 - 0)/13.43899
t_b1<-(1.26621 - 0)/.07857
t_b0; t_b1
[1] -10.69245
[1] 16.11569
```

Tanto più è ridotto lo SE di b_1 (al denominatore), tanto più probabilmente t risulterà significativo, perché il rapporto tra effetto al numeratore e SE sarà più a favore del primo. Quindi, per aumentare la probabilità di trovare un modello lineare significativo dovremmo ridurre lo SE di b_1.

La formula dello SE ci suggerisce come farlo: al denominatore troviamo la devianza di X, quindi, se **aumentiamo la variabilità di X riduciamo lo SE dell'effetto di X**, aumentando le chances di trovare un effetto significativo nel campione, se esiste in popolazione.

$$SE_{b1} = \sqrt{\frac{MS_R}{\sum(x_i - \bar{X})^2}}$$

Perciò, sarà utile programmare nel campionamento anche soggetti con valori estremi in X.

Due ovvietà:

1) se nel 95%CI di b_1 (o b_0) è compreso 0, accettiamo H_0 per$\alpha = .05$;
2) nella regressione lineare semplice il modello lineare è definito da un solo predittore. Quindi, il test *overall* F di adattamento del modello e il test t di significatività del parametro danno esattamente **lo stesso risultato**, tanto più che un quantile t è la radice quadrata di un quantile F, o se preferite, il quantile F è il quadrato di un quantile t:

```
F; t_b1^2
[1] 259.7467
[1] 259.7156
t_b1; sqrt(F)
[1] 16.11569
[1] 16.11666
```

Nella regressione multipla, con più predittori, il modello complessivo può avere un buon adattamento (F significativo, R^2 soddisfacente), ma **uno o più dei predittori possono non essere significativi**, ovvero possono non contribuire a spiegare la variazione di Y: l'effetto degli altri è sufficientemente forte da "coprire" il loro non-effetto, se considerati assieme. Ecco perché in quel caso sarà importante leggere sia t sia F.

Calcolate il modello lineare della relazione BDI~PTQ di cui abbiamo visto il grafico: quali conclusioni potremo trarre sulla relazione tra rimuginio e depressione?

Concludiamo (per ora) il discorso sui parametri dando un po' di lustro interpretativo alla trascurata intercetta. Nel caso in cui b_1 non sia significativamente diverso da 0, e quindi quando la relazione Y~X non è significativa, il valore di un soggetto sarà dato da: $y_i = (b_0 + [0 \times x_i]) + e_i$, cioè $y_i = b_0 + e_i$.

$$y_i = b_0 + b_1 x_i + e_i \;\rightarrow\; y_i = (b_0 + [0 \times x_i]) + e_i \;\rightarrow\; y_i = b_0 + e_i$$

Dato che l'intercetta è $b_0 = \bar{Y} - b_1\bar{X}$, quando $\boldsymbol{b_1 = 0}$ avremo: $b_0 = \bar{Y} - 0\bar{X}$, cioè $\boldsymbol{b_0 = \bar{Y}}$:

$$b_0 = \bar{Y} - b_1\bar{X} \;\rightarrow\; b_0 = \bar{Y} - 0\bar{X} \;\rightarrow\; b_0 = \bar{Y} \;\rightarrow\; y_i = \bar{Y} + e_i$$

Perciò, se $\boldsymbol{b_1 = 0}$ allora $\boldsymbol{b_0 = \bar{Y}}$, e il valore in Y un soggetto sarà dato dalla media di Y sommata alla quota di errore che la media compie nella stima di quel particolare valore. In questo caso particolare l'intercetta rappresenta la cosiddetta *grand mean*, ovvero la media della popolazione cui il soggetto appartiene.

9.3 Intervallo di fiducia della retta e intervallo di predizione

Nei paragrafi precedenti, abbiamo usato `abline` per rappresentare la retta di regressione campionaria: nel caso di dati "veri", però, è estremamente più interessante tracciare **attorno alla retta di regressione** campionaria il suo **intervallo di fiducia**: il ***CI* delle rette probabili** (o **confidence bands**) rappresenta il range entro il quale, con una prefissata probabilità, sta la vera retta di regressione (il modello lineare) in popolazione. Vediamo la logica del suo calcolo, poi la faremo tracciare da una comoda funzione.

Abbiamo già visto (§9.1) la formula del ***CI* di un valore predetto $\hat{y}_k$**: $CI_{\hat{y}_k} = \hat{y}_k \pm t_{(N-2,\alpha/2)} \times \boldsymbol{SE_{\hat{y}_k}}$

Finiamo di descrivere la sua formula concentrandoci sul modo in cui viene calcolato lo SE del valore predetto $\hat{y}_k$:

$$SE_{\hat{y}_k} = \sqrt{MS_R \times \left(\frac{1}{N} + \frac{(x_k - \bar{X})^2}{\sum(x_i - \bar{X})^2}\right)}$$

La MS_R nella formula fa sì che:

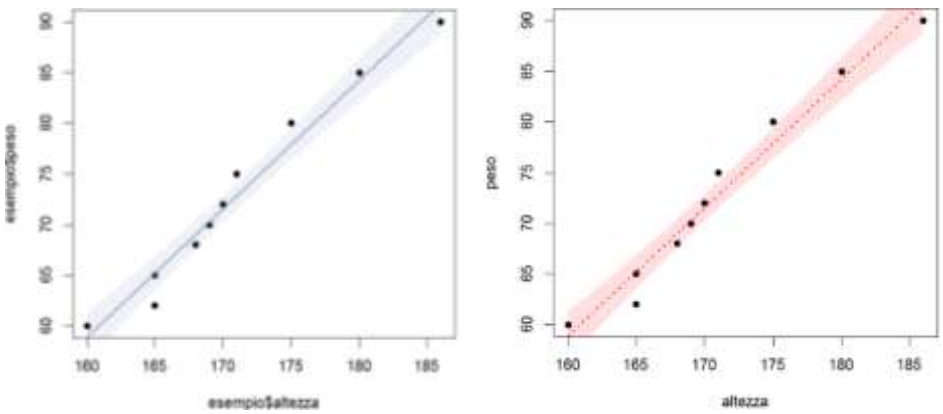
- l'ampiezza del CI aumenti al crescere di MS_R: al crescere dell'errore del modello, aumenta l'incertezza sulla stima;
- l'ampiezza diminuisca all'aumentare di N: l'abbiamo già visto nei CI di altre statistiche;
- l'ampiezza diminuisca al crescere della devianza di X: se volete una maggiore precisione della predizione, oltre a raccogliere molti soggetti sceglieteli (ragionevolmente) eterogenei;
- **il margine di errore vari in funzione di X**: ha valori minimi (e quindi rappresenta una previsione più precisa) quando x_k è prossimo alla media di X, e tende a crescere (previsioni sempre meno precise) quanto più x_k è distante dalla media, nelle due direzioni → il CI del valore predetto **non ha un'ampiezza costante**.

Ora, tutti i CI dei punti predetti sulla retta determinano **il *CI* della retta**: un $95\%CI$ garantisce, con una probabilità del 95%, di contenere la retta di regressione attesa in popolazione secondo il modello $\hat{y}_i = b_0 + b_1 X_i$, in campionamenti ripetuti, Proprio a ragione della variabilità dei CI dei valori predetti, il CI della retta assume una **forma curva**: ristretto al centro, più ampio all'estremità. Attenzione a **non equivocare**: non è che le rette di regressione in popolazione diventano curve! È **l'insieme** tutte le rette probabili secondo il livello di verosimiglianza scelto che assume questa forma.

Costruiamo il CI della relazione `peso~altezza`: basta **tracciare il solito plot a dispersion**e di X e Y, poi usare la funzione **`lines.lm(x= formula)`** di `Desctools`, che inserisce nel plot sia la retta sia il suo CI. L'argomento `x=` si aspetta una formula `lm(y~x, data=dataframe)`, o un oggetto precedentemente costruito con questa formula; `conf.level=` gestisce la verosimiglianza, di default `=.95`. `lines.lm` accetta solo vettori che fanno parte di dataframe o matrici. Nella vita reale non è un problema, dato che in genere si lavora su dataframe, ma per il nostro esempio siamo obbligati al passaggio preliminare della costruzione del mini-dataframe **`esempio`**, che contiene `altezza` e `peso`.

```
esempio<-
  data.frame(altezza=altezza
  , peso=peso)
modello<-lm(peso~altezza,
  data=esempio)
plot(esempio$altezza,
  esempio$peso, pch=19)
lines.lm(x = modello)
```

Potete giocare con colori e tipo di linee del CI con i soliti argomenti `col=`, `lty=`, `lwd=`, ... in `lines.lm`



Costruito un modello, può sorgere la necessità di **stimare valori aggiuntivi** al campione(interpolazione): questo nuovo valore predetto è calcolato esattamente come qualsiasi altro, ma **cambia il suo intervallo di predizione** (*prediction interval*). L'intervallo di predizione è il range che contiene, con una verosimiglianza prefissata, il **valore di Y per una nuova osservazione**, dato uno specifico valore di X.

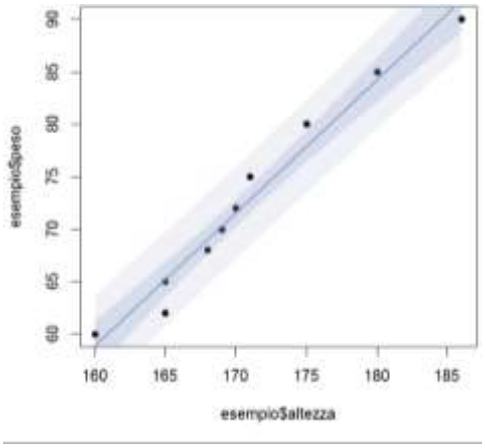
Osserviamo la parte sotto radice quadrata, che è lo SE del valore y aggiuntivo[82]: è facile notare, confrontandola con la formula precedente, che determina più incertezza:

$$\hat{y}_k \pm t_{(N-2,\alpha/2)} \times \sqrt{MS_R \times \left(1 + \frac{1}{N} + \frac{(x_k - \bar{X})^2}{\sum(x_i - \bar{X})^2}\right)}$$

Quindi, un **intervallo di predizione è sempre più ampio del CI della retta**. Mentre il CI della retta considera la sola incertezza della stima derivante dal campionamento (rispetto alla popolazione), l'intervallo di previsione considera anche l'incertezza della stima derivante dalla variabilità degli individui attorno alla media predetta di Y.
Intervalli di predizione e CI della retta sono perciò correlati, ma si calcolano diversamente e per scopi differenti: mentre un qualsiasi CI intende stimare un parametro, l'intervallo di predizione intende stimare una **variabile aleatoria,** per una data verosimiglianza.

Se necessario, potete tracciare l'intervallo di predizione aggiungendo l'argomento `pred.level=` a `lines.lm`. Di default, è `FALSE`, quindi non viene tracciato; se aggiungete la verosimiglianza desiderata per il per l'intervallo, sarà tracciato con una sfumatura più chiara di quella dedicata al CI della retta.

```
lines.lm(x = modello, pred.level=.95)
```



9.4 L'influenza sul modello: outliers e casi influenti

Una delle fonti di errore più comuni, che rende le previsioni di un modello non attendibili e non generalizzabili alla popolazione, è la presenza nel campione di un (piccolo) numero di casi che esercitano una **eccessiva** influenza sul modello – retta. L'influenza sul modello viene valutata osservando la variazione dei suoi parametri (in particolar modo del coefficiente angolare) quando il soggetto viene tolto dal campione: se la variazione è rilevante, quel caso è un **influential case**. Dato che la sua presenza nel campione rende instabile il modello, viene soppresso. Naturalmente, **non** si procede per tentativi ed errori, eliminando un soggetto per volta e ricalcolando i parametri[83] fino ad escludere la presenza di casi influenti o fino a toglierli tutti, ma si calcolano due quantità per ogni soggetto, in base alle quali viene definito come caso influente o non influente. Infatti, un influential case è un soggetto che **possiede contemporaneamente due sgradevoli caratteristiche**: è un **outlier bivariato e** possiede un **alto valore di leverage**. Si possono individuare queste due proprietà in ogni caso della distribuzione bivariata, valutare se coincidono in uno o più soggetti e verificare il cambiamento dei parametri eliminando queste "combo". Vediamo come procedere.

[82] Il valore 1 entro le parentesi tonde è dovuto all'introduzione di un'altra fonte di varianza di errore – ma possiamo soprassedere.
[83] Un procedimento di questo tipo (bootstrap) ha altri scopi.

9.4.1 Gli outliers bivariati

I casi outliers bivariati sono **coloro che hanno i maggiori residui**, ovvero quelli per cui l'errore di predizione del modello è di gran lunga maggiore rispetto agli altri $\hat{y}$. Nella regressione, un outlier bivariato **è un caso i_{yx} che ha un insolito y_i rispetto al suo x_1**. Nell'esempio peso~altezza, potrebbe essere un soggetto alto 195 cm e pesante 65 kg, o alto 150 cm e pesante 80 kg: di per sé, 65kg e 80kg non sono affatto pesi estremi in una popolazione di adulti, ma non ce li aspetteremmo in persone rispettivamente molto alte e molto basse. A differenza di un outlier univariato, quindi, l'outlier bivariato non necessariamente ha valori estremi in Y: è l'associazione $y_i x_i$ a essere "strana".

Anche in un modello con un buon fit, qualche outliers può comparire: per stabilire quando il residuo è tanto grande da non essere una semplice oscillazione casuale dalla retta, possiamo **standardizzare in punti z i residui** (dividendoli per una stima della loro deviazione standard). Così, indipendentemente dalla loro unità di misura, si potrà dire che:

1) se **una osservazione** ha un residuo standardizzato $> |\mathbf{2}|$ **o** $> |\mathbf{2.5}|$, è molto probabile che non rappresenti un'oscillazione casuale;
2) se più **dell'1%** del campione ha residui standardizzati $> |\mathbf{2.5}|$ (o se più del **5%** ha residui standardizzati $> |\mathbf{2}|$), il modello ha un livello di errore complessivo inaccettabile, e probabilmente un brutto fit.

I residui standardizzati sono disponibili in `rstandard(modello)`: possono essere plottati, riassunti, salvati, eccetera. Per esempio, un'operazione preliminare **potrebbe** essere quella di chiedere il `summary` dei residui standardizzati del modello e verificare se ci sono valori $> |2.5|$ (o $> |2|$, ma è un valore che molti giudicherebbero troppo conservativo): in caso negativo, evidentemente non ci sono outliers bivariati e di conseguenza non ci saranno nemmeno casi influenti; quindi, si può passare a un altro problema. Se invece il `summary` evidenziasse almeno un outlier, bisogna approfondire e identificarlo.

Nel modello peso~altezza non abbiamo residui standardizzati critici – anche se ci siamo andati vicini:

```
summary(rstandard(modello))
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.95700 -0.49690  0.05885 -0.03260  0.68120  1.26800
```

Se volete imparare una cosa nuova, `which.min(variabile)` e `which.max(variabile)` identificano i casi con valore minimo e massimo di una distribuzione, in questo caso quella dei residui; possiamo visualizzare i loro valori così:

```
rstandard(modello)[which.min(rstandard(modello))]
        7
-1.956883
rstandard(modello)[which.max(rstandard(modello))]
       4
1.267944
```

Vediamo dati veri con il dataset `cuccioli`, che approfondiremo nella regressione multipla. I dati riguardano cagnolini di varie razze, osservati nel loro allevamento in un setting controllato, in cui per cinque minuti potevano giocare, interagire con persone note ed estranee, correre, starsene per i fatti loro, eccetera. Sono riportate le percentuali del tempo trascorso a fare le varie azioni codificate dagli etologi, insieme a un giudizio sui tratti di personalità del cucciolo (amichevole, estroverso, riservato...) dato dagli osservatori. Vediamo se il giudizio sul nevroticismo (`$nevroticismo`) del cucciolo è predetto / dipende dal tempo passato dal cucciolo a osservare l'ambiente, senza giocare, muoversi o interagire con gli umani (`$osserva`):

```
nevro<-lm(cuccioli$nevroticismo~cuccioli$osserva)
```

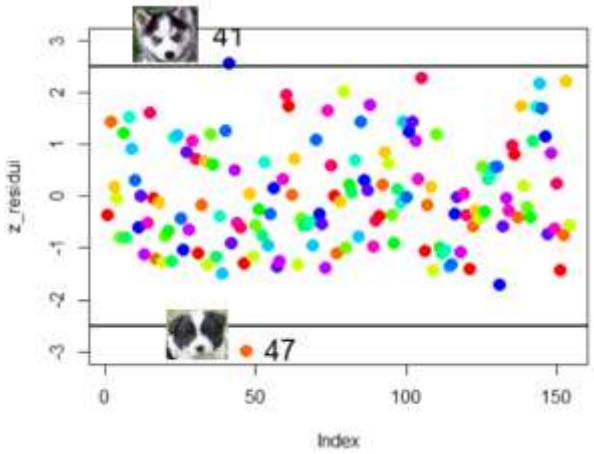
```
summary(nevro)
Call:
lm(formula = cuccioli$nevroticismo ~ cuccioli$osserva)
Residuals:
    Min      1Q  Median      3Q     Max
-12.805  -3.768  -0.677   3.533  12.858
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       14.7727     0.5568  26.530  < 2e-16
cuccioli$osserva   0.8374     0.1019   8.215 8.56e-14
---

Residual standard error: 5.012 on 152 degrees of freedom
Multiple R-squared:  0.3075,     Adjusted R-squared:  0.3029
F-statistic: 67.49 on 1 and 152 DF,  p-value: 8.561e-14

confint(nevro)
                     2.5 %    97.5 %
(Intercept)      13.6725256 15.872788
cuccioli$osserva  0.6359898  1.038742
```

Sembra che il comportamento ritroso del cucciolo influenzi il giudizio sulla dimensione di personalità: per ogni punto percentuale in più di tempo passato a osservare senza fare nulla, il giudizio sul nevroticismo aumenta di .8 punti ($b_1 = .84$); in popolazione, la variazione in Y è compresa tra .6 e 1.1 punti in più. La relazione tra Y e X è significativa; lo leggiamo sia dal $p - value$ del coefficiente angolare ($p - value < .05$) sia dal fatto che il $95\%CI$ di b_1 non contiene il valore previsto da $H_0 = 0$. Il comportamento del cucciolo spiega oltre il 30% della variabilità dei giudizi ($R^2 = .31$); in popolazione, la stima della varianza spiegata è praticamente immutata ($R^2_{adj} = .30$). La varianza dei giudizi spiegata dal predittore comportamento è 67.5 volte maggiore della varianza dei giudizi attribuibile a qualsiasi altra causa ($F = 67.5$). Gli errori del modello (attenti: nel `summary` li vediamo con l'unità di misura di Y, non sono standardizzati) hanno una mediana prossima a zero, il che va bene, ma una sottostima e una sovrastima massime piuttosto rilevanti: per **almeno** due cuccioli, la sola osservazione del tipo di comportamento del cucciolo ci ha fatto sbagliare di quasi 13 punti nell'indovinare il giudizio attribuito alla loro personalità, per sovrastima (-12.8) e per sottostima (12.9). **Almeno** uno di questi cuccioli potrebbe essere un **outlier bivariato**?

```
summary(rstandard(nevro))
      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.978000 -0.756600 -0.135800 -0.002419  0.707900  2.574000
```

Oh sì, più di uno. Approfondiamo la loro identificazione: possiamo plottarli e vederli sul grafico, o usare `which(criterio)`, che indica i numeri di riga dei casi con residui standardizzati $> |2|$:

```
z_residui<-rstandard(nevro)
 plot(z_residui, col=rainbow(15), pch=19, cex=1.5,
  ylim=c(-3,3))
abline(h = c(-2.5,2.5), lwd=2)
identify(z_residui)
[1]  41  47
```



Oppure, usando `which`:

```
which(abs(z_residui)>2.5)
41  47
  41  47
cuccioli[c(41,47), 1:2]
     cucciolo         razza
41 Q82_Adoniz       alaskan
47   Q115_Emy border collie
```

Sono soprattutto i piccoli border collie Emy e Alaskan malamute Adoniz a preoccuparci un po'.

Ora dobbiamo procedere a valutare l'altra proprietà del caso, cioè l'avere un alto valore di leverage: se uno degli outlier bivariati avrà anche un eccessivo valore di leverage, sarà un probabile influential case.

9.4.2 Casi con alto valore di leverage e influential cases

Quanto più i casi hanno valori insoliti in X (cioè, **molto lontani dalla media di X in entrambe le direzion**i), tanto più hanno la **potenzialità** di influire sulla (cioè di **agire come leve sulla***)* retta di regressione.

Si dirà quindi che il caso con valore insolito in X ha un **alto valore di leverage**, dato dal rapporto tra la distanza al quadrato del valore x_k dalla media e la devianza di X. notate, nella formula, che avevamo già trovato il valore di leverage nello SE di $\hat{y}_k$.

$$leverage_{x_k} = \frac{(x_k - \bar{X})^2}{\sum(x_i - \bar{X})^2}$$

Ricordiamo, tuttavia, che un alto leverage da solo non necessariamente incide sui coefficienti di regressione: solo quando il caso ha un **alto leverage ed è un outlier bivariato** rispetto al suo valore in Y, **influenzerà fortemente sia** b_0 **sia** b_1**,** configurandosi come un vero *influential case***.** Un outlier bivariato avrà più probabilmente anche alto leverage in campioni piccoli; d'altronde, un caso con alto valore di leverage può avere un residuo piccolo, perché "tira" la retta verso di sé.

Quindi, i valori di **leverage o hat values** h_{i0} stimano la potenziale **influenza di ogni** y_i **su tutti gli** $\hat{y}$ del modello*:* nella regressione semplice, misurano la distanza dalla media di X; nella regressione multipla, misurano la distanza dal centroide[84] delle X. **Variano da $1/N$** (nessuna influenza) **a 1** (influenza decisiva).

Se nessun caso esercitasse un'influenza significativa, ci aspetteremmo *hat value* prossimi all'**hat value medio**, dove N è il numero di soggetti e k è il numero di predittori:

$$\bar{h} = \frac{k+1}{N}$$

Dovrebbero essere **ulteriormente indagati casi che hanno un hat value pari a due volte la media degli hat value** (Welsch, 1978), mentre quelli con un leverage pari a **tre volte il leverage** medio potrebbero essere eliminati dal modello (Stevens, 2002). Con grandi campioni, però, è possibile che questa regola "catturi" casi che potrebbero essere ignorati.

Come altro metodo per individuare gli influential case, possiamo usare la **distanza di Cook**. La distanza di Cook (Cook, 1977) stima l'influenza di ogni singolo caso sulla stima dei parametri, se il caso è tolto dal modello.

$$D_{Cook} = \frac{\sum(\hat{y}_i - \hat{y}_{ii})}{SS_R} \times \frac{N-k-1}{k-1}$$

La distanza è appunto quella esistente tra il parametro calcolato mantenendo nel modello il caso in esame (b_i) e il parametro calcolato senza il soggetto in esame (b_{ii}): le differenze (al quadrato) tra i valori predetti con e senza il caso in oggetto sono sommate, divise per la SS_R e moltiplicate per i df. Se un caso ha una **distanza di Cook > 1**, potrebbe essere un *influential case* (Cook & Weisberg, 1982).

Hatvalues e distanza di Cook sono disponibili in R: `hatvalues(modello)` e `cooks.distance`Errore. Il segnalibro non è definito.`(modello)`; quest'ultima è anche rappresentata in uno dei grafici di diagnostica che vedremo nel §9.5.

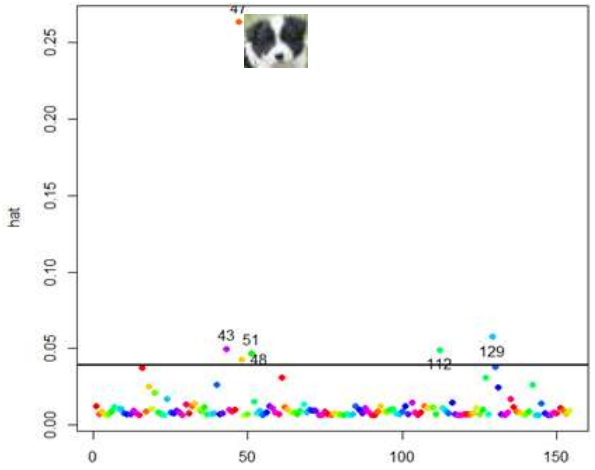
Applichiamo entrambi a `nevro`. Plottiamo prima gli *hatvalues*; nel grafico rappresentiamo anche le linee corrispondenti ai valori corrispondenti a $2 \times media_{hi}$ e a $3 \times media_{hi}$, per evidenziare gli eventuali casi "sensibili".

[84] O centro geometrico: il baricentro della distribuzione delle medie dei vettori X_1, X_2, ...X_k.

```
hat<-hatvalues(nevro)
plot(hat, col=rainbow(15), pch=19, cex=1)
abline(h = mean(hat)*3, lwd=2)
identify(hat)
[1]  43  4748  51  112 129
```

oppure

```
which(hat>3*mean(hat))
43  47  48  51 112 129
 43  47  48  51 112 129
```



I casi con alto valore di leverage sono un bel po', ma **solo uno era anche un outlier bivariato**: la piccola Emy (47) è quindi un ***influential case***.

Il metodo alternativo, la distanza di Cook, dovrebbe portarci a prendere la stessa decisione:

```
summary(cooks.distance(nevro))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0000000 0.0005718 0.0028980 0.0153600 0.0069480 1.5880000
>which(cooks.distance(nevro)>1)
47
 47
```

Sì, decisamente solo la piccola Emy, outlier bivariato e con alto (altissimo!) valore di leverage, è un caso influente.

Come cambiano i parametri del modello, eliminando Emy? Ricordiamo quelli del modello completo:

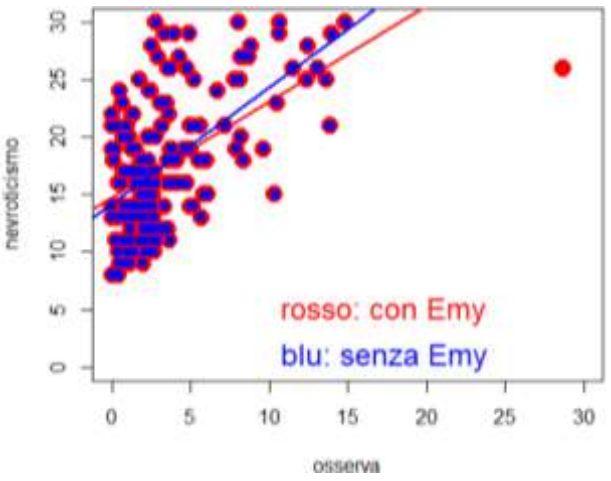
```
nevro$coefficients
     (Intercept) cuccioli$osserva
      14.7726569        0.8373661
```

Togliamo Emy e rifacciamo il modello:

```
senza_emy<-cuccioli[-47, ]
senza<-lm(senza_emy$nevroticismo~senza_emy$osserva)
senza$coefficients
      (Intercept) senza_emy$osserva
        14.210833          1.016771

sum(nevro$residuals^2); sum(senza$residuals^2)
[1] 3817.84
[1] 3595.132
```

Senza il caso influente Amy, lo **slope della retta aumenta e la quota di errore (SS_R) diminuisce**; vediamolo graficamente, rappresentando sovrapposti i plot del campione completo (cerchi in rosso) e senza Amy (quadrati in blu), con i rispettivi modelli:

```
plot(cuccioli$osserva,cuccioli$nevroticismo,
   pch=19, cex=2,col="red", xlab="osserva",
   ylab="nevroticismo, ylim=c(0,30), xlim=c(0,30))
abline(nevro, col="red",lwd=2)
lines(senza_emy$nevroticismo~senza_emy$osserva,
   pch=15, col="blue", type = "p")
abline(senza, col="blue", lwd=2)
text(x=10, y=c(1,5), labels = c("blu: senza
Emy","rosso: con Emy"),pos = 4, col=c("blue",
"red"),cex= 1.5)
```



Il caso influente "tirava" davvero verso di sé la retta di regressione.

La funzione `influence.measures(modello)` fornisce per tutti i casi alcune misure di influenza[85], tra cui l'hat values e la distanza di Cook; potete usarla per avere contemporaneamente le due informazioni:

```
influence.measures(modello)
Influence measures of
         lm(formula = peso ~ altezza) :

    dfb.1_ dfb.altz   dffit cov.r  cook.d hat inf
1  -0.2309  0.24999  0.5090 0.972 0.11904   0.132
2  -0.0859  0.07771 -0.2098 1.345 0.02400   0.116
3   0.0134 -0.00995  0.0810 1.430 0.00372   0.102
4   0.0127  0.00608  0.4423 0.927 0.08933   0.100
5  -0.0164  0.01411 -0.0558 1.452 0.00178   0.107
6  -0.0384  0.03656 -0.0580 1.558 0.00192   0.166
7  -0.7488  0.71211 -1.1303 0.426 0.38060   0.166
8   0.4262 -0.41480  0.4987 1.677 0.13231   0.325
9  -0.2101  0.21764  0.2786 1.649 0.04298   0.257
10  1.5115 -1.54451 -1.7143 1.483 1.22521   0.531
```

Cosa notate in questo output?

Se i casi sono molti, le informazioni di `influence.measures` possono essere **salvate come oggetto**; il `summary` dell'oggetto **evidenzia le osservazioni critiche con un asterisco**: gli **asterischi attribuiti alla distanza di Cook indicano i potenziali casi influenti**:

```
influenti<-influence.measures(nevro)
summary(influenti)
Potentially influential observations of
        lm(formula = cuccioli$nevroticismo ~ cuccioli$osserva) :

    dfb.1_  dfb.ccc$ dffit   cov.r   cook.d  hat
16   0.00   -0.01    -0.01    1.05_*  0.00    0.04
41   0.19   -0.05     0.22    0.93_*  0.02    0.01
43  -0.04    0.11     0.12    1.06_*  0.01    0.05_*
47   1.04_* -1.81_*  -1.83_*  1.22_*  1.59_*  0.26_*
48   0.00    0.01     0.01    1.06_*  0.00    0.04_*
51   0.02   -0.05    -0.05    1.06_*  0.00    0.05_*
105  0.15   -0.02     0.19    0.95_*  0.02    0.01
112  0.09   -0.23    -0.25    1.05_*  0.03    0.05_*
127 -0.02    0.05     0.06    1.04_*  0.00    0.03
129 -0.06    0.13     0.14    1.07_*  0.01    0.06_*
130 -0.04    0.10     0.11    1.05_*  0.01    0.04
144  0.12    0.01     0.18    0.96_*  0.02    0.01
153  0.17   -0.06     0.19    0.96_*  0.02    0.01
```

Attenzione, comunque: Ripetere il modello con e senza influential cases è un modo efficace per stabilire in che modo influenzino il modello, non per cercare di far prevalere il modello desiderato!

9.5 La verifica dei prerequisiti per un GLM

La verifica dei prerequisiti di applicabilità del metodo dei minimi quadrati a un modello lineare generale è essenziale per avere garanzie sulla **replicabilità** del modello, ovvero sulla **possibilità che esso sia generalizzabile** alla popolazione. I prerequisiti sono tanti: distinguiamoli, se non altro per amor di chiarezza, in prerequisiti relativi alla natura delle variabili X e Y e in prerequisiti riferiti agli errori del modello, o test di specificazione.

[85] `Diffbeta`, `diffit` e `covres` sono misure ottenute con il ricampionamento *jacknife*: in breve, indicano rispettivamente il cambiamento nei parametri (.1 per b_0, .X per b_1), nel fit e nella varianza di errore eliminando il caso.

9.5.1 Assunzioni sulle variabili

Dei requisiti relativi alla natura di X e Y e alla loro relazione, tre sono assolti con cautele metodologiche e non statistiche:

a. la distribuzione Y deve essere **metrica**, **continua** e **non tronca**; X può essere quantitativa o categoriale.

b. **indipendenza dei valori Y**: tutti gli y_i derivano da un diverso caso.

c. **Relazione tra X e altre covariate**: il predittore non deve essere correlato con alcun'altra variabile, **esterna al modello**, che potrebbe influenzare Y. Questo requisito, in realtà, è difficilmente verificabile *in toto*, soprattutto in ricerche non sperimentali.

Il quarto requisito pone problemi che potremo efficacemente risolvere nel capitolo 15, con i Multilevel o mixed models:

d. **l'effetto del predittore è stesso per tutti i soggetti**: le differenze tra gli effetti, per tutti i soggetti sottoposti a uno stesso livello di X, sono prossime a zero. Questo è un assunto problematico nella realtà, in cui l'impatto di X su Y è sottoposto a diverse forme di modulazione a opera di **covariate**, non tutte controllabili metodologicamente o statisticamente (analisi della covarianza, §13.4).

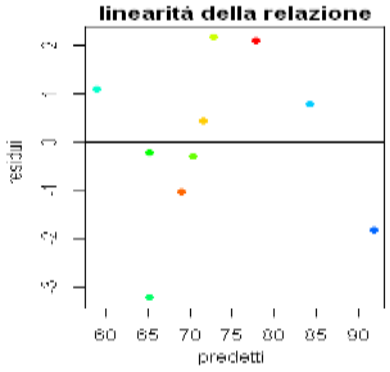
Il quinto invece, pare scontato, ma va verificato:

e. **Relazione tra Y e X**: la relazione bivariata tra Y e X deve essere **lineare**.

La linearità della relazione tra Y e X si verifica plottando in Y i residui standardizzati e in X i valori predetti del modello: se la relazione fosse lineare, i punti dovrebbero simmetricamente attorno alla linea orizzontale di riferimento con $b_0 = 0$ senza manifestare una conformazione riconoscibile ("nube di punti").

Vediamo com'è la linearità della relazione tra peso e altezza:

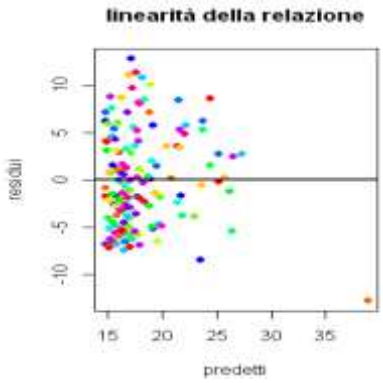
```
pesoaltezza<-lm(peso~altezza)
plot(predict(pesoaltezza), rstandard(pesoaltezza), pch=19, col=ra
 inbow(15), xlab="predetti", ylab="residui", main="linearità dell
 a relazione")
abline(h = 0, lwd=2)
```



La disposizione è soddisfacente.

Vediamo il modello `nevro`:

```
plot(predict(nevro), rstandard(nevro), pch=19, col=rainbow(15), x
 lab="predetti", ylab="residui",
 main="linearità della relazione")
abline(h = 0, lwd=2)
```



Abbastanza buono, a parte la presenza di sappiamo chi...

E se la relazione non è lineare? Beh, non si applica un modello lineare... nel capitolo 14, vedremo qualche esempio di *smoothing* (per regressioni non parametriche) e approfondiremo uno specifico esempio di modello **generalizzato** applicabile a relazioni non lineari.

9.5.2 Assunzioni sui residui, o test di specificazione

I residui descrivono la variabilità di Y indipendentemente dall'influenza di X, quindi indicano come dovrebbe essere distribuita Y sotto condizione di H_0. Affinché le assunzioni su cui si basa la regressione con il metodo dei minimi quadrati reggano, è essenziale che gli **errori di predizione del modello rispettino tre caratteristiche** fondamentali:

1. **Indipendenza**: per ogni **coppia di osservazioni** x_i **e** x_{i+1}, i rispettivi **residui non devono essere correlati**, cioè il valore del residuo e_i deve essere indipendente dal residuo di ogni altro caso e_j della distribuzione. Se i residui sono indipendenti, non è riconoscibile al loro interno una partizione in gruppi (**cluster**) al cui interno i residui sono più simili tra loro di quanto siano simili ai residui di altri cluster. La dipendenza può essere quantificata dal **coefficiente di autocorrelazione** (di primo ordine) **o correlazione seriale**: se $r_{seriale} = 0$, i residui sono indipendenti; quanto più il coefficiente tende a 1, in direzione positiva o negativa, tanto più i residui tenderanno ad avere lo stesso segno (+ o -) passando da un caso all'altro della distribuzione. La violazione del requisito non incide sulla stima dei parametri (Durbin, 1950), ma sul calcolo della MS_R e quindi su F: la MS_R campionaria non corrisponde alla varianza dei residui del modello atteso in popolazione (il cosiddetto **modello generatore dei dati**: la sua varianza è σ_R^2, ne riparleremo nel capitolo 11), e questo *bias* su F può portare, in maniera impredicibile, a un incremento dell'errore di I o di II tipo. Nel deprecato caso che la correlazione tra i residui non sia trascurabile, potremmo gestire la dipendenza inserendo nel modello, come X, la variabile che determina il raggruppamento in cluster dei residui (variabile contestuale), ammesso che si sia interessati al suo effetto, naturalmente; in caso contrario, meglio cambiare tipo di modello e adottare un mixed model, che gestisce la dipendenza tra i residui (capitolo 15).
2. **Normalità**: la distribuzione degli errori dev'essere normale. Il problema della mancata normalità (e dell'eteroschedasticità, punto 3.) è che in questa condizione la quantità di errore del modello non è costante per tutta la distribuzione del predittore X, e quindi il potere predittivo di X è diverso a seconda del valore in Y cui si applica: almeno teoricamente, il predittore indica "cose" diverse per i diversi livelli di Y. Quando la normalità è rispettata (o, quantomeno, quando non è gravemente violata), la media degli errori, che si compensano reciprocamente sopra e sotto il valore centrale, non è significativamente diversa da 0: la differenza tra il modello e i dati reali è perlopiù uguale o equivalente a 0, e le **differenze $\neq 0$ compaiono solo casualmente**. È tradizione affermare che il modello lineare è **robusto** alla violazione della normalità dei residui, ovvero che le conclusioni su H_0 derivanti da F e rispettivo $p - value$ non sono gravemente distorte da errori non normali. Tuttavia, se a questa violazione si aggiunge anche quella dell'omoschedasticità (punto 3), è meglio non fidarsi troppo di questa presunta "robustezza" del modello.
3. **Omoschedasticità**: la varianza degli errori deve essere costante per tutti i valori della distribuzione X, cioè deve essere indipendente dal valore predetto loro associato dal modello lineare. È probabilmente il requisito più importante da rispettare (Seher e Lee, 2003): come nel caso della non normalità, la sua violazione (eteroschedasticità) descrive un potere predittivo di X diverso a seconda del valore Y, e mette in discussione la sola casualità degli errori presenti nel modello, suggerendo che sia all'opera una fonte di errore sistematico. In caso di eteroschedasticità, può essere tentata una **trasformazione non lineare di Y**, oppure si può adottare un approccio di regressione non parametrico.

Vediamo come eseguire i test di specificazione sui residui con R.

Per verificare che la media non sia significativamente $\neq 0$, si può applicare il t-test per un campione (§6.7.1.) ai residui, sperando che il $p - value$ sia $p > .05$: `t.test(residuals(modello))`.

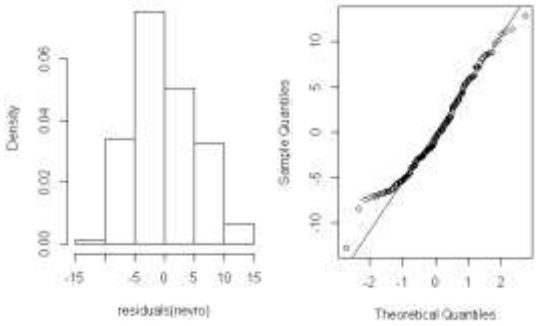
```
t.test(residuals(nevro))
        One Sample t-test
data:  residuals(nevro)
t = 4.3431e-17, df = 153, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.7952426  0.7952426
sample estimates:
   mean of x
1.748241e-17
```

La media degli errori è = .00000000000000001, non significativamente diversa da 0.

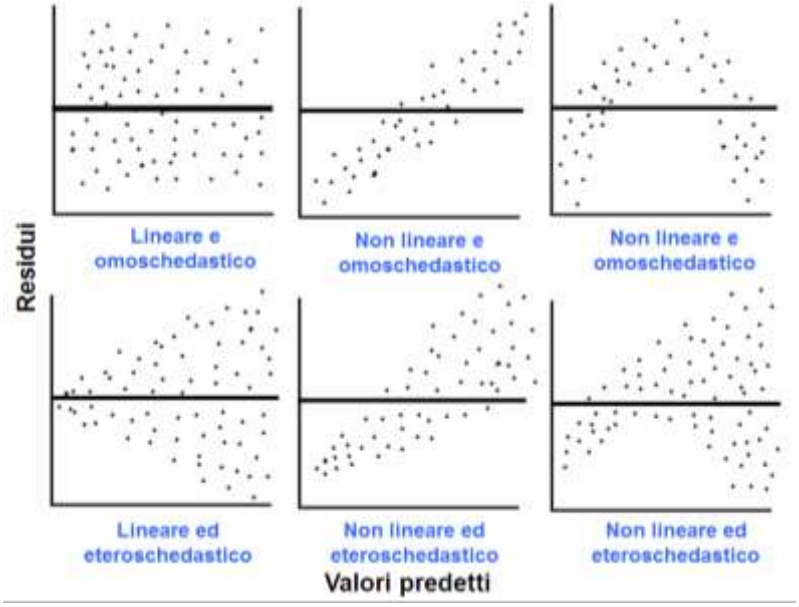
Per verificare la forma della distribuzione degli errori rispetto alla normale, possiamo usare il Q-Q plot /o l'istogramma, nonché il test di Shapiro, già affrontati:

```
shapiro.test(residuals(nevro))
         Shapiro-Wilk normality test
data:  residuals(nevro)
W = 0.97346, p-value = 0.004517
```

Il test è significativo (anche se osserviamo che W tende a 1 e il test è potente: $N = 154$), con curtosi negativa.



Lo stesso plot predetti – residui usato per valutare la linearità $Y \sim X$ consente di valutare anche **l'omoschedasticità degli errori**: se è rispettata, la dispersione verticale degli errori attorno alla linea di riferimento dovrebbe essere costante per tutti i valori di X. In caso di **eteroschedasticità**, si verificherebbe una configurazione a **imbuto** (***funnel***): per esempio, le distanze degli errori dalla linea di riferimento potrebbero essere ridotte per bassi valori di Y e maggiori per alti valori di X. Vediamo qualche configurazione di esempio:



Per verificare l'ipotesi nulla di omoschedasticità, si possono usare vari test inferenziali, tra cui il **test di Breusch – Pagan** (1979; AKA Breusch-Pagan-Godfrey test): il test stima se la **varianza della distribuzione degli errori** di un modello **dipende dall'effetto del predittore** (o dei predittori) inserito nel modello. Si effettua quindi una regressione lineare **ausiliaria** sui residui (al quadrato) del modello, usando gli stessi predittori: se i predittori spiegano una buona quantità della varianza dei residui, i residui non sono indipendenti dal predittore e l'omoschedasticità è violata.

La statistica del test si calcola molto semplicemente: $\chi^2_{bP} = N \times R^2_{ausiliaria}$. χ^2_{bP} si distribuisce come un quantile χ^2, per $df = numero\ di\ X$ nel modello, per cui è possibile attribuirle un $p - value$: se cade nella zona di fiducia di H_0, è rispettata l'**omoschedasticità**. Il risultato del test è, però, pienamente affidabile solo se la distribuzione dei residui è normale.

Vediamo la logica dell'analisi, poi useremo funzioni dedicate: cominciamo a creare il **vettore dei residui** del modello `nevro`:

```
residui<-nevro$residuals
```

Lo inseriamo, al quadrato, come Y in un modello lineare con lo stesso predittore usato per `nevro`:

```
summary(lm(residui^2~osserva, data=cuccioli))
Call:
lm(formula = residui^2 ~ osserva, data = cuccioli)

[ omissis]

Residual standard error: 30.4 on 152 degrees of freedom
Multiple R-squared:  0.02626,    Adjusted R-squared:  0.01985
```

Moltiplichiamo R^2 per N, ottenendo la statistica di Breusch-Pagan, e assegniamo un $p - value$ a questo valore o a uno più estremo con la funzione di ripartizione `pchi(chi, df, lower.tail=FALSE)`:

```
round(0.02626*154, 2); pchisq(q = 4.04, df = 1, lower.tail = F)
[1] 4.04
[1] 0.0443382
```

Male, siamo sotto la soglia alfa: l'omoschedasticità non è garantita.

D'ora in poi, potremo usare `bptest(modello)` di **`lmtest`**, che riporta la statistica e il $p - value$ (anche se, in casi dubbi come questo, calcolare l'R^2 del modello di regressione ausiliare può aiutare a decidere sulla gravità della violazione all'omoschedasticità):

```
bptest(nevro)
        studentized Breusch-Pagan test
data:  nevro
BP = 4.0435, df = 1, p-value = 0.04434
```

Notate che il test è definito `studentized`: non preoccupatevene troppo, basti sapere che oltre ai residui standardizzati come z di cui abbiamo parlato prima, è possibile trasformare diversamente i residui in quantili t o residui **studentizzati**, considerando anche il loro valore di *leverage* nella trasformazione: in questa forma, il risultato del test è uguale a quello ottenuto con la regressione ausiliaria. Se però indichiamo l'argomento `studentized = FALSE`:

```
bptest(nevro, studentize = FALSE)
        Breusch-Pagan test
data:  nevro
BP = 3.0808, df = 1, p-value = 0.07922
```

, la statistica BP si riduce e il suo $p - value$ cade nella regione di fiducia di H_0, a ulteriore dimostrazione che quando il $p - value$ è a soglia, sopra o sotto che sia, nessuna decisione su H_0 acriticamente dicotomica è corretta.

Se volete usare package già noti, potete usare **`car`** e `ncvTest(modello)`: `ncv` sta per *Non Costant Variance*, e la statistica χ^2 nell'output è appunto la statistica del test di Breusch-Pagan **non** studentizzato.

```
ncvTest(nevro)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 3.080751, Df = 1, p = 0.079224
```

Abbiamo creato in precedenza il modello `senza`, eliminando dal campione l'outlier Emy: verificate come si comporta la varianza dei residui di questo modello, confrontandola con quella di `nevro`, e meditate sull'effetto degli outlier.

Il requisito di **indipendenza degli errori** si valuta con il test di **Durbin – Watson**. Il test (nella formula, e_i indica il residuo) valuta la presenza di autoregressione (o autocorrelazione, o correlazione seriale) di primo ordine (AR1), che è appunto il tipo di correlazione che temiamo tra i residui: quella tra residui contigui.

$$DW = \frac{\sum_{i=2}^{N}(e_i - e_{i-1})^2}{\sum_{i=1}^{N} e_i^2}$$

La statistica d del test varia da 0 a 4, con un **punto centrale = 2 che indica assenza di autocorrelazione**; in genere, tra 1.5 e 2.5 l'assenza di autocorrelazione è sufficientemente garantita. Allontanandosi da 2, la stima dell'autocorrelazione è via via più intensa nelle due direzioni: tra 2 e 4, tra 2 e 0. Possiamo usare **`dwt(modello)`** di **`car`**, **`dwtest(modello)`** di **`lmtest`** o **`DurbinWatsonTest(modello)`** di **`DescTools`**: solo nell'output di `dwt` troviamo anche il valore di autocorrelazione, che dovrebbe tendere a 0, mentre in tutti e tre è riportata la statistica DW con relativo $p-value$, che ci auguriamo **non significativo** a indicare assenza di **autocorrelazione** .

```
dwt(nevro)
 lag Autocorrelation D-W Statistic p-value
   1      0.08409876      1.829043   0.256
 Alternative hypothesis: rho != 0
dwtest(nevro)
       Durbin-Watson test
data:  nevro
DW = 1.829, p-value = 0.1401
alternative hypothesis: true autocorrelation is greater than 0
DurbinWatsonTest(nevro)
       Durbin-Watson test
data:  nevro
DW = 1.829, p-value = 0.1401
alternative hypothesis: true autocorrelation is greater than 0
```

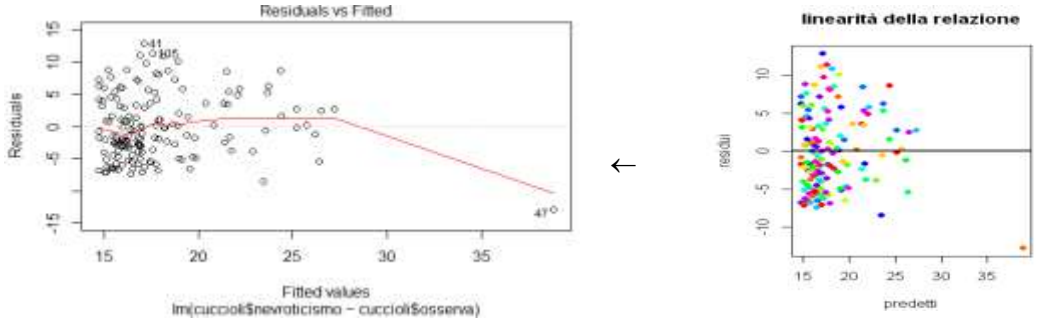
Per fortuna, l'autocorrelazione è scongiurata: la statistica DW è compresa tra 1.5 e 2.5, e il $p-value$ conferma che non è significativamente diversa da 2.

Notate che, di default, H_1 è bidirezionale in `dwt`, monodirezionale destra in `dwtest` e `DurbinWatsonTest`.

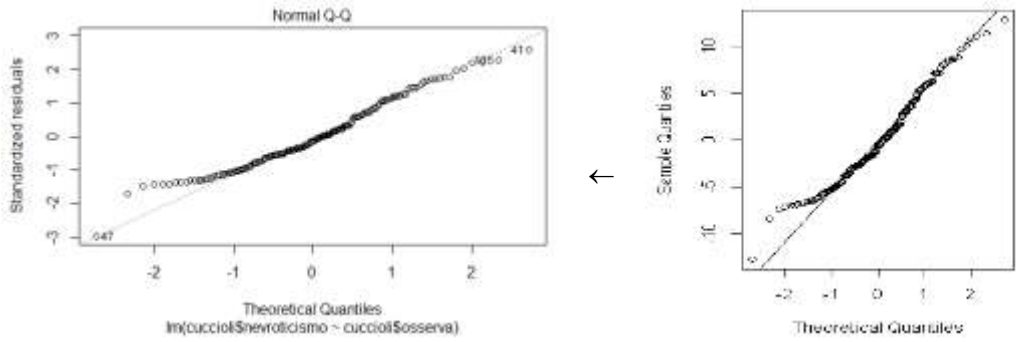
Gli amanti dei grafici possono "semplicemente" usare i grafici prodotti da **`plot(modello)`** per verificare le assunzioni sui residui: `plot(modello)` produce **quattro grafici diagnostici** di seguito:

```
plot(nevro)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

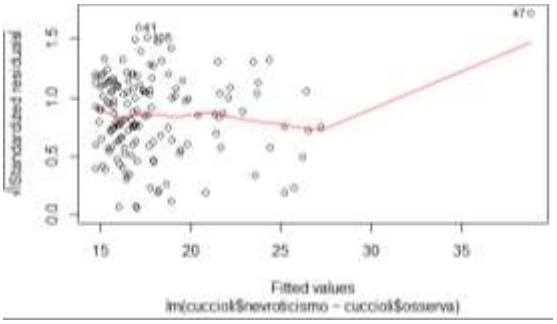
Il primo è il grafico predetti – residui che consente di stimare la linearità della relazione $Y \sim X$ e l'omoschedasticità degli errori: ci auguriamo di vedere nubi di punti senza configurazioni riconoscibili.



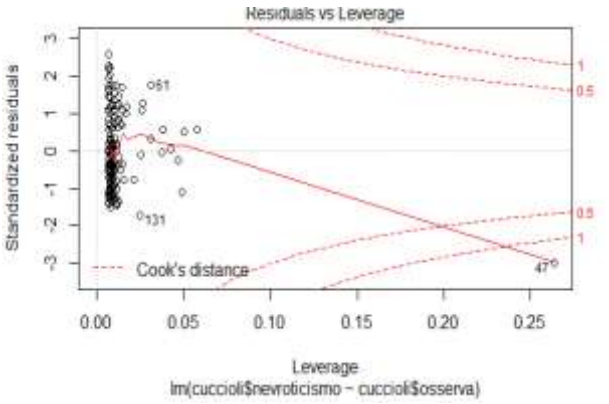
Il secondo è il Q-Q plot per la normalità della distribuzione degli errori, che abbiamo già visto:



Il terzo mostra se i residui sono equamente dispersi lungo la distribuzione dei predittori, usando in ordinata la radice quadrata dei residui: **consente di individuare probabili outlier e valutare l'omoschedasticità** (i punti dovrebbero disporsi casualmente attorno alla retta):



Infine, il quarto plot **consente di individuare la presenza di casi influenti**: in ascissa troviamo il valore di leverage, in ordinata il residuo standardizzato di ogni caso. I casi con alto valore di leverage in ascissa ed estremi residui standardizzati in alto o in basso (outlier bivariati) sono casi influenti. Per facilitarne l'individuazione, attorno alla nube dei residui sono tracciate le righe che separano i punti disposti entro una $Cook\ _{distance} < 1$ da quelli con $Cook\ _{distance} > 1$, che sono molto probabilmente influential cases:

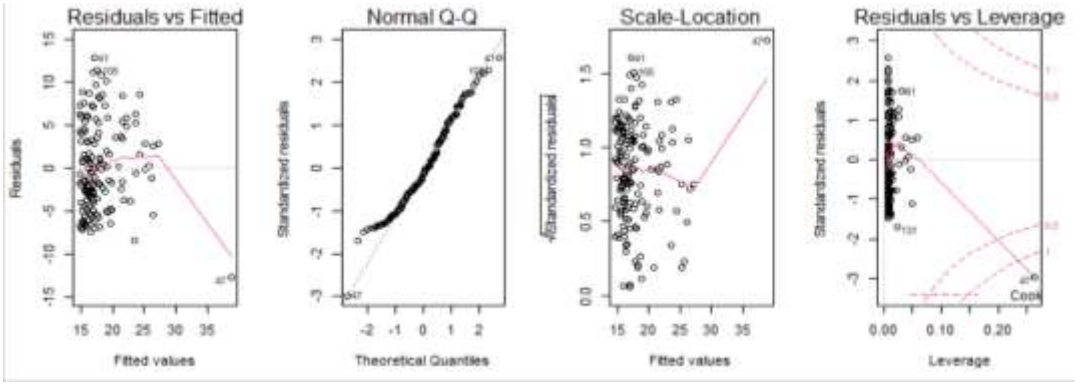


Per visualizzarli in un'unica finestra, invece di averli come quattro grafici consecutivi, ricordiamo la funzione che ripartisce la finestra dei grafici per righe (o colonne) vista nel capitolo 4:

```
par(mfrow=c(1,4))
```

Torniamo a un'unica finestra:

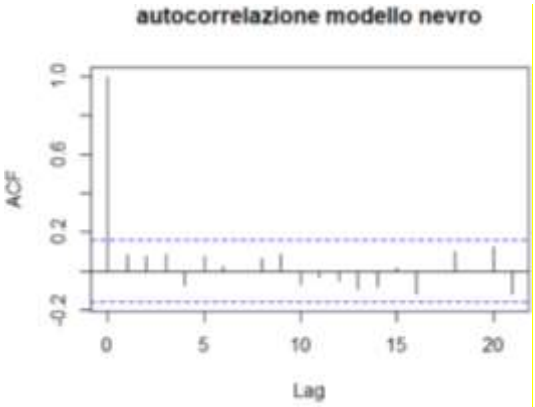
```
par(mfrow=c(1,1))
```



Avrete notato che l'**autocorrelazione** non si legge nei grafici diagnostici, ma si può rappresentare con la funzione di base `acf(residuals(modello))`. In Y è indicato il valore della correlazione tra i residui di ogni **coppia di osservazioni x_i e x_{i+1}**, in X sono disposte le diverse coppie per cui è calcolata (*lags*). Il primo residuo è correlato con se stesso, quindi la prima barra arriva a 1, mentre **tutti gli altri dovrebbero avere barre lontane da 1 e prossime a 0**.

Nel grafico è **evidenziata l'area delle autocorrelazioni trascurabili** (tra -.2 e +.2), entro cui dovrebbero stare tutte le autocorrelazioni della distribuzione dei residui. Nel modello `nevro`, questa condizione è ben rispettata:

```
acf(residuals(nevro), main= "autocorrelazione modello
   nevro")
```



Il package `performance` contiene funzioni per una serie di test e coefficienti di fit per modelli di regressione semplice e multipla, agevolando al massimo la lettura degli output; ne presentiamo qui alcune, ma suggerendo estrema cautela nel dare fiducia a output tanto banali, che spingono a una interpretazione piuttosto superficiale.

Per rilevare la **presenza di casi influenti** si usa `check.outliers(lm(Y~X, data=dataframe), method= "cook")`; in `method=` specifichiamo `"cook"` per indicare quale valore di influenza utilizzare. Nell'output, un warning, in minaccioso colore rosso, indica il numero di riga del caso sospetto, se rilevato.

```
check_outliers(nevro, method= "cook")
Warning: 1 outliers detected (cases 47).
check_outliers(lm(amicalita~gioca, data= cuccioli), method="cook")
OK: No outliers detected.
```

La stessa funzione serve anche a rilevare **outliers univariati**, trasformando la variabile in punti z.

```
check_outliers(cuccioli$nevroticismo)
Warning: 10 outliers detected (cases 40, 41, 43, 61, 79, 105, 129, 142, 144, 145).
```

Passando ai test di specificazione, per la normalità della distribuzione dei residui `check_normality(modello)` produce solo il $p - value$ del test di Shapiro e l'interpretazione da attribuirgli.

```
check_normality(nevro)
Warning: Non-normality of residuals detected (p = 0.006).
```

L'omoschedasticità è testata da `check.heteroshedasticity(modello)`, su residui non studentizzati:

```
check_heteroscedasticity(nevro)
OK: Error variance appears to be homoscedastic (p = 0.079).
```

Infine, `check_autocorrelation(modello)` produce il test di Durbin – Watson riportando solo il *p*-value (per H_1 solo bidirezionale), con la spiegazione della sua interpretazione:

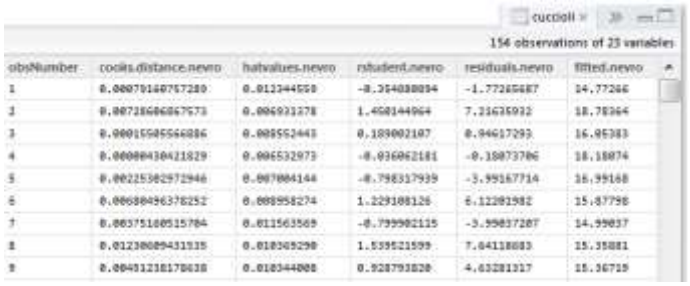
```
check_autocorrelation(nevro)
OK: Residuals appear to be independent and not autocorrelated (p = 0.280).
```

Con `Rcommander` si può impostare facilmente una funzione di regressione semplice. Una volta caricato il dataframe, si sceglie `Regressione lineare` nel menu `Statistiche`.

Nella finestra si seleziona la variabile di risposta Y e la variabile esplicativa X (per ora ne usiamo una sola); potete dare un nome al modello. Se dovete fare la regressione solo su una parte del campione, si indica la variabile filtro (ad esempio: `cuccioli$razza=="border"`) nell'espressione di selezione:

Nella sezione dei risultati viene stampato il `summary` del modello.

Se avete salvato il modello come oggetto, nel menu Modelli potete trovare molte analisi disponibili; alcune le useremo solo nella regressione multipla. Per ora, limitiamoci ad **`Aggiungi le Statistiche…`**:


cuccioli

156 observations of 23 variables

obsNumber	cooks.distance.nevro	hatvalues.nevro	rstudent.nevro	residuals.nevro	fitted.nevro
1	0.00079148757280	0.012344559	-0.354088894	-1.77265687	14.77266
2	0.00718686867673	0.006921378	1.458144964	7.21635932	18.78364
3	0.00015585566886	0.008552443	0.189002107	0.94617293	16.05383
4	0.00000430421829	0.006532973	-0.036062181	-0.18073706	18.18074
5	0.00225362572946	0.007004144	-0.798317939	-3.99167714	18.99168
6	0.00680496376252	0.008958274	1.229108126	6.12281982	15.87718
7	0.00375160515784	0.011563569	-0.799902115	-3.99837287	14.99837
8	0.01230089431535	0.010369298	1.559521599	7.84118683	15.35881
9	0.00451218178638	0.010344008	0.928793820	4.63281317	15.36719

Questo menu aggiunge al dataframe `cuccioli` il valore predetto, l'errore, il valore di *leverage*, la distanza di Cook ecc. per ogni caso. Le nuove variabili sono ora disponibili per ulteriori osservazioni, ad esempio per evidenziare casi influenti.

Intervalli di confidenza operazione calcola `confint(modello)`:

Nel menu `Diagnostici numerici` troviamo il test di Breush-Pagan per l'eteroschedasticità e quello di Durbin-Watson per l'autocorrelazione. Nel menu `Grafici` potete ottenere il modello di regressione (grafico degli effetti) e i grafici diagnostici per la verifica dei requisiti del modello.

*Una buona empatia può predire la **qualità della relazione** con un gatto? Verificate l'ipotesi prima usando l'empatia verso – genericamente – gli animali stimata dalla scala AES, poi usando l'empatia verso i gatti; almeno in questo secondo caso, valutate anche la presenza di influential cases e discutete il rispetto dei prerequisiti del modello.*

*Una buona empatia verso i gatti può predire la capacità **di riconoscere correttamente le intenzioni comunicative** del micio?*

- *Usate il campione complessivo per valutare l'impatto dell'empatia sul totale dei riconoscimenti corretti e commentate il dato;*
- *Rifate la stessa analisi separatamente per i soggetti che vivono con un gatto e per quelli che non vivono con un gatto: commentate ciascuno dei due risultati e confrontate quanto ottenuto con l'analisi riferita al punto precedente. Cosa notate? Come potete spiegare quello che (si spera) dovreste notare ☺ ?*

Capitolo 10

Distribuzioni bivariate: un solo predittore categoriale a due livelli, Y continua

In questo capitolo useremo i dataframe ***karate***, ***vecchietti*** *e* ***sicurezza*** *pubblicati su Elly; prima di proseguire con la lettura, eseguite:*

1. apriteli e leggetene la descrizione nei file associati;

2. descrivete la composizione dei campioni nei tre dataframe

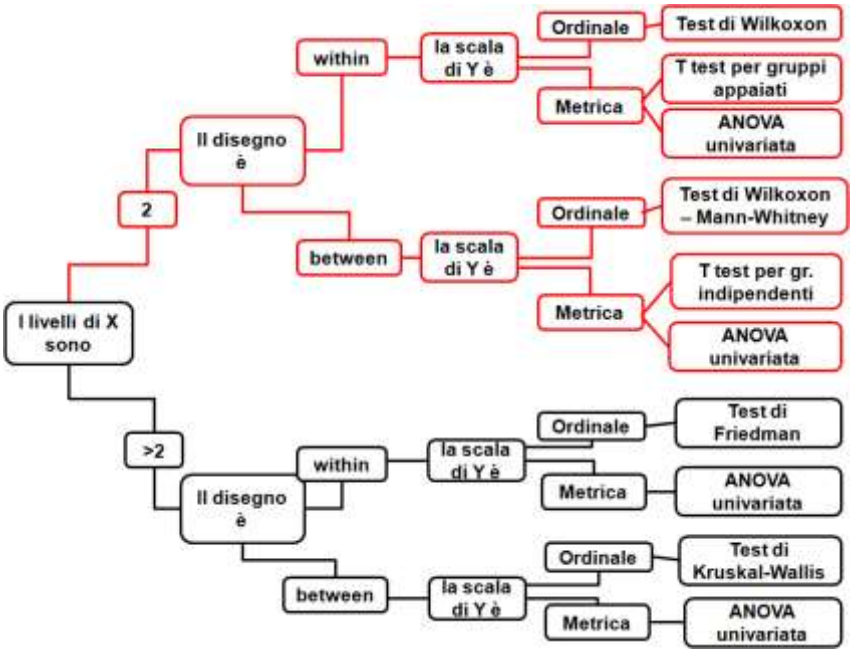
Quando il modello lineare vede una Y continua e (almeno) un **predittore X**, **categoriale** ad **almeno due livelli**, **l'H_0 di non relazione tra X e Y** può essere "tradotta" nell'equivalente **H_0** di **non differenza tra le medie dei due livelli di X**, che costituiscono i **gruppi** (disegni **between groups**) o le **condizioni** (disegni **within subjects**) del disegno sperimentale.

Infatti, se il predittore X non esercita alcun effetto su Y, allora le differenze tra la media di X_1 e la media di X_2 sono solo oscillazioni casuali attorno alla *grand mean*, ovvero alla media della popolazione da cui sono tratti i soggetti: i casi dei **due campioni / livelli di X appartengono alla medesima popolazione**.

Questo tipo di H_0 si applica a misure **metriche e ordinali** raccolte in:

- disegni **sperimentali**, in cui i soggetti sono casualmente estratti dalla stessa popolazione e casualmente assegnati a N gruppi o condizioni indipendenti, corrispondenti ai diversi livelli di X (ad esempio, **sperimentale** versus **controllo**);
- disegni **osservazionali** (o **quasi sperimentali** o per **gruppi non equivalenti**), in cui i soggetti sono casualmente estratti da popolazioni ritenute differenti (ad esempio, popolazione **clinica** versus **popolazione non clinica**).

L'algoritmo per l'individuazione del corretto test inferenziale è:



In questo capitolo ci occuperemo solo del "ramo" superiore del diagramma (in rosso); nel capitolo 11 termineremo con il ramo inferiore (in nero).

10.1 Test per disegni between groups

L'H_0 cui si applicano i test di questo paragrafo è che i soggetti (che si dà per assunto siano estratti in maniera randomizzata), assegnati al gruppo definito dal livello X_1 del fattore X (ad esempio, gruppo sperimentale) **o** al gruppo definito dal livello X_2 del fattore X (ad esempio, gruppo clinico), appartengano a **una medesima popolazione**: questo comporta che le **medie $\bar{Y}_1$ e $\bar{Y}_2$** rilevate nei due gruppi sono **equivalenti,** ovvero **solo casualmente differenti**. Pertanto, il **fattore X non esercita un effetto significativo su Y**, dato che non rende differenti i due gruppi. H_1 può essere espressa in forma bidirezionale o monodirezionale, tranne che nell'analisi della varianza, che accetta solo H_1 bidirezionali.

10.1.1 Analisi della varianza (ANOVA) a una via, per gruppi indipendenti , a due livelli

> *"The separation of the variance ascribable to one group of causes from the variance ascribable to other groups"*
>
> *(Fisher, 1925;1970[86])*

L'analisi della varianza è nient'altro che un modello lineare generale che applica a **una Y metrica** e **continua** l'effetto di **un predittore X categoriale**; nel caso più semplice da cui partiamo, i **livelli** di X sono solo **due** e costituiscono due **gruppi indipendenti**: i soggetti che appartengono a X_1 (vivo, promosso, ingegnere) sono diversi da quelli che appartengono a X_2 (morto, bocciato, architetto). Vedremo altre applicazioni di ANOVA nel caso di modelli per disegni within subjects con una X a due livelli (§10.2), per X a più di due livelli (Capitolo 12) e per più X (Capitolo 13).

- $H_0: \mu_1 = \mu_2$
- $H_1: \mu_1 \neq \mu_2$

Nel modello lineare applicato a questo tipo di relazione, la variabile dicotomica X diventa una **fittizia variabile quantitativa** (**dummy**), in cui il livello X_0 (livello di riferimento, o base) assume valore **0** e la categoria X_1 (livello di confronto) assume valore **1**. I soggetti che appartengono al gruppo X_0 si vedono pertanto assegnato 0 nella distribuzione X; quelli che appartengono al gruppo X_1 si vedranno assegnare invece 1.

Il punteggio del soggetto i del livello / gruppo j di X… … *più la quota di errore del modello riferita al soggetto ij.*

$$y_{ij} = (b_0 + b_1 G_j) + e_{ij}$$

… è dato dalla media del gruppo di controllo "0" … *più l'effetto dovuto alla sua appartenenza al livello/gruppo j*

Chiariamo la logica del fattore dummy con un esempio tratto da Glaz e Slinker[87], che usano dati immaginari (!!) riferiti a marziani in gita sulla Terra. Si vuole verificare se il livello di nausea, valutato con una misura convenzionale (il **burp**) in un gruppo di **5marziani esposti al fumo di sigarette** ($\boldsymbol{sperimentali = X_1}$) è diverso da quello di un gruppo di **3 marziani presi come controllo** ($\boldsymbol{controllo = X_0}$).

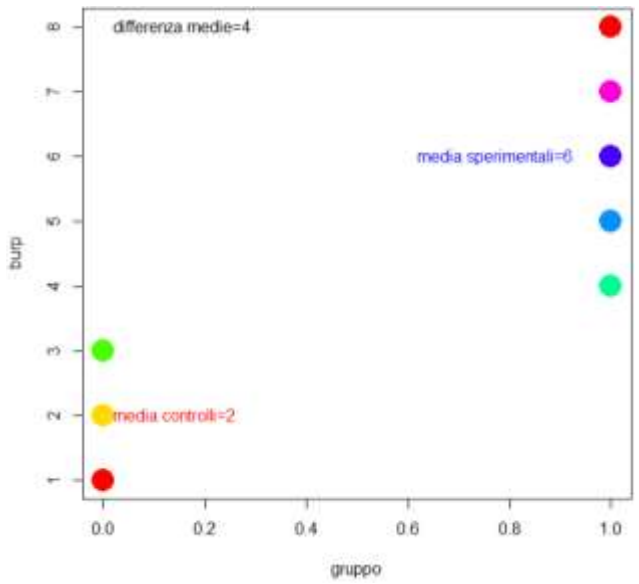
```
marziano<-c("M1", "M2", "M3","M4", "M5", "M6", "M7", "M8")
gruppo<-c(0,0,0,1,1,1,1,1)
burp<-c(1,2,3,4,5,6,7,8)
marziani<-data.frame(marziano, gruppo, burp)
```

[86] *Statical methods for research workers,*1952/1970, 14ª ed., pag. 213.
[87] *Primer of applied regression and analysis of variance,* McGraw-Hill, 2001.

```
marziani
  marziano gruppo burp
1       M1      0    1
2       M2      0    2
3       M3      0    3
4       M4      1    4
5       M5      1    5
6       M6      1    6
7       M7      1    7
8       M8      1    8
class(gruppo)
[1] "numeric"
(media_controlli<-(1+2+3)/3)
[1] 2
(media_sperimentali<-(4+5+6+7+8)/5)
[1] 6
(differenza_medie<-media_controlli-media_sperimentali)
[1] -4
```

La media del gruppo sperimentale X_1 è tre volte superiore a quella del gruppo di controllo X_0: sembra che il fumo di sigaretta aumenti davvero la nausea. Rappresentiamo i dati: invece di usare `boxplot(Y~X)` o `plotmeans(Y~X)` come forse saremmo tentati di fare, usiamo lo scatterplot del capitolo precedente:

```
plot(marziani$gruppo, marziani$burp,
col=rainbow(7), pch=19, cex=3, xlab="gruppo",
ylab="burp")
text(x=c(0,.6,0), y=c(2,6,8), labels = c("media
controlli=2", "media sperimentali=6", "differenza
medie=4"), pos = 4, col=c("red", "blue","black"))
```



Ora **adattiamo un modello lineare** alla relazione tra Y (continua) e X (dummy) con `lm`, e **vediamone i parametri**:

```
mars<-lm(marziani$burp~marziani$gruppo)
mars
Call:
lm(formula = marziani$burp ~ marziani$gruppo)

Coefficients:
    (Intercept)  marziani$gruppo
              2                4
```

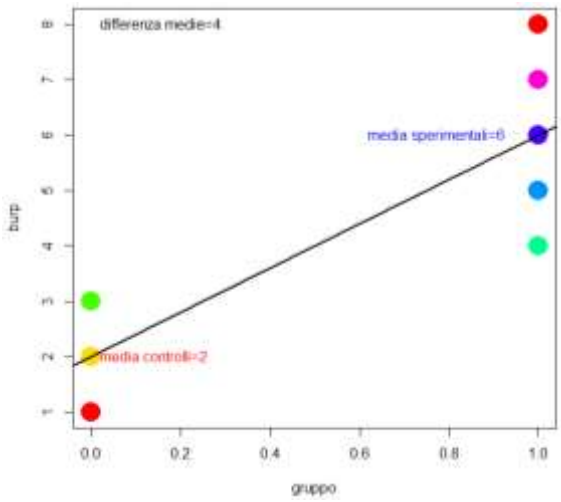
$b_0 = 2$ → **il valore in Y quando $X = 0$**, quando X è dummy, corrisponde **alla media in Y del livello H_0** → corrisponde alla media del **gruppo di controllo**, quello **in cui $X = 0$**, cioè non agisce la variabile indipendente. b_0 è considerata una **stima della misura in popolazione** e si definisce per questo ***grand mean***.

$b_1 = 4$ → **la variazione unitaria in Y al variare di una unità in X**, quando X è dummy, corrisponde **alla variazione media in Y passando dal** livello X_0 al livello X_1, cioè alla **differenza tra le medie** di X_0 e X_1→ il **segno** indica se passando da X_0 a X_1 la **media si alza** (+: nell'esempio, i marziani sperimentali hanno in media più nausea) o si abbassa (-).

Vediamo l'intero `summary` e i CI; aggiungiamo anche la retta di regressione nel grafico:

```
summary(mars)
Call:
lm(formula = marziani$burp ~ marziani$gruppo)
Residuals:
   Min     1Q Median     3Q    Max
    -2     -1      0      1      2
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.0000     0.8165   2.449  0.04983
marziani$gruppo   4.0000     1.0328   3.873  0.00824
---
Residual standard error: 1.414 on 6 degrees of freedom
Multiple R-squared:  0.7143,    Adjusted R-squared:  0.6667
F-statistic:    15 on 1 and 6 DF,  p-value: 0.008237


round(confint(mars),3)
                2.5 % 97.5 %
(Intercept)     0.002  3.998
marziani$gruppo 1.473  6.527
```

La media del gruppo di controllo (b_0) è =2 *burp* ed è significativamente ≠0. Passando dal gruppo di controllo al gruppo sperimentale, la media della nausea si alza di 4 *burp*, e la variazione è significativa, come attestato dal $p - value$ del t-test, da quello del rapporto F e dal fatto che il CI (parecchio ampio, peraltro) nella popolazione del pianeta rosso non comprende 0. La variabilità nella nausea spiegata dall'appartenenza al gruppo è 15 volte più grande della variabilità attribuita all'errore; per la precisione, l'appartenenza al gruppo spiega il 71.4% della variabilità della nausea nel campione (la stima scende al 66.7% in popolazione).



La retta di regressione passa per la media di X_0 e X_1**.**

Come si è arrivati al rapporto F? Il calcolo delle SS e delle MS prevede le **stesse formule** viste nel Capitolo 9:

$SS_M = \sum(\hat{y}_i - \bar{Y})^2 \rightarrow$ `SS_M<-sum((predict(mars)-mean(marziani$burp))^2)`

$SS_R = \sum(y_i - \hat{y}_i)^2 \rightarrow$ `SS_R<-sum((marziani$burp-predict(nausea))^2)`

$SS_T = \sum(y_i - \bar{Y}_i)^2 \rightarrow$ `SS_T<-sum((marziani$burp - mean(marziani$burp))^2)`

Con i ***df*** di SS_M (numero di gruppi meno 1: $\boldsymbol{k-1}$) e di SS_R (numero di casi meno numero di gruppi: $\boldsymbol{N-k}$), si ottengono MS_M e MS_R:

```
MS_M<-SS_M/1
MS_R<-SS_R/(8-2)
```

E dalle due varianze il rapporto F:

```
SS_M;SS_R;SS_T  MS_M;MS_R   F<-MS_M/MS_R
[1] 30          [1] 30      [1] 15
[1] 12          [1] 2
[1] 42
```

Attenzione, comunque: nel caso di un'ANOVA per gruppi indipendenti, SS_M e MS_M corrispondono rispettivamente a ***SS*** e ***MS*** di Y **tra i gruppi** (***be****tween groups o sperimentale:* SS_b ***e*** MS_b), mentre SS_R e MS_R corrispondono a ***SS*** e ***MS*** di Y **entro il gruppo** (*within group o d'errore:* SS_w *e* MS_w). Infatti, la variabilità di Y **tra** i gruppi, cioè attribuita all'effetto della variabile X, è messa in rapporto alla variabilità di Y **all'interno** di ciascun gruppo, quella cioè attribuibile alle diversità

tra i soggetti che appartengono a ciascun gruppo e non sono dovute alla loro appartenenza a un differente gruppo. Perciò, se la varianza in Y tra i gruppi (MS_b) è uguale o inferiore alla varianza in Y entro ciascun gruppo (MS_w), il rapporto F non risulterà significativo.

Le formule per calcolare SS_b e SS_w sono diverse, ma portano, naturalmente, allo stesso risultato: SS_b è data dalla somma degli scarti al quadrato della media del gruppo j dalla media complessiva, moltiplicati per la numerosità n_j *del* gruppo j; SS_w è data dalla somma degli scarti al quadrato dei punteggi dei soggetti i dalla media del loro gruppo j.

$SS_b = \sum n_j(\bar{y}_j - \bar{Y})^2$ → `(SS_b<-(3*(2-4.5)^2)+(5*(6-4.5)^2))` → `[1] 30`

$SS_R = \sum(y_{ij} - \bar{y}_j)^2$ → `marziani$scarti2 <- ifelse(marziani$gruppo== 0, (marziani$burp-2) ^ 2, (marziani$burp - 6)^2)` → `(SS_w<-sum(marziani$scarti2))` → `[1] 12`

L'output di `lm` non mostra SS e MS (tranne, come visto, la radice quadrata della MS_R: il *residual standard error*) e fornisce solo il risultante F. Una diversa funzione, che useremo nell'ANOVA con più predittori between groups ed è una sorta di "riassunto" di `lm`, esegue il modello lineare restituendo nell'output le SS, i loro df, le MS e F con $p - value$ (la "tradizionale" tabella fattoriale): **`aov(Y~X)`**.

```
summary(aov(marziani$burp~marziani$gruppo))
                Df Sum Sq Mean Sq F value  Pr(>F)
marziani$gruppo  1     30      30      15 0.00824
                 ↑      ↑       ↑
                dfM    SSM     MSM
Residuals        6     12       2
                 ↑      ↑       ↑
                DfR    SSR     MSR
```

Potremo comunque vedere il `summary` di un modello creato con `lm` come se fosse stato creato con `aov` usando `summary.aov(modello)`:

```
summary.aov(mars)
                Df Sum Sq Mean Sq F value  Pr(>F)
marziani$gruppo  1     30      30      15 0.00824
Residuals        6     12       2
```

Un dettaglio su R:^{2} nella regressione lineare semplice, coincideva con il coefficiente r di Pearson al quadrato. Quando X è categoriale, il coefficiente R^2 mantiene la stessa interpretazione (proporzione di varianza di Y spiegata da X), ma ora coincide con il **coefficiente di correlazione punto-biseriale** (di Pearson) **al quadrato**.

$$r_{pb} = \frac{\bar{X}_1 - \bar{X}_2}{S_X} \times \sqrt{\frac{n_1 n_2}{N(N-1)}}$$

E infatti:

```
(r_pb<-((2-6)/sd(burp))*sqrt((3*5)/(8*7)))
[1] -0.8451543
r_pb*r_pb
[1] 0.7142857
```

Concludiamo aggiungendo valori predetti e residui al dataframe:

```
marziani$predetti<-predict(mars)
marziani$residui<-round(residuals(mars),1)
marziani
  marziano   gruppo       burp    predetti    residui
1       M1       0          1           2          -1
2       M2       0          2           2           0
3       M3       0          3           2           1
4       M4       1          4           6          -2
5       M5       1          5           6          -1
6       M6       1          6           6           0
7       M7       1          7           6           1
8       M8       1          8           6           2
```

I valori **predetti** corrispondono alla **media del gruppo di appartenenza** dei soggetti.

Vediamo quindi la nostra equazione di regressione, del tutto analoga a quella vista nel capitolo precedente:

… più l'effetto del predittore, cioè la differenza fra i gruppi…

Il valore Y del soggetto i del livello j di X… → $y_{ij} = (b_0 + b_1 G_j) + e_{ij}$ ←*più l'errore del modello riferito al soggetto ij*

… è dato dalla media del gruppo di controllo — *…moltiplicato per il valore del livello X_j cui appartiene il soggetto i*

Ovvero: il punteggio del marziano M7 $y_{M7} = 7$ è dato dalla **media della nausea in popolazione**, stimata dalla media del gruppo di **controllo** X_0 ($\boldsymbol{b_0 = +2}$), più **l'effetto dell'appartenenza al gruppo**, che è quantificato come **differenza tra le medie** dei due gruppi ($\boldsymbol{b_1 = +4}$) **moltiplicato** per il **valore del suo gruppo di appartenenza** ($\boldsymbol{X_1 = \times 1}$) più la quota di errore del modello nello stimare la sua nausea ($\boldsymbol{residuo = +1}$):

```
2+(4*1)+1
[1] 7
```

Oppure: il punteggio del marziano M3 $y_{M3} = 3$ è dato dalla **media della nausea in popolazione**, stimata dalla media del gruppo di **controllo** X_0 ($b_0 = +2$), più **l'effetto dovuto all'appartenenza al gruppo**, quantificato come **differenza tra le medie dei due gruppi** ($b_1 = +4$) **moltiplicato** per il **valore del suo gruppo di appartenenza** ($X_0 = \times 0$), più la quota di errore del modello nello stimare la sua nausea ($\boldsymbol{residuo} = +1$):

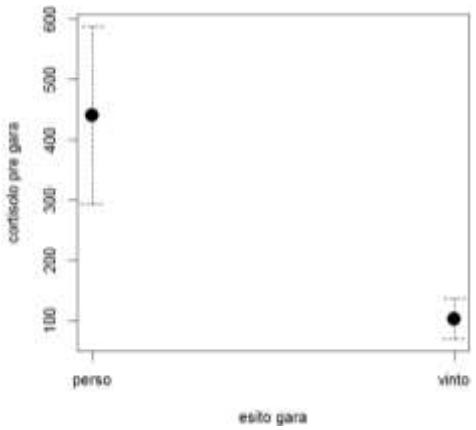
```
2+(4*0)+1
[1] 3
```

Eccetera.

Vediamo un'applicazione di ANOVA a dati veri (dataframe **karate**: lo ritroveremo nella regressione logistica, capitolo 14), relativi a un campione di atleti professionisti partecipanti a una gara internazionale di karate. Oltre all'esito del loro match (**X**: $\boldsymbol{X_1 = perso,\ X_2 = vinto}$), abbiamo un indicatore ormonale di stress, il cortisolo salivare raccolto prima della gara, e un tratto temperamentale, l'Harm Avoidance. Entrambi sono legati alla **serotonina**: un'alta produzione di serotonina è associata a inibizione comportamentale (alto HA) e a maggiore produzione di cortisolo. C'è una **differenza non casuale** nel **cortisolo– pre-gara** (**Y: metrica, continua**) **tra chi ha vinto e chi ha perso? Conoscere l'esito della gara consente di predire, con un errore il più piccolo possibile, il livello di cortisolo pre-gara?**

Descriviamo:

```
Desc(karate$cortisolo_prima~karate$esito_gara)
------------------------------------------------------------
          perso    vinto
mean    441.168  103.276
median  324.500   80.800
sd      357.745   72.228
IQR     475.000  102.500
n            25       21
np       54.348%  45.652%
NAs            0        0
0s             0        0

abs(441.1680-103.2762)
[1] 337.8918
```



Sembra che la differenza tra gli esiti sia consistente.

Vediamo X:

```
class(karate$esito_gara)
[1] "factor"
levels(karate$esito_gara)
[1] "perso" "vinto"
```

Le etichette del fattore sono “perso” e “vinto”; come abbiamo detto nel capitolo 2, i livelli delle variabili `factor` sono **numeri**, cui è possibile assegnare **etichette** nominali per ricordare meglio il livello. Non è necessario rinominare le etichette in “0” e “1” per fare il modello lineare, perché sappiamo che R gestisce i livelli dei `factor` come numeri cui sono assegnate etichette. Quando deve applicare *dummies* nei modelli lineari, **assegna “0” al livello / gruppo di riferimento** e “**1” al livello con cui viene calcolata la differenza**. Se non diversamente specificato, il livello di riferimento **0 è il primo in ordine alfanumerico**; si può verificare chiedendo di visualizzare i **contrasti** (**differenze tra i livelli**) predisposti nel fattore con **`contrasts(factor)`**:

```
contrasts(karate$esito_gara)
      vinto
perso     0
vinto     1
```

Possiamo procedere al modello lineare; sappiamo già che $\boldsymbol{b_0 = 441.1680}$ (media del gruppo 0-perso) e il coefficiente angolare $\boldsymbol{b_1 = 337.8918}$ (differenza tra le medie passando dal gruppo 0-perso al gruppo 1-vinto), con segno negativo (il cortisolo si abbassa):

```
stress<-lm(karate$cortisolo_prima~karate$esito_gara)
summary(stress)

Residuals: cortisolo - cortisolo predetto

    Min       1Q  Median      3Q     Max
-351.07  -193.32  -33.03   66.75  1005.83
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)             441.17      53.73   8.210 2.02e-10
karate$esito_garavinto  -337.89      79.53  -4.249  0.00011

Residual standard error: 268.7 on 44 degrees of freedom
Multiple R-squared:  0.2909,    Adjusted R-squared:  0.2748
F-statistic: 18.05 on 1 and 44 DF,  p-value: 0.0001098
confint(stress)
                           2.5 %    97.5 %
(Intercept)             332.8772  549.4588
karate$esito_garavinto -498.1649 -177.6187
```

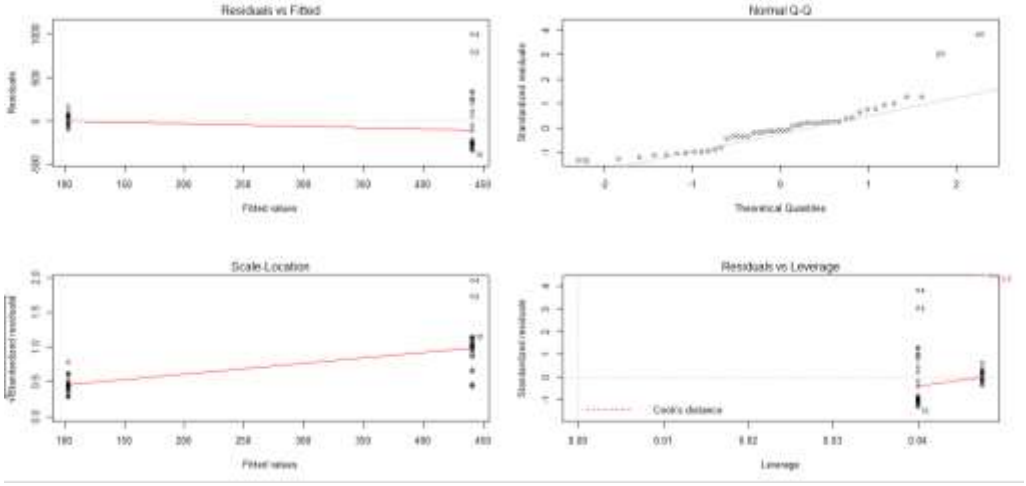
Sembra che lo stress pre-gara sia significativamente più alto tra i perdenti: il cortisolo si abbassa di 337.9 ml passando dal gruppo dei perdenti al gruppo dei vincitori ($b_1 = -337.9$); in popolazione, la variazione nel cortisolo è compresa tra -498.2 e -.177.6. La relazione tra cortisolo ed esito è significativa: lo leggiamo sia dal $p - value$ del coefficiente angolare ($p < .05$) sia dal fatto che il $95\%CI$ di b_1 non contiene il valore previsto da $H_0 = 0$. L’esito della gara spiega il 29.1% della variabilità della concentrazione di cortisolo ($R^2 = .291$); in popolazione, la stima della varianza spiegata cala di poco ($R^2_{adj} = .275$). La varianza del cortisolo spiegata dal predittore Esito è 18 volte maggiore della varianza del cortisolo attribuibile a qualsiasi altra causa ($F = 18.05$). Gli errori del modello hanno una mediana decisamente alta rispetto all’auspicabile 0, e una sottostima massima piuttosto rilevante: la presenza di outliers bivariati è quasi certa (a giudicare dal CI, nel gruppo dei perdenti).

Verificate se nel modello ci sono davvero outlier bivariati, quanti sono e a quale gruppo appartengono; calcolate anche il rispettivo valore predetto dal modello e il loro residuo.

*Verificate se anche il tratto temperamentale **HA** è diverso nei due gruppi, traendone le dovute conclusioni.*

I **pre-requisiti** sui residui del modello lineare con una X categoriale si valutano come visto nel capitolo 9. Sintetizziamo, cominciando a individuare eventuali outliers e casi influenti:

```
summary(rstandard(stress))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.3340 -0.7344 -0.1260  0.0000  0.2546  3.8210
summary(cooks.distance(stress))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0001429 0.0009243 0.0035790 0.0209000 0.0211900 0.3042000
shapiro.test(residuals(stress))
       Shapiro-Wilk normality test
data:  residuals(stress)
W = 0.8584, p-value = 5.121e-05

par(mfrow = c(2,2))
plot(stress)
```



In ANOVA, la verifica della omoschedasticità degli errori coincide con la **verifica dell'uguaglianza delle varianze nei gruppi** ($H_0 = s^2_{x1} = s^2_{x2}$): come abbiamo già rilevato, infatti, la SS_R **in ANOVA per gruppi indipendenti coincide con la varianza entro i gruppi**. Uno dei test tradizionalmente più usati è il **test di Bartlett (1937)**, che usa la distribuzione di probabilità χ^2 e in R è una funzione di base: `bartlett.test(y,x)` o `bartlett.test(y~x)`.

```
bartlett.test(karate$cortisolo_prima, karate$esito_gara)
       Bartlett test of homogeneity of variances
data:  karate$cortisolo_prima and karate$esito_gara
Bartlett's K-squared = 37.928, df = 1, p-value = 7.341e-10
```

Tuttavia, il test di Bartlett è estremamente **sensibile**, in senso negativo, alla **violazione della normalità**: in questa condizione porta con facilità a **un errore di I tipo**, perché il rifiuto dell'H_0 relativa all'omoschedasticità è dovuto, in realtà, solo alla non normalità della distribuzione.

Se la distribuzione degli errori non è normale, meglio usare il **test di Levene** (1960), applicabile a due o più gruppi. Potete trovarlo nel package **car**: `leveneTest(y=, x=, center=)`. La funzione consente di **modificare il centro degli scarti** alla base delle varianze, scegliendo scarti dalla media (`center= "mean"`), che va bene solo se gli errori sono normalmente distribuiti, oppure scarti dalla mediana (di default; `center= "median"`)

```
leveneTest(karate$cortisolo_prima,karate$esito_gara,center="mean")
Levene's Test for Homogeneity of Variance (center = "mean")
      Df F value    Pr(>F)
group  1  23.564 1.558e-05
      44
leveneTest(karate$cortisolo_prima,karate$esito_gara,center="median")
Levene's Test for Homogeneity of Variance (center = "median")
      Df F value    Pr(>F)
group  1   15.06 0.0003449
      44
```

DescTools ha una funzione del tutto identica, con l'eccezione del nome: **LeveneTest(y,x, center= "mean", "median")**:

```
LeveneTest(karate$cortisolo_prima,karate$esito_gara,kruskal.test= TRUE)
Levene's Test for Homogeneity of Variance (center = median: TRUE)
      Df F value    Pr(>F)
group  1   15.06 0.0003449
      44
```

Il test si trova anche nel package **lawstat**: **levene.test(y,x)**: oltre a media e mediana (location= "mean", location="median"), questa funzione consente di centrare anche sulla **media troncata**, da preferire (come la mediana) in presenza di outliers (location="trim.mean" e trim.alpha= proporzione di casi da troncare a ogni coda), e di stimare la differenza usando il test di Kruskal-Wallis, test non parametrico che rivedremo nel capitolo 12 e abbiamo già notato nell'output di Desc(Y~X), utile in caso di violazione della normalità (kruskal.test=TRUE):

```
levene.test(karate$cortisolo_prima,karate$esito_gara,location="trim.mean", trim.alpha=.025)
      Modified robust Levene-type test based on the absolute deviations from the trimmed mean
      (none not applied because the location is not set to the median )
Data: karate$stress
Test Statistic = 23.564, p.value = 1.558e-05
levene.test(karate$cortisolo_prima,karate$esito_gara,kruskal.test= TRUE)
      rank based (Kruskal- Wallis) modified robust Brown.Forsythe Levene-type test based on
      absolute deviations from the median
Data: karate$stress
Test Statistic = 23.1134, p.value = 1.527e-06
```

Vedremo cosa fare in caso di violazione dei pre-requisiti nel §4.4

10.1.2 Il t-test di Student per campioni indipendenti

Anche il t-*test* per campioni indipendenti non è altro che l'adattamento a una X dicotomica del solito modello lineare $y_{ij} = (b_0 + b_1 G_j) + e_{ij}$.

Come in ANOVA, H_0 è che i due campioni, che rappresentano i due livelli X_1 e X_2 di un fattore X, provengano dalla **medesima popolazione**, per cui le **medie** dei due **gruppi** sono **equivalenti ovvero solo casualmente differenti**→ X non esercita un effetto significativo su Y, dato non rende differenti i due gruppi. Il t-test consente di verificare anche H_1 **monodirezionali**:

- H_0: $\mu_1 = \mu_2$;
- H_1: $\mu_1 \neq \mu_2$ oppure $\mu_1 > \mu_2$ oppure $\mu_1 < \mu_2$

A differenza dell'ANOVA, che usa la distribuzione F per attribuire un $p - value$ al rapporto F tra MS_M e MS_R, nel t-test si valuta il rapporto tra la differenza delle medie di due gruppi indipendenti e la variabilità dei dati. Il test consente di attribuire un **valore di probabilità** alla differenza osservata tra i gruppi: **quanto è probabile**, se i gruppi appartenessero alla stessa popolazione, **riscontrare una differenza tra le loro medie uguale o maggiore di quella osservata**?

L'effetto di X è quantificato dalla **differenza tra la media del gruppo X_0 e la media del gruppo X_1, ponderata per la variabilità in popolazione della differenza tra le medie**:

$$t_{N_1-1+N_2-1} = \frac{\bar{x}_1 - \bar{x}_2}{SE_{(\bar{x}_1-\bar{x}_2)}}$$

Possiamo ricavare dai dati le medie campionarie, ma non lo SE della loro differenza in popolazione.

Possiamo fare a meno, però, dello SE della differenza tra le medie: è dimostrabile che se le distribuzioni X_1 e X_2 sono **indipendenti** (e questo **assunto** è garantito dalla randomizzazione, o dalla cautela del ricercatore), la **varianza di una differenza fra due distribuzioni casuali** (cioè il **quadrato del nostro denominatore**) è data dalla **somma delle corrispondenti varianze**.

Sappiamo anche che se una distribuzione di valori casuali è **normalmente** distribuita (e questo è il secondo assunto che vediamo), con media μ e varianza σ^2, allora la distribuzione campionaria di campioni di dimensione n si distribuisce normalmente, con $\bar{x}_{DCM} = \boldsymbol{\mu}$ e $\sigma^2 = \boldsymbol{\sigma^2/N}$.

Poiché, però, non sono mai o quasi mai note le varianze in popolazione, dobbiamo **stimare σ_1^2 e σ_2^2** usando i valori campionari **S_1^2 e s_2^2**, ottenendo perciò:

$$\frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

Ora avremmo tutti i dati disponibili per calcolare la statistica del test, ma il risultato di questo rapporto **non si distribuisce secondo una distribuzione di probabilità nota**, quindi non potremmo assegnargli un *p*-value.

Dobbiamo ricorrere al **terzo assunto**, l'**omoschedasticità**: **se** le due varianze sono uguali ($\sigma_1^2 = \sigma_2^2$), possiamo usare al loro posto la loro **varianza congiunta *(pooled)*** σ_p^2: se $N_1 = N_2$ (disegno bilanciato), essa equivale alla **media aritmetica** di σ_1^2 **e** σ_2^2:

$$\frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{s_p^2}{N_1} + \frac{s_p^2}{N_2}}}$$

Se $N_1 \neq N_2$, la varianza congiunta sarà stimata come **media ponderata di σ_1^2 e σ_{22}^2**: il **criterio di ponderazione** sono i df **$(N-1)$** dei gruppi: pesa di più il gruppo più numeroso, da ritenersi più probabilmente rappresentativo della popolazione:

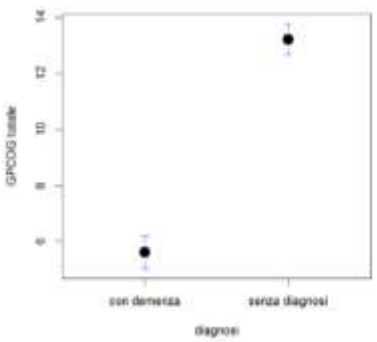
$$s_p^2 = \frac{\sum(x_{i1} - \bar{x}_1)^2 + \sum(x_{i2} - \bar{x}_2)^2}{(N_1 - 1) + (N_2)}$$

Finalmente, il rapporto tra la differenza fra le medie e la radice quadrata della varianza congiunta, indipendentemente dal fatto che sia stimata come media aritmetica o ponderata, di **distribuisce come un quantile t**, per $\boldsymbol{df = N1 - 1 + N2 - 1 = N - 2}$ **.**

$$t_{N-2} = \frac{\bar{x}_1 - \bar{x}_2}{S_{pooled}}$$

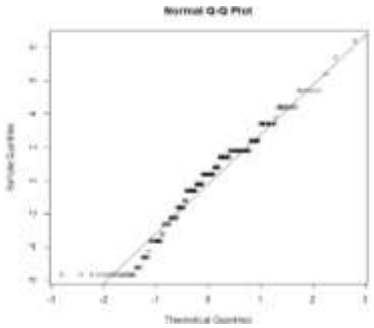
Vediamo all'opera il test con un esempio su dati veri: quelli contenuti nel dataframe **`vecchietti`** sono relativi alla validazione del test GPCOG, uno strumento di valutazione del funzionamento cognitivo nella terza età. È pensato per i medici di Medicina Generale, affinché lo usino come test di screening per individuare gli assistiti che manifestano i primi sintomi di demenza, da indirizzare successivamente alla valutazione geriatrica specialistica. Se il punteggio della scala totale è <5, si ha declino cognitivo; se è tra 5 e 8, si ha Mild Cognitive Impairment (MCI); se >8, il funzionamento cognitivo è normale. Valutiamo se il test è in grado di discriminare anziani con diagnosi di demenza già ricevuta (campione clinico), da soggetti di pari età, senza diagnosi di demenza o di MCI (campione non clinico). Accorciamo `vecchietti` in `v` e descriviamo:

```
Desc(v$GPCOG_totale~v$diagnosi)
------------------------------------------------------------
Summary:
n pairs: 200, valid: 200 (100.0%), missings: 0 (0.0%), groups: 2
          con demenza  senza diagnosi
mean            5.610         13.203
median          6.000         14.000
sd              3.443          2.176
IQR             5.000          3.250
n                 136             64
np             68.000%        32.000%
NAs                  0              0
0s                  16              0
```



Il disegno è tutt'altro che bilanciato, la numerosità dei gruppi è molto diversa.
Per applicare il t-test dobbiamo verificare i due classici **assunti** dei modelli lineari: **normalità** e **l'omoschedasticità**: procediamo.

```
totale<-lm(v$GPCOG_totale~v$diagnosi)
shapiro.test(totale$residuals)
        Shapiro-Wilk normality test
data:  totale$residuals
W = 0.96863, p-value = 0.0001918
```



*Si direbbe **improbabile** che la differenza tra la distribuzione degli errori e la normale sia solo casuale. Tuttavia, abbiamo 200 soggetti, il test di Shapiro è piuttosto potente, e la statistica W tende a 1. Quale conclusione potremmo trarne?*

```
levene.test(v$GPCOG_totale,v$diagnosi)
Modified robust Brown.Forsythe Levene-type test based on absolute deviations from the median
Data: v$GPCOG_totale
Test Statistic = 15.343, p.value = 0.0001233
LeveneTest(v$GPCOG_totale,v$diagnosi)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value    Pr(>F)
group   1  15.343 0.0001233
      198
```

Prendiamo atto che le **varianze** degli errori sono **diverse**: è **un'informazione che dobbiamo dare a R** nello scrivere **`t.test(Y~X, var.equal=TRUE/FALSE)`**. Se le varianze sono uguali (**`=TRUE`**), viene prodotto il classico test t di Student per campioni indipendenti. Se le varianze sono diverse (**`=FALSE`**, di default), è prodotto il **test di Welch**[88] (***Welch's approssimate t***; 1937), in cui è modificata la stima dello SE della differenza tra le medie. Il test di Welch è applicabile anche nel caso di gruppi $k > 2$, ma quando è applicato a due campioni presenta il grave inconveniente che alcuni suoi valori critici sono minori di quelli della distribuzione t per equivalenti df, il che aumenta la probabilità di un errore di I tipo. **Satterthwaite (1946)** risolve questo problema proponendo una **stima dei df**, adottata da R, tale per cui risultano **sempre minori di quelli non corretti** ($N - 2$), in modo tanto **più marcato** quanto **maggiore è la differenza tra le due varianze:** il test ne risulta più prudente nel rifiutare H_0. **La stima di Satterthwaite**[89] non fornisce valori interi: come df è scelto il valore arrotondato **in difetto**.

Tra gli argomenti opzionali, possiamo specificare la direzione di H_1 (`alternative= "two.sided", "greater", "less"`), il livello di verosimiglianza del CI (`conf.level=`) e se il test si applica a campioni indipendenti o appaiati (`paired= TRUE / FALSE`): di default, il t-test è considerato per campioni indipendenti. Vedremo il t-test per dati appaiati nel §4.2.2. Se ci sono NA, possono essere gestiti con `na.action=na.exclude`.

```
t.test(v$GPCOG_totale~v$diagnosi, var.equal = FALSE)
        Welch Two Sample t-test
data:  v$GPCOG_totale by v$diagnosi
t = -18.913, df = 181.38, p-value < 2.2e-16   ← Notate i df approssimati. Non confermiamo H0
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:                ← CI della differenza tra le medie
 -8.384945 -6.800717                           ← ristretto e non contiene H0: μ1=μ2= 0. La differenza attesa tra le popolazioni
                                                  sta fra |6.8| e |8.3| punti; il segno indica se μ1>μ2 o μ1<μ2.
sample estimates:
   mean in group con demenza mean in group senza diagnosi
                    5.610294                     13.203125
```

[88] La formula del test di Welch non è piacevole alla vista: $w = \frac{\sum w_i\left(\frac{(\bar{X}_i-\bar{X})^2}{g-1}\right)}{1+\frac{2(g-2)}{g^2-1}\sum\left(\left(\frac{1-w_i}{u}\right)^2\times(n_i-1)\right)}$; g: numero dei gruppi; w: n_i/s^2_i;; u: Σw_i; $\bar{X} = \frac{\sum w_i\bar{X}_i}{u}$

[89] Anche quella della stima di Satterthwaite non è agevole: $d' = \frac{\left(\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}\right)}{\frac{\left(\frac{S_1^2}{n_1}\right)^2}{n_1-1}+\frac{\left(\frac{S_2^2}{n_2}\right)^2}{n_2-1}}$

Se avessimo ignorato l'eteroschedasticità e avessimo fatto un t-test senza correzioni, avremmo ottenuto:

```
t.test(v$GPCOG_totale~v$diagnosi, var.equal = TRUE)
        Two Sample t-test
data:  v$GPCOG_totale by v$diagnosi
t = -16.175, df = 198, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8.518514 -6.667148
sample estimates:
   mean in group con demenza mean in group senza diagnosi
                    5.610294                    13.203125
```

Non possiamo terminare l'analisi senza un coefficiente di effect size, che per questa analisi è il coefficiente d di Cohen, già affrontato; rimandiamo al §6.6 per rinfrescare formula e interpretabilità, e qui ci dedichiamo al suo calcolo. In R ci sono varie funzioni per calcolare gli indici di ES; usiamo, **per esempio**, `cohen.d(y~x)` di `effsize`, che abbiamo già trovato nel CI del t-test per campione unico (questa volta non indicheremo `mu=`):

```
cohen.d(v$GPCOG_totale~v$diagnosi)
Cohen's d
d estimate: -2.451927 (large)
95 percent confidence interval:
      inf       sup
-2.839091 -2.064763
```

La differenza non è semplicemente significativa: **l'effetto del gruppo di appartenenza** sul punteggio GPCOG totale è **forte**, sia nel **campione** (-2.45), sia nella sua **stima in popolazione** al 95% di verosimiglianza (-2.84 — -2.06): tra le medie dei gruppi ci sono oltre due deviazioni standard di differenza. Purtroppo**,** nello script compare anche la **convenzionale** etichetta che definisce **convenzionalmente** l'entità della differenza: non va usata **acriticamente!**

Un coefficiente analogo a d, ma **corretto per piccoli campioni** ($N < 20$) è il g **di Hedges**, che abbiamo citato nel paragrafo dedicato alla meta-analisi; per campioni numerosi come il nostro, d e g sono sostanzialmente identici. In effetti, le formule di d e g sono uguali, a parte il fatto che g calcola la *pooled sd* usando $N - 1$, anziché N come fa d (*stima non biased*). Potete trovarlo in `effectsize`, dove trovate anche `cohens_d(formula)`; `hedges_g(formula)` è molto semplice: di default, i campioni sono ritenuti indipendenti (`paired= FALSE`) e la verosimiglianza del CI è = .95 (`ci= .95`). L'interpretazione di g è analoga a quella di d.

```
hedges_g(v$GPCOG_totale~v$diagnosi)
Hedges' g |         95% CI
--------------------------
-2.44     | [-2.82, -2.06]
- Estimated using pooled SD.

cohens_d(v$GPCOG_totale~v$diagnosi)
Cohen's d |         95% CI
--------------------------
-2.45     | [-2.83, -2.07]
- Estimated using pooled SD.
```

Infine, **se riscontriamo eteroschedasticità** può essere opportuno utilizzare il coefficiente **Delta Δ di Glass**, che utilizza al denominatore solo la sd del secondo gruppo, invece di calcolare la sd *pooled*, e si interpreta come d e g. Troviamo anch'esso in `effectsize`: `glass_delta(formula)`. Nel nostro esempio, in cui le sd sono significativamente differenti, il coefficiente Δ enfatizza ulteriormente il già rilevante effetto:

```
glass_delta(v$GPCOG_totale~v$diagnosi)
Glass' delta |         95% CI
----------------------------
-3.49        | [-4.19, -2.78]
```

Se avessimo fatto **ANOVA**, invece del t-test, sulle stesse variabili:

```
summary(lm(v$GPCOG_totale~v$ Residuals:
    Min      1Q  Median      3Q     Max
-5.6103 -2.2031  0.3897  1.7969  8.3897
Coefficients :
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)               5.6103     0.2655   21.13   <2e-16
v$diagnosisenza diagnosi  7.5928     0.4694   16.18   <2e-16
---
Residual standard error: 3.097 on 198 degrees of freedom
Multiple R-squared:  0.5692,     Adjusted R-squared:  0.5671
F-statistic: 261.6 on 1 and 198 DF,  p-value: < 2.2e-16
```

… avremmo saputo **anche** che la variabilità del punteggio attribuita al gruppo di appartenenza è pari al 56.7% (nel campione quanto in popolazione), e che il gruppo con diagnosi (X_0) ha una media (5.6) di 7.6 punti inferiore al gruppo senza diagnosi (X_1). Notate che il valore del t-test associato al coefficiente angolare è uguale al valore del t-test per campioni indipendenti che assume varianze uguali, nonché uguale alla radice quadrata del rapporto F:

```
sqrt(261.6)
[1] 16.17405
```

Quando i livelli di X sono due, infatti, ANOVA tra gruppi e t-test per campioni indipendenti sono del tutto equivalenti: si preferisce, però, usare t-test invece di ANOVA quando N è piccolo (indicativamente $N < 20$ soggetti per gruppo).

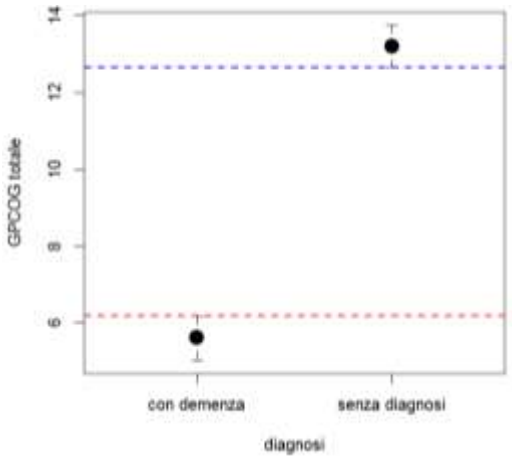
Facciamo un passo indietro al plot dei CI: quali informazioni sulla significatività della differenza possiamo ricavare dalla **sovrapposizione** dei CI delle medie di **due gruppi indipendenti**?

Se i CI (per $\alpha \leq .95$) di due campioni indipendenti **non si sovrappongono**, i due campioni sono **estratti da due diverse popolazioni**, con $\mu_1 \neq \mu_2$, e il $p - value$ della corrispondente statistica t sarà $p \leq .01$ (Cumming, 2005).

Questa relazione tra CI e $p - value$ è tanto più predicibile quanto più i due gruppi hanno una numerosità $N > 10$ e i due **margini di errore sono simili**, cioè se w_{2_1} **non** è **>2 volte più grande o più piccolo** di w_{2_2} .

In questo esempio, il margine di errore medio è:

```
tapply(v$GPCOG_totale, v$diagnosi, MeanCI)
$`con demenza`
    mean   lwr.ci   upr.ci
5.610294 5.026406 6.194182
$`senza diagnosi`
    mean   lwr.ci   upr.ci
13.20312 12.65953 13.74672
w_con<-abs(5.026406-6.194182); w_senza<-abs(12.65953-13.7
4672)
w_con;w_senza
[1] 1.167776
[1] 1.08719
(w_medio<-(1.16776 + 1.08719)/2)
[1] 1.127475
```



w_{2_1} e w_{2_2} sono molto simili; il margine di errore medio è piccolo, circa 1.1 punti; la mancata sovrapposizione è quantificata dalla distanza, in valore assoluto, tra il limite superiore del gruppo con demenza (6.19) e il limite inferiore del gruppo senza demenza (12.66):

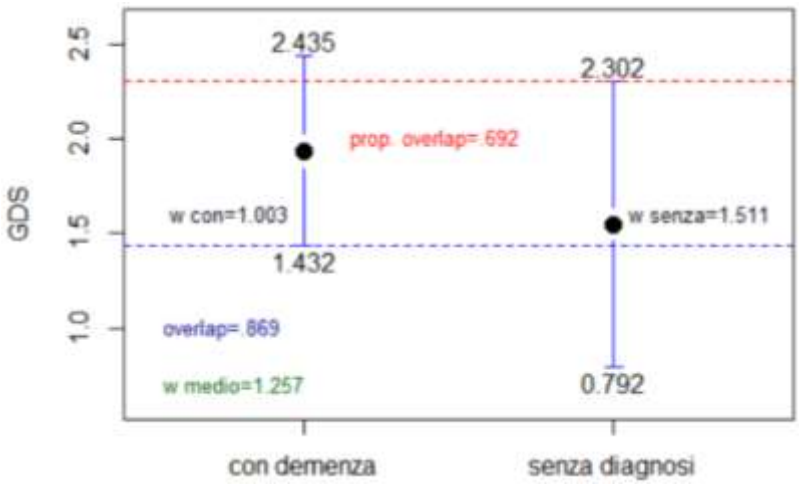
```
(no_overlap<-abs(6.194182-12.65953))
[1] 6.465348
```

La mancata sovrapposizione è oltre 6 volte più grande del margine di errore medio: siamo ragionevolmente sicuri che il $p - value$ associato alla differenza sarà decisamente inferiore a p .01.

Se i CI (per $\alpha \geq .95$) di due campioni indipendenti **si sovrappongono**, la differenza tra le medie **può essere significativa o non significativa** per $\alpha \leq .05$(Cumming, 2005). L'interpretazione del grafico dei CI, quindi, deve essere più cauta.

Vediamo la differenza tra pazienti e non pazienti nella **scala di depressione GDS**:

```
tapply(v$GDS, v$diagnosi, MeanCI)
$`con demenza`
    mean   lwr.ci   upr.ci
1.933824 1.432362 2.435285
$`senza diagnosi`
     mean    lwr.ci    upr.ci
1.5468750 0.7916655 2.3020845
```



Le medie sono piuttosto vicine; vediamo il margine di errore medio (arrotondiamo):

```
(w_con<-2.435285-1.432362)
[1] 1.002923
(w_senza<-2.3020845-.7916655)
[1] 1.510419
(w_medio<-(w_con + w_senza)/2)
[1] 1.256671
```

$w_{2_{senza}}$ e $w_{2_{con}}$ non sono proprio uguali, ma il rapporto tra loro è inferiore a 2: l'inferenza sul $p - value$ dovrebbe essere affidabile. Visivamente, la sovrapposizione tra i CI è ampia, apparentemente maggiore del margine di errore medio:

```
(overlap<-abs(1.432362-2.3020845))
[1] 0.8697225
```

Infatti. Calcoliamo la **sovrapposizione proporzionale** (***proportional overlap***), ovvero il **rapporto tra sovrapposizione** dei CI (*overlap*) e il **margine di errore medio** dei CI ($w_{2_{medio}}$):

```
(prop_overlap=overlap/w_medio)
[1] .6920842
```

Quando la **sovrapposizione proporzionale è maggiore di .50, il $p - value$ associato alla statistica t sarà maggiore di $p = .05$** → i due campioni appartengono alla stessa popolazione. Possiamo verificare:

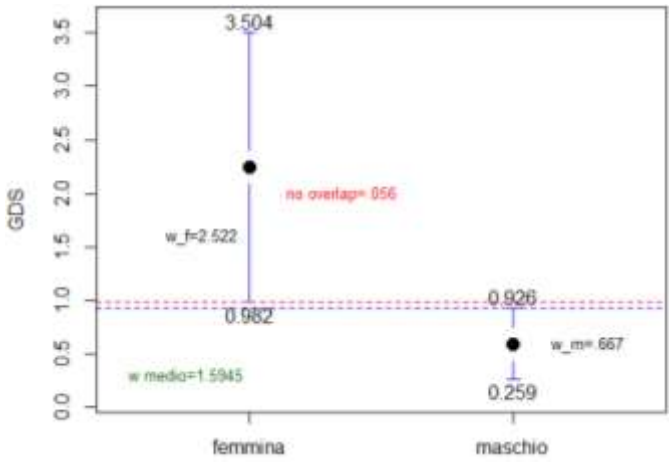
```
t.test(v$GDS~v$diagnosi, var.equal= TRUE)
        Two Sample t-test
data:  v$GDS by vecchietti$diagnosi
t = 0.85711, df = 198, p-value = 0.3924
```

Vediamo esempi meno estremi.

Cominciamo col chiederci se il genere di appartenenza ha un effetto sulla depressione (**GDS~sesso**), **prima** nel campione dei soggetti **non clinici**, **poi** nel campione dei soggetti **clinici**.

```
senza<-subset(v,v$diagnosi=="senza diagnosi")
con<-subset(v,v$diagnosi=="con demenza")
```

```
tapply(senza$GDS, senza$sesso, MeanCI)
$femmina
     mean    lwr.ci    upr.ci
2.2432432 0.9824038 3.5040827
$maschio
     mean    lwr.ci    upr.ci
0.5925926 0.2587373 0.9264479

(w_f<-abs(.9824038-3.5040827)
[1]2.521679
(w_m<-abs(.2587373-.9264479)
[1] 0.6677106
(w_medio<-(w_f+w_m)/2)
[1] 1.5945
.9824038-.9264479
[1] 0.0559559
```



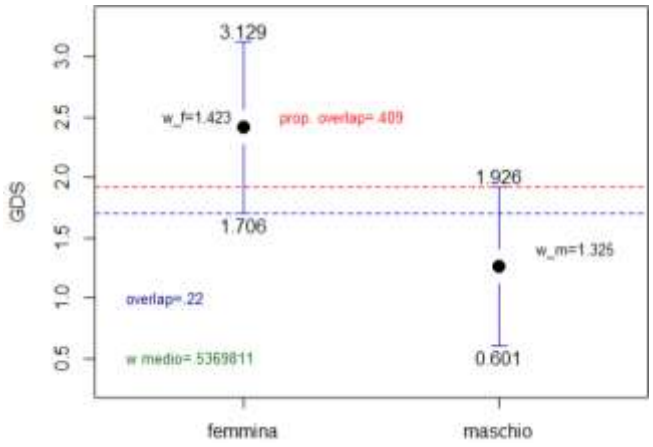
I CI non si sovrappongono, quindi la statistica t dovrebbe avere un $p < .01$; però, diversamente dal primo esempio, la distanza tra i due CI non è ampia, ed è **decisamente inferiore al margine di errore medio**: quindi, il $p - value$ non sarà $p < .01$, ma **compreso tra $p < .05$ e $p > .01$**. Inoltre, i due margini di errore sono profondamente diversi (w_{2_M} è quasi quattro volte inferiore a w_{2_F}), il che rende l'inferenza basata sull'ispezione visiva un po' problematica. Vediamo:

```
t.test(senza$GDS~senza$sesso,correlazione.equal = TRUE)
       Two Sample t-test
data:  senza$GDS by senza$sesso
t = 2.2235, df = 62, p-value = 0.02983
```

Confermiamo l'impressione visiva: i campioni appartengono probabilmente a popolazioni differenti.

Ora verifichiamo l'effetto del genere sulla depressione nel campione dei **pazienti con demenza**:

```
tapply(con$GDS, con$sesso, MeanCI)
 $femmina
    mean   lwr.ci   upr.ci
2.417722 1.706141 3.129302
$maschio
    Mean   lwr.ci   upr.ci
1.263158 0.600717 1.925599
```



I CI sono parzialmente sovrapposti, la significatività della differenza è in dubbio. Quantifichiamo la sovrapposizione proporzionale rispetto al margine di errore medio; questa volta, w_2 sembra molto simile nei due gruppi.

```
(w_f<-abs(1.706141-3.129302);(w_m<-abs(.600717-
1.925599)
[1] 1.423161
[1] 1.324882
(w_medio<-(w_f+w_m)/2)
[1] 0.53698111
(overlap<-abs(1.926-1.706))
[1] 0.22
(prop_overlap=overlap/w_medio)
[1] 0.409
```

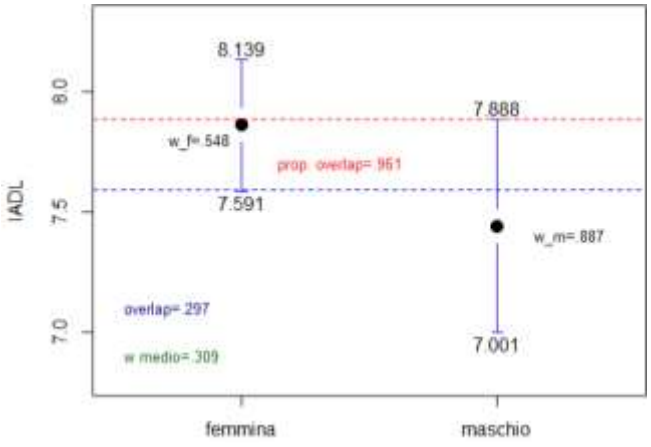
Quando la **sovrapposizione proporzionale è inferiore a .50**, se $N > 10$ e w_{2_1} **non** è **> 2 volte più grande / piccolo** di w_{2_2}, **il $p - value$ associato alla statistica t sarà $p < .05$** → i due campioni appartengono a diverse popolazioni.

Infatti:

```
t.test(con$GDS~con$sesso,variabile.equal = FALSE)
       Welch Two Sample t-test
t = 2.3711, df = 132.98, p-value = 0.01917
…
95 percent confidence interval:
 0.1914238 2.1177035
```

Infine, una sovrapposizione più evidente: la differenza **tra uomini e donne** del **campione normativo** nella misura di autonomia nelle attività di vita quotidiana (**IADL**):

```
tapply(senza$IADL, senza$sesso, MeanCI)
$femmina
    mean   lwr.ci   upr.ci
7.864865 7.590798 8.138932
$maschio
    mean   lwr.ci   upr.ci
7.444444 7.001032 7.887857
```



I CI sono più sovrapposti che nell'esempio precedente, la significatività della differenza è fortemente in dubbio; w_{2_M} è 1.6 volte maggiore rispetto a w_{2_F}. Quantifichiamo la sovrapposizione proporzionale rispetto a $w_{2_{medio}}$:

```
(w_f<-abs(7.590798-8.138932)); (w_m<-
abs(7.001032-7.887857))
[1] 0.548134
[1] 0.886825
(w_medio<-(w_f+w_m)/2)
[1] 0.3090429
(overlap<-abs(7.590798-7.887857))
[1] 0.297059
(prop_overlap=overlap/w_medio)
[1] 0.9612224
```

La sovrapposizione proporzionale è superiore a .50, quindi la differenza non sarà significativa. Calcoliamo il $p-value$ esatto:

```
t.test(senza$IADL~senza$sesso, correlazione.equal = TRUE)
       Two Sample t-test
data:  senza$IADL by senza$sesso
t = 1.7325, df = 62, p-value = 0.08816
```

Confermiamo.

Attenzione: nel caso di due **campioni dipendenti** / a misure ripetute, questo **tipo di inferenza visiva non regge**, perché non considera la **dipendenza**, cioè la **correlazione**, tra le misure, che entra nel calcolo della significatività della differenza. Lo vedremo nel §10.2.2.

Concludendo, possiamo descrivere l'analisi eseguita sul punteggio GPCOG per gruppo più o meno così:

*Il **plot** delle medie ha mostrato due distribuzioni con CI chiaramente non sovrapponibili, tanto da rendere già scontata l'appartenenza dei due campioni a **diverse popolazioni per il parametro punteggio GPCOG totale**. Sono i **pazienti con diagnosi**, indipendentemente da qualsiasi **altro potenziale fattore interveniente**, a mostrare un **maggior deficit cognitivo**. La distribuzione dei pazienti è più **variabile al suo interno**, come dimostra la sua maggiore **deviazione standard**: il test di Levene conferma che le varianze entro i gruppi sono significativamente diverse La **probabilità** che i circa **7.5** punti in meno dei pazienti con demenza **riflettano semplicemente una differenza casuale**, e non una differenza attribuibile al loro gruppo di appartenenza, è molto **bassa** ($\boldsymbol{p < .01}$). Il **$\boldsymbol{CI}$** indica che la differenza tra le popolazioni è quantificabile in un divario compreso tra poco più di 8 e poco meno di 7 punti, con una verosimiglianza del 95%. **L'intensità dell'effetto** esercitato della variabile X_{Gruppo} è definibile come **forte** nel **campione** ($d = |2.45|$), e verosimilmente anche in popolazione (tra $|2.83|$ e $|2.06|$). L'ampia numerosità potrebbe facilmente rendere **statisticamente significativa** una differenza tra gruppi anche piccola: il coefficiente d ci conforta, invece, nel definire come anche **interpretativamente rilevante** il dato.*

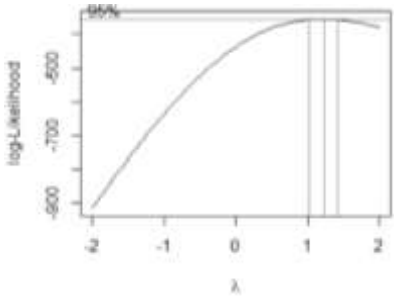
10.1.3 Test non parametrici o robusti

Il modello lineare, nella forma di ANOVA o t-test, richiede che indipendenza degli errori, normalità e omoschedasticità vengano rispettati, per dare risultati attendibili. Come visto, se la normalità dei residui è rispettata, per dati eteroschedastici è **sostituire** *t*-test o ANOVA con il test di Welch, completo di correzione di Satterthewite, oppure, se il problema è la normalità, è possibile provare a **trasformare non linearmente la distribuzione** Y (capitolo 4).

Nei test F e t, la normalizzazione rende **minima** MS_R, e quindi rende i test più potenti, con maggiore probabilità di risultati significativi. Un metodo per individuare quale sia la migliore trasformazione non lineare di Y da usare in un modello di regressione è quello di **Box-Cox**, che abbiamo già visto nel Capitolo 4 per distribuzioni univariate: con `boxcox(modello)` del package `MASS` determiniamo il valore lambda (λ) che indica l'esponente cui elevare la variabile per ottenere la migliore normalizzazione possibile, arrotondato all'intero più prossimo.

Per esempio, abbiamo giudicato la distribuzione degli errori del modello `totale<-lm(v$GPCOG_totale~v$diagnosi)` sufficientemente simile alla normale teorica, secondo il QQplot e la statistica W del test di Shapiro; quindi, il metodo di Box-Cox dovrebbe suggerirci di applicare un esponenziale $= 1$ alla distribuzione da trasformare, ovvero di lasciarla non trasformata. Mettiamo l'ipotesi alla prova; dobbiamo anche operare una trasformazione lineare aggiungendo una **costante positiva** ai dati, perché nel range di `$GPCOG_totale` è presente il valore 0:

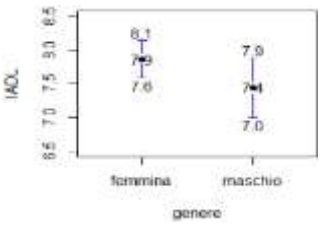
```
totale<-lm(v$GPCOG_totale+.5~v$diagnosi)
tot<-boxcox(totale)
tot$x[which.max(tot$y)]
[1] 1.151515
```



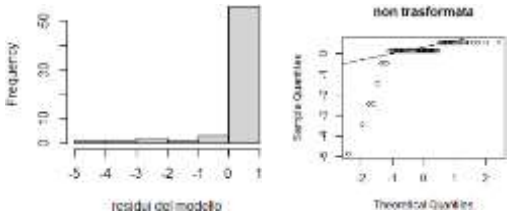
Effettivamente, il valore lambda che ottimizza la normalizzazione di Y è molto prossimo a 1, e i limiti del suo $95\%CI$ sono lontanissimi da 0 e 2. La decisione migliore, in questo caso, è non trasformare la variabile.

Prendiamo ora i soli soggetti senza diagnosi e verifichiamo la significatività delle differenze tra i generi nelle capacità di funzionamento quotidiano (IADL), che in letteratura sono riportate spesso debolmente maggiori tra le donne:

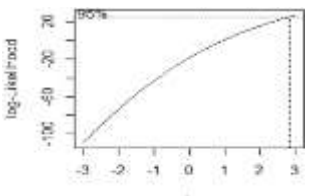
```
senza<-subset(vecchietti, vecchietti$diagnosi=="senza diagnosi")
summary.aov(lm(senza$IADL~senza$sesso))
             Df Sum Sq Mean Sq F value Pr(>F)
senza$sesso   1   2.76  2.7590   3.002 0.0882
Residuals    62  56.99  0.9192
```



La differenza è piccola e non significativa, ma l'analisi è affidabile? La distribuzione degli errori, in effetti, ha una terribile asimmetria positiva e una curtosi altrettanto forte. Potremmo pensare a una trasformazione esponenziale cubica; confermiamola con il metodo di Box-Cox.




```
iadl<-boxcox(lm(senza$IADL~senza$sesso))
iadl<-boxcox(lm(senza$IADL~senza$sesso), lambda = seq(-3,
      3, 1/10))
iadl$x[which.max(iadl$y)]
[1] 3
```



Infatti, il lambda suggerito è $\lambda = 3$. Proviamo a eseguire il modello con la Y `$IADL` elevata alla terza, e vediamo l'effetto sulla potenza dell'analisi:

```
summary.aov(lm(senza$IADL^3~senza$sesso))
             Df Sum Sq Mean Sq F value Pr(>F)
senza$sesso   1  59793   59793   4.939 0.0299
Residuals    62 750534   12105
```

Ora il rapporto F è maggiore (meno MS_R al denominatore) e la significatività è raggiunta. Avremmo lo stesso effetto di incremento della potenza inserendo Y^3 nel t-test, ovviamente, nonché – un po' più ridotto – nel test di Welch:

```
t.test(senza$IADL~senza$sesso)
        Welch Two Sample t-test
t = 1.6516, df = 45.365, p-value = 0.1055
```

```
t.test(senza$IADL^3~senza$sesso)
        Welch Two Sample t-test
t = 2.0462, df = 37.948, p-value = 0.04772
```

Diversi autori[90], tuttavia, preferiscono usare un **approccio non parametrico** invece della trasformazione non lineare, soprattutto se alla deviazione dalla normalità rilevante si aggiunge l'eteroschedasticità, come accade sempre quando la numerosità dei dati è scarsa. D'altronde, i test non parametrici **perdono poco in potenza** ($< 5\%$) rispetto ai parametrici, anche quando la distribuzione è esattamente normale, e sono tanto più potenti di quelli parametrici quanto maggiori sono la non normalità e/o l'eteroschedasticità. A dimostrazione, anticipiamo proprio il test non parametrico di Wilcoxon - Mann-Whitney di cui stiamo per parlare, applicandolo alla relazione `senza$IADL~senza$sesso`, che abbiamo visto non essere significativa con la distribuzione grezza. Con questo test, invece:

```
wilcox.test(senza$IADL~senza$sesso)
        Wilcoxon rank sum test with continuity correction
data:  senza$IADL by senza$sesso
W = 612, p-value = 0.008041
```

L'effetto del genere sulla distribuzione **grezza** delle IADL è indubbiamente significativo: essendo fortemente violato il requisito di normalità, il **test non parametrico è più potente** del suo corrispettivo parametrico.

Tra i diversi test non parametrici per il confronto di due gruppi indipendenti, vedremo il **test di Wilcoxon per gruppi indipendenti** nella sua forma generalizzata (**test di U di Mann-Whitney**) e i primi esempi di **statistica robusta di Wilcox** (attenti a non confondere i due statistici!).

Wilcoxon (1945) ha proposto il suo test per valutare l'H_0 di appartenenza alla stessa popolazione di due gruppi indipendenti X_1 e X_2 **di uguale numerosità** (successivamente è stato esteso a $N_1 \neq N_2$), per $\boldsymbol{Y}$ **almeno ordinale**, purché le distribuzioni di Y in X_1 e X_2 siano **continue** e abbiamo la stessa forma rispetto alla simmetria (o **entrambe simmetriche** o **entrambe asimmetriche**). La logica del test è semplice: si considerano tutte le osservazioni raccolte, indipendentemente dal gruppo ($N_1 + N_2$), e si sostituisce ogni osservazione con il rispettivo rango, di solito in senso crescente, per cui i valori bassi hanno ranghi bassi (1,2,3…) e i valori più alti avranno i ranghi più alti. Se X non ha effetto su Y, avremo una distribuzione casuale di ranghi alti e bassi nei due gruppi: perciò, dopo aver assegnato i ranghi a tutte le osservazioni e poi calcolato **separatamente per gruppo** le **sommatorie** dei **ranghi** ($\sum_{R_{X1}}$ e $\sum_{R_{X2}}$), se $\sum_{R_{X1}} = \sum_{R_{X2}}$ confermeremo H_0. Se invece si troverà che $\sum_{R_{X1}} \neq \sum_{R_{X2}}$, allora probabilmente X avrà esercitato un effetto nell'accumulare i ranghi bassi e i ranghi alti in due gruppi diversi. La $\boldsymbol{\sum_R}$ **più piccola** tra le due (chiamata $\boldsymbol{W}$ o T) è confrontata con valori critici per confermare o disconfermare H_0. Se ci sono casi con valori uguali (**ties**), a ciascuno è assegnato il **rango medio**, cioè la somma dei ranghi che sarebbero loro assegnati divisa per il numero di valori pari merito, già visto nel §3.2.3: `rank(Y, ties.method="average")`.

[90] Ad esempio, Moser e Stevens, 1992; Jerrold, 1996, pag. 663: "*If there are severe deviations from the normality and/or equality-of variance assumptions, the nonparametric test could be employed, as it is not adversely affected by violations of these assumptions, and some researchers would prefer that procedure to the modified test above*."

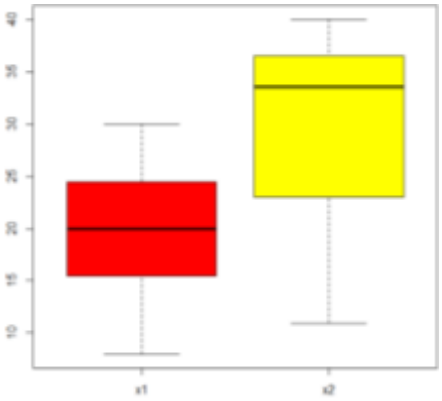
Successivamente, il test è stato integrato da Mann e Whitney (1947): in questa versione, preferita perché **non** richiede il **requisito di simmetria** e quindi è più generalizzabile, è chiamato **test robusto U di Wilcoxon-Mann–Whitney** ed è **quello che ci presenta R**, che però, confondendoci leggermente le idee, lo chiama `wilcox.test(Y~X)`.

La procedura del test U non si basa sull'attribuzione dei ranghi, ma sulle **precedenze**, cioè un'operazione che può ricordare quella del test τ di Kendall. Come nel test di Wilcoxon, si inizia ordinando in senso crescente tutte le osservazioni raccolte, indipendentemente dal gruppo ($N_1 + N_2$), poi la procedura cambia: si conta **quante volte ogni dato di un gruppo è preceduto da dati dell'altro gruppo**. La statistica del test ($\boldsymbol{U}$) è il **numero minore di precedenze**, e a essa è attribuito un $p - value$; il numero maggiore di precedenze è chiamato $\boldsymbol{U'}$. U e U' sono legati dalla relazione $\boldsymbol{N_1 \times N_2 = U + U'}$, in cui N_1=numerosità del gruppo minore e N_2= numerosità del gruppo maggiore.

Chiariamo la procedura con un esempio, che tra l'altro rende abbastanza problematico anche l'assunto di indipendenza degli errori: riprendiamo i cagnolini dell'allevatrice delusa (§8.4.1) e cerchiamo di aiutarla a migliorare i suoi risultati ai concorsi. Sono risultati significativamente più belli i cagnolini nati dalla coppia Tato e Tata (X_1) o quelli nati dalla coppia Tito e Tita (X_2)? Consideriamo i punteggi ottenuti negli ultimi concorsi per la valutazione estetica di due cucciolate per coppia, per allargare un po' il campione.

```
punteggi<-c(8,15,16,20,23,26,30,11,22,24,32,35,36,40,37)
cucciolata<-c(rep("x1",7),rep("x2",8))
mammapapa<-data.frame(punteggi,cucciolata)
mammapapa <-mammapapa[order(mammapapa$punteggi),]
mammapapa
```

	punteggi	cucciolata
1	8	x1
8	11	x2
2	15	x1
3	16	x1
4	20	x1
9	22	x2
5	23	x1
10	24	x2
6	26	x1
7	30	x1
11	32	x2
12	35	x2
13	36	x2
15	37	x2
14	40	x2



Contiamo quante volte i punteggi di X_1 sono preceduti da X_2:

- 8 non è preceduto da alcun valore= 0 precedenze;
- 15,16 e 20 sono preceduti ciascuno da un solo valore $X_2(11)$. Il loro totale è = 3 precedenze;
- 23 è preceduto da due valori $X_2(11, 22)$= 2 precedenze;
- 26 e 30 sono preceduti ciascuno da tre valori $X_2(11, 22, 24)$. Il loro totale è = 6 precedenze.

Quindi, $0 + 2 + 3 + 6 =$ **11 precedenze complessive per X_1**

Ora contiamo quante volte i punteggi di X_2 sono preceduti da X_1:

- 11 è preceduto da un valore $X_1(8)$ = 1 precedenza;
- 22 è preceduto da quattro valori $X_1\ (8, 15, 16, 20)$= 4 precedenze;
- 24 è preceduto da cinque valori $X_1(8, 15, 16, 20, 23)$= 5 precedenze;
- 32, 35, 36, 37, e 40 sono preceduti ciascuno da sette valori $X_1(8, 15, 16, 20, 23, 26, 30)$. Il loro totale è = 35 precedenze.

Quindi, $1 + 4 + 5 + 35 =$ **45 precedenze complessive per X_2**. La statistica di riferimento U è la somma minore tra le due, perciò $\boldsymbol{U = 11}$.

Per fortuna, R ci evita di calcolare le precedenze, fornendoci U e il relativo $p - value$ nell'output di `wilcox.test(Y~X)`; `paired= FALSE` (default) indica di effettuare un test per campioni indipendenti e non per dati appaiati. È possibile richiedere il CI con l'argomento `conf.int=TRUE`, e variare la sua verosimiglianza (di default .95) con `conf.level=`. La statistica campionaria attorno alla quale è costruito il CI (`difference in location`) è lo stimatore di **Hodges e Lehmann** per due campioni indipendenti: è la **mediana di tutte le differenze a coppie** tra i valori di X_1 e di X_2X ($8-11, 8-22, 8-24, \ldots, 30-40, 30-37$), che, secondo H_0, in popolazione dovrebbe essere $stimatore = 0$ (`true location shift`):

- $\boldsymbol{H_0}$: $\boldsymbol{mediana_{\Delta_{x1-x2}} = 0}$
- $\boldsymbol{H_1}$: $\boldsymbol{mediana_{\Delta_{x1-x2}} \neq 0}$ oppure $\boldsymbol{mediana_{\Delta_{x1-x2}} > 0}$ oppure $\boldsymbol{mediana_{\Delta_{x1-x2}} < 0}$

Nel nostro esempio:

```
wilcox.test(mammapapa$punteggi~ mammapapa$cucciolata,conf.level = .99, conf.int = TRUE)
        Wilcoxon rank sum test
data:  mammapapa$punteggi by mammapapa$cucciolata
W = 11, p-value = 0.05408
alternative hypothesis: true location shift is not equal to 0
99 percent confidence interval:  ← CI della mediana delle differenze a coppie: contiene H0: mediana x1-x2= 0
 -24   5
sample estimates:
difference in location        ← stimatore di Hodges-Lehman: mediana delle differenze a coppie tra X1 e X2
                 -10.5
```

Quindi, la mediana delle differenze a coppie tra i valori di X_1 e X_2 nel campione è $stimatore = -10.5$ (il gruppo X_1 ha valori mediamente inferiori), e in popolazione la vera mediana delle differenze sta tra -24 e +5, con il 95% di verosimiglianza: il CI contiene il valore previsto da H_0. L'allevatrice è davvero sfortunata: anche questa volta, non è possibile rifiutare H_0 con serenità.

Volendo (?) calcolare lo stimatore di Hodges e Lehmann dai dati, si costruisce la matrice delle differenze tra ogni x_{1i} e ogni x_{2i} con la funzione **`outer(x, y, FUN="funzione")`**, che produce il **risultato dell'operazione** richiesta da `FUN=` tra **due vettori o matrici** (`x` e `y`); nel nostro esempio, avremo $7 * 8 = 56$ differenze a coppie, in una matrice composta da 7 righe (N_1) e 8 colonne (N_1), a cui, per maggior chiarezza, assegniamo gli elementi di X_1 e X_2 come nomi per le righe e le colonne:

```
x1<-c(8,15,16,20,23,26,30)
x2<-c(11,22,24,32,35,36,37,40)
matrice_differenze_a_coppie<-outer(x1,x2,"-")
rownames(matrice_differenze_a_coppie)<-x1; colnames(matrice_differenze_a_coppie)<-x2
matrice_differenze_a_coppie
   11  22  24  32  35  36  37  40
8  -3 -14 -16 -24 -27 -28 -29 -32
15  4  -7  -9 -17 -20 -21 -22 -25
16  5  -6  -8 -16 -19 -20 -21 -24
20  9  -2  -4 -12 -15 -16 -17 -20
23 12   1  -1  -9 -12 -13 -14 -17
26 15   4   2  -6  -9 -10 -11 -14
30 19   8   6  -2  -5  -6  -7 -10
median(matrice_differenze_a_coppie)
[1] -10.5
```

Notate che nell'ouput la statistica è chiamata W, come nella versione originale del test di Wilcoxon, e che il test è chiamato `Wilcoxon rank sum test`. In effetti, è possibile ricavare U / W anche passando dalla somma dei ranghi del test di Wilcoxon, ottenendo un risultato del tutto equivalente: alla $\sum_R$ più piccola tra le due si sottrae il **minimo valore**

teoricamente ottenibile della $\sum_R$, cioè quello che si avrebbe se tutti i soggetti del gruppo in questione occupassero le prime posizioni della distribuzione ordinata.

Nel nostro esempio, dopo aver calcolato i ranghi del punteggio per l'intero campione, calcoliamo le sommatorie dei ranghi delle due cucciolate:

```
mammapapa$ranghi<-rank(mammapapa$punteggi, ties.method = "average")
tapply(mammapapa$ranghi, mammapapa$cucciolata, sum)
x1 x2
39 81
```

Il riferimento è la $\sum_R$ del gruppo X_1. Se i sette cuccioli di X_1 avessero i sette punteggi più bassi, cioè occupassero le prime sette posizioni, la loro $\sum_R$ sarebbe:

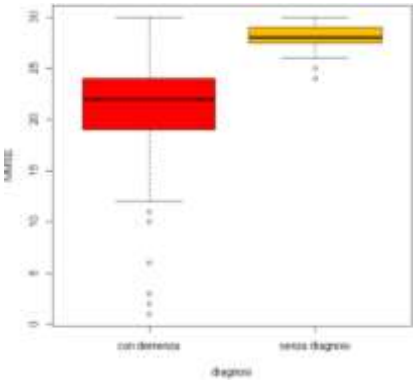
```
S_ranghi_minima<-1+2+3+4+5+6+7
S_ranghi_minima
[1] 28
```

Quindi, la statistica W è data dalla $\sum_R$ effettivamente rilevata nel gruppo X_1 **meno** la $\sum_R$ minima ottenibile da X_1:

```
(W<-39-28)
[1] 11
```

Vediamo il test applicato ai dati veri e decisamente più numerosi di `vecchietti`. Il GPCOG è risultato diverso nei due gruppi con e senza diagnosi di demenza: esiste un'analoga differenza tra i soggetti con e senza diagnosi nel test Mini Mental State Examination (MMSE), che è abitualmente usato come test di screening per il funzionamento cognitivo patologico?

```
tapply(v$MMSE,v$diagnosi,summary)
$`con demenza`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
1.00   19.00   22.00   20.74   24.00   30.00
$`senza diagnosi`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
24.00   27.75   28.00   28.17   29.00   30.00

MMSE<-lm(v$MMSE~v$diagnosi)
shapiro.test(MMSE$residuals)
       Shapiro-Wilk normality test
data:  MMSE$residuals
W = 0.87086, p-value = 4.993e-12
leveneTest(v$MMSE, v$diagnosi)
Levene's Test for Homogeneity of Variance (center = "median")
       Df F value    Pr(>F)
group   1  26.008 7.928e-07
      198
```



L'analisi dei requisiti del modello non lascia illusioni: usiamo un approccio non parametrico.

```
wilcox.test(v$MMSE~v$diagnosi,conf.int = T)
      Wilcoxon rank sum test with continuity correction
data:  vecchietti$MMSE by vecchietti$diagnosi
W = 346, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -7.000007 -5.999978
sample estimates:
difference in location
             -6.999938
```

Anche questo test neuropsicologico discrimina in maniera significativa tra i soggetti con e senza diagnosi. La mediana delle differenze tra i punteggi dei due gruppi è $= -6.99$ nel campione (i soggetti con diagnosi – X_1 hanno punteggi mediamente inferiori), e oscilla in popolazione entro un intervallo piuttosto ristretto (da -7.01 a -5.99) che non contiene il valore previsto da $H_0 = 0$.

Quando i campioni sono sufficientemente numerosi (***N* > 50**, in R), per l'attribuzione del $p - value$ si sfrutta una **approssimazione alla distribuzione normale**: ecco la ***continuity correction*** (*correction to continuity*) dell'output, gestita dall'argomento **correct=** che di default è **TRUE**. Il quantile z cui è associato il $p - value$ ottenuto o uno inferiore non è mostrato nell'output, ma all'occorrenza può essere ricavato con la nota funzione `qnorm(q, mean, sd, lower.tail)`; l'argomento `q` è il $p - value$ del test di Wilcoxon, diviso per due se H_1 era stata impostata come bidirezionale: `qnorm(q= p value/2, mean= 1, sd=1, lower.tail= FALSE)`. Se invece nella distribuzione non ci sono *ties* e i soggetti non sono molti, si ottiene un $p - value$ esatto (**exact=TRUE**): nel caso si specifichi questo argomento come **TRUE** e ci siano *ties*, R avvisa che è stata fatta una richiesta inadeguata e riporta l'approssimazione alla distribuzione normale:

```
wilcox.test(v$MMSE~v$diagnosi,exact = TRUE)
        Wilcoxon rank sum test with continuity correction
data:  vecchietti$MMSE by vecchietti$diagnosi
W = 346, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
Warning message:
In wilcox.test.default(x = c(27L, 26L, 27L, 27L, 22L, 30L, 27L,  :
impossibile calcolare p-value esatto in presenza di ties
```

Cosa sarebbe successo se invece del test non parametrico avessimo adattato ai dati delle cucciolate un test parametrico, sia nella versione corretta per eteroschedasticità sia ignorando la violazione del requisito? Avremmo – con qualche difficoltà – respinto H_0, dimostrando che effettivamente l'approccio non parametrico è un po' meno potente, in presenza di pochi dati:

```
t.test(mammapapa$punteggi~ mammapapa$cucciolata, var.equal = FALSE)
        Welch Two Sample t-test
data:  mammapapa$punteggi by mammapapa$cucciolata
t = -2.2229, df = 12.77, p-value = 0.04493
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -19.5601074  -0.2613212
sample estimates:
mean in group x1 mean in group x2
        19.71429         29.62500

t.test(mammapapa$punteggi~ mammapapa$cucciolata, var.equal = TRUE)
        Two Sample t-test
data:  coppie$punteggi by coppie$cucciolata
t = -2.1798, df = 13, p-value = 0.04826
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -19.7332584  -0.0881702
sample estimates:
mean in group x1 mean in group x2
        19.71429         29.62500
```

E per stimare l'intensità dell'effetto? Non sono molte le misure di ES per i test non parametrici; nel caso di due gruppi, si può usare il coefficiente ***delta* di Cliff**, che altro non è che il **coefficiente di correlazione biseriale** (Pearson, ricordate?) **applicato ai ranghi** dei **due gruppi**. Lo troviamo in `effectsize` con la funzione `cliffs_delta(formula)`, di default `paired= FALSE`. Varia da -1 a +1, come qualsiasi coefficiente di correlazione: valori prossimi a -1 indicano che tutti i valori del secondo campione tendono a essere più grandi di quelli del primo campione, mentre valori tendenti a +1 indicano che i valori del primo campione tendono a essere più grandi. Nell'esempio della differenza del MMSE tra i gruppi, dovremmo aspettarci che i ranghi del secondo gruppo ("senza diagnosi") siano quasi universalmente maggiori dei ranghi del primo ("con diagnosi"), e quindi un delta negativo:

```
cliffs_delta(v$MMSE~v$diagnosi)
r (rank biserial) |         95% CI
---------------------------------
-0.92             | [-0.94, -0.89]
```

Infatti… la differenza va nella direzione attesa ed è di forte intensità.

Le **statistiche robuste di Wilcox** si basano su **stimatori robusti**, le cui stime non sono distorte dalla presenza di casi anomali; tra gli stimatori di tendenza centrali robusti più usati troviamo mediane e medie *trimmed*, tra gli indicatori di dispersione robusti abbiamo la varianza ***winsorized***. Una distribuzione è *winsorized* (da Charles Winsor) quando un determinato **quantitativo dei suoi valori estremi è sostituito da valori meno estremi**: a differenza di una distribuzione *trimmed*, quindi, N resta immutato. I valori sostituitivi sono rispettivamente il più basso e il più alto tra quelli rimanenti[91].

Questi test robusti sono disponibili nel package `WRS2` (Wilcox Robust Statistics, Wilcox, 2012): troviamo diverse possibilità applicabili per un test robusto con X a due livelli.

Le funzioni `t1way` e `med1way` sono **l'analogo non parametrico di ANOVA**; possono essere applicate **anche per X a più di due livelli** (le ritroveremo nel capitolo 12), e non richiedono omoschedasticità. La funzione `t1way(y~x, trim=)` è una generalizzazione del test di Welch che lavora su medie troncate (`trim=` accetta al massimo una proporzione=.25 per ogni coda), mentre `med1way(y~x, data)` esegue un'ANOVA su mediane e non dovrebbe essere utilizzata in presenza di molti ties.

```
t1way(mammapapa$punteggi~mammapapa$cucciolata, trim=.20)
Call:
t1way(mammapapa$punteggi~ mammapapa$cucciolata, trim=.20)

Test statistic: 6.6708
Degrees of freedom 1: 1
Degrees of freedom 2: 8.94
p.value: 0.02971

t1way(mammapapa$punteggi~mammapapa$cucciolata, trim=.25)
Call:
t1way(mammapapa$punteggi~ mammapapa$cucciolata, trim=.25)

Test statistic: 5.0667
Degrees of freedom 1: 1
Degrees of freedom 2: 5.1
p.value: 0.07314

med1way(punteggi~cucciolata, data=mammapapa)
Call:
med1way(punteggi~cucciolata, data=mammapapa)

Test statistic: 3.6504
Critical value: 2.6964
p.value: 0.024
```

Due funzioni sono dedicate **esclusivamente** a **una X a due livelli**: **`yuen`** esegue il test di Yuen (1974) su medie tronche, mentre **`pb2gen`** esegue un t-test basato su stimatori robusti che possono essere selezionati con l'argomento `estim="stimatore"`.

```
yuen (punteggi~cucciolata, data=mammapapa, trim=.2)
Call:
yuen (punteggi~cucciolata, data= mammapapa, trim=.2)

Test statistic: 2.5828 (df = 8.94), p-value = 0.02971
Trimmed mean difference: -11
95 percent confidence interval:
-20.6438     -1.3562
```

[91] Potete usare `Winsorize(distribuzione)` di `DescTools` se volete vederne un esempio: è sostituito il 5% dei valori più alti e più bassi (`minvalue=, maxvalue=`). Con `Trim(distribuzione)` potete confrontare l'esito dei due metodi.

Come si può notare, quando X è due livelli il risultato è perfettamente **equivalente** **a quello di** `t1way` per la stessa proporzione di medie troncate.

```
pb2gen(punteggi~cucciolata, data=mammapapa, estim= "mom")
Call:
pb2gen(punteggi~cucciolata, data= mammapapa, estim= "mom")

Test statistic: -12.5714, p-value = 0.06177
95 percent confidence interval:
-21.1429     .525
```

`mom` è lo stimatore MOM (*modified one-step estimator*) basato sullo stimatore Psi di Huber.

Recuperate il dataframe ***gamblers****:*

- *verificate se maschi e femmine giocano lo stesso numero di giorni alla settimana e lo stesso numero di ore al giorno;*
- *verificate se la gravità del disturbo distimico e quella del disturbo d'ansia siano uguali nei due generi*

10.2 Test per disegni within subjects

In questo paragrafo confrontiamo ancora le medie di due distribuzioni, ma questa volta **appaiate o dipendenti**.
Siamo quindi nel campo di disegni **whitin subjects**, che possono essere:

- **trasversali**: **un** gruppo i cui soggetti sono stati casualmente estratti dalla stessa popolazione e sottoposti a **entrambi i livelli** di una variabile indipendente X, nel corso di un'**unica somministrazione**;
- **longitudinali**: **un** gruppo i cui soggetti sono stati casualmente estratti dalla stessa popolazione e sottoposti a **entrambi i livelli** di una variabile indipendente X, nel corso di **due somministrazioni** (**prima-dopo**).

In entrambi i casi, H_1 prevede una **differenza**, **attribuibile all'effetto di X**, tra la prestazione dello stesso soggetto al livello X_1 e quella al livello X_2, mentre H_0 **nega l'effetto di X** sulla prestazione, e pertanto **prevede un'assenza di differenza** tra la performance al livello X_1 e quella al livello X_2 dello stesso soggetto.

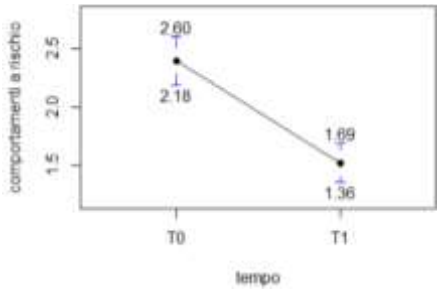
10.2.1 ANOVA a misure ripetute per una X a due livelli

Nell'ANOVA **a misure ripetute**, che affrontiamo qui nella sua forma più semplice (**una sola Y**, **una sola X within subjects** a **due livelli**), assistiamo una particolare **suddivisione della devianza totale** rispetto a quella vista nell'ANOVA **between** groups (§10.1.1). In quest'ultima, la variabilità complessiva di Y (SS_{totale}) è spartita tra la variabilità attribuita **all'effetto di** $X_{Gruppo\ di\ appartenenza}$ (SS_M o $SS_{between}$) e quella residua o **d'errore** attribuita alle differenze individuali **entro** ciascun gruppo (SS_R o SS_w). In ANOVA a misure ripetute, poiché **tutti** i soggetti sono esposti a **tutti** i livelli di X, si ricerca l'**effetto del predittore entro i soggetti**, distinguendolo **la variabilità del soggetto da X_1 a X_2 attribuita all'effetto di X da quella attribuita all'errore** (tutto ciò che non è X):

La devianza tra i soggetti SS_B, che raccoglie le differenze interindividuali, non partecipa al calcolo del rapporto F: è quindi opportuno avere campioni **ragionevolmente omogenei** per le caratteristiche in analisi, in modo da concentrare la variabilità di Y sulle misure entro ogni soggetto.

Vediamo le modalità di partizione usando il dataframe **sicurezza**, che racchiude variabili relative alla **valutazione di efficacia di un corso sulla sicurezza** nell'ambiente di lavoro: i suoi obiettivi erano aumentare le **conoscenze** sulla sicurezza, favorire gli **atteggiamenti** positivi, ridurre i **comportamenti pericolosi** sul luogo di lavoro e migliorare la **salute** dei partecipanti. Per ciascun obiettivo sono state prese misure prima del corso (T_0), subito dopo il suo termine (T_1) e dopo tre mesi dalla fine del corso (T_2). **Per ora**, concentriamoci sui **comportamenti a T_0 e T_1**: il corso è stato efficace nel **ridurre i comportamenti a rischio** sul posto di lavoro, registrati su una scheda ad hoc da un osservatore, lavoratore per lavoratore?
Il corso è stato erogato in due diverse modalità (obbligatorio versus non obbligatorio), ma useremo questa variabile solo nell'ANOVA fattoriale mista; c'è **anche un gruppo di controllo**, che non ha seguito il corso: **eliminiamo** questi soggetti (per ora), e teniamo i 91 soggetti che hanno frequentato. Creiamo il subset **s**, in cui esportiamo solo le variabili che ci servono: l'identificativo dei soggetti e le due rilevazioni dei comportamenti a rischio, a T_0 e T_1:

```
s<-subset(sicurezza, sicurezza$gruppo!="controllo",select = c(codice,comportamenti_t0, comportamenti_t1))
head(s, 3)
  codice comportamenti_t0 comportamenti_t1
1 BC0Y29              2.3              1.2
2 BG3M14              3.8              1.0
3 BL9D18              2.5              1.3

summary(s$comportamenti_t0);summary(s$comportamenti_t1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.750   2.300   2.392   2.900   5.400
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   1.200   1.525   1.700   4.800
which(scale(s$comportamenti_t0)>=2)
[1] 23 43 56 88
which(scale(s$comportamenti_t1)>=2)
[1] 44 46 77 88
```



La media dei comportamenti a rischio a T_1 si è complessivamente abbassata, nonostante persistano outliers più a rischio a T_0 e T_1.
Vediamo media e varianza **complessive:** sono rispettivamente chiamate **grand mean** e **grand variance**, e sono **indipendenti** dalle condizioni sperimentali:

```
totale<-cbind(s$comportamenti_t0,s$comportamenti_t1)
length(totale)
[1] 182
(grand_mean<-mean(totale))
[1] 1.958791
```

Il numero medio di comportamenti a rischio, indipendentemente dal momento in cui è stato rilevato, è $\bar{x} = 1.96$.

```
(grand_variance<-sd(totale)^2)
[1] 0.9937068
```

Come già avevamo visto, dato che $MS = SS/df$, otteniamo la devianza con $SS = MS \times df$; i df per la **devianza totale sono dati dal $numero\ di\ misure - 1$**. Quindi, dalla **grand variance** possiamo ricavare la **devianza totale SS_T, indipendentemente dal momento della rilevazione:**

```
(SS_T<-grand_variance*(182-1))
[1] 179.8609
```

Procediamo a calcolare SS_W: la **variabilità nei comportamenti a rischio di ogni soggetto** nel passare da prima a dopo ($\boldsymbol{SS_W}$) è data dalla somma degli **scarti al quadrato dalla media di ciascun soggetto**. I suoi df corrispondono al numero di soggetti ($\boldsymbol{N}$) per il numero di condizioni $\boldsymbol{k-1}$: $df_w = N(k-1) = 91$

```
s$media<-(s$comportamenti_t0+s$ comportamenti_t1)/2
s$scarti<-(s$ comportamenti_t0-s$media)^2+(s$ comportamenti_t1-s$media)^2
head(s,4)
```

	sogg	comportamenti_T0	comportamenti_T1	media	scarti
1	BC0Y29	2.3	1.2	1.75	**0.605**
2	BG3M14	3.8	1.0	2.40	**3.920**
3	BL9D18	2.5	1.3	1.90	**0.720**
4	BR3E44	1.7	1.7	1.70	**0.000**

```
(SS_W<-sum(s$scarti))
[1] 79.625
```

Questa variabilità entro i soggetti passando da una condizione all'altra può essere dovuta sia all'effetto del corso (X), sia ad altri eventi (stanchezza, motivazione, ecc.: errore): ora **dovremo separare l'effetto di X (SS_M) dall'errore (SS_R)**. Si calcola prima la variabilità attribuita a X – SS_M, data dalla **somma** degli **scarti** al **quadrato della media in ogni condizione dalla grand mean**, **moltiplicati** per il **numero di soggetti**. I suoi df sono dati dal ***numero di condizioni*** $\boldsymbol{k-1}$: $\boldsymbol{df_M = 1}$

```
(SS_M<-91*((mean(s$comportamenti_t0)-grand_mean)^2)+91*((mean(s$comportamenti_t1)-
grand_mean)^2))
[1] 34.20445
```

Ne consegue che la variabilità entro i soggetti attribuita all'errore $\boldsymbol{SS_R}$ è data dalla **differenza** tra la complessiva variabilità within subjects SS_W e questa quota di variabilità attribuita a X, SS_M. I suoi df sono dati dal ***numero di soggetti*** $\boldsymbol{N-1}$**:** $\boldsymbol{df_R = 90}$

```
(SS_R<-SS_W-SS_M)
[1] 45.42055
```

Anche se non partecipa al calcolo di F, la variabilità tra i soggetti SS_B si può calcolare (e poi dimenticare) come differenza tra la variabilità totale SS_T e la variabilità within subjects SS_W:

```
(SS_B<-SS_T-SS_W)
[1] 100.2359
```

Avendo ora SS_M e SS_R, trasformiamole in varianze MS_M e MS_R per leggerne il rapporto F e relativo $p-value$:

```
(MS_M<-SS_M/1)
[1] 34.20445
(MS_R<-SS_R/90)
[1] 0.5046728
(F<-MS_M/MS_R)
[1] 67.7755
pf(67.7755, df1 = 1, df2 = 90, lower.tail = FALSE)
[1] 1.34948e-12
```

Sì, il numero dei comportamenti a rischio dei soggetti da T_0 a T_1 è significativamente diminuito – ma, in realtà, **non** abbiamo dimostrato in maniera inequivocabile che sia stato l'aver frequentato il corso ad abbassarlo, dato che molte possibili covariate non sono state inserite nel modello.

Quali sono le covariate più plausibili che avremmo dovuto controllare? Qual è il rimedio più ovvio per dare una maggior credibilità all'ipotesi che sia stato davvero il corso di formazione, e non semplicemente il passare del tempo, a determinare una variazione significativa nei comportamenti?

Naturalmente, R ci esonererà, d'ora in poi, da tutti questi passaggi, anche se ci pone un problema preliminare: la **struttura del dataframe**. Finora, abbiamo usato dataframe in ***wide format*** (§2.2), in cui ogni riga racchiude tutte le misure relative a un singolo soggetto, ciascuna rappresentata in una colonna. **Nelle misure ripetute,** dobbiamo inserire i dati in una struttura in ***long format***, in cui **ogni riga contiene un livello di un soggetto**.

Come perlopiù capita, questo significa **trasporre un dataframe** in *wide format* in *long format*: possiamo usare **`melt`** di **`reshape2`**, in cui indichiamo: **`data=`** il dataframe di partenza (se R lo legge come *wide format*, lo trasforma in *long*, e viceversa); **`id.vars=`** la variabile o le variabili che rappresentano **misure uniche** del soggetto (in questo esempio solo il codice del soggetto: `$codice`); **`measure.vars=`** le misure **ripetute** della variabile X entro i soggetti (`$T0` e `$T1`).

```
melt_s <- melt(data = s,id.vars = "codice", measure.vars = c("comportamenti_t0", "comportamenti_
  t1"))
```

```
head(melt_s)
  codice         variable value
1 BC0Y29 comportamenti_t0   2.3
2 BG3M14 comportamenti_t0   3.8
3 BL9D18 comportamenti_t0   2.5
4 BR3E44 comportamenti_t0   1.7
5 BR3P27 comportamenti_t0   1.7
6 BR3S10 comportamenti_t0   1.3
```

```
tail(melt_s)
    codice          variable value
177 SL7X03 comportamenti_T1   1.0
178 SM3G22 comportamenti_T1   1.7
179 SM5H02 comportamenti_T1   4.1
180 SR3C10 comportamenti_T1   2.5
181 VN3E28 comportamenti_T1   2.5
182 VS2J04 comportamenti_T1   2.2
```

`Variable` e **`value`** sono le etichette di default; con la funzione **`names`** possiamo rinominarle in maniera più esplicita:

```
names(melt_s)<-c("sogg","rilevazioni","comportamenti")
```

Se la visualizzazione risulta più chiara, possiamo ordinare per partecipante anziché per livello di X, con **`order()`**:

```
melt_s<-melt_s[order(melt_s$sogg),]
head(melt_s)
      sogg      rilevazioni comportamenti
1   BC0Y29 comportamenti_t0           2.3
92  BC0Y29 comportamenti_t1           1.2
57  BC1R11 comportamenti_t0           3.3
148 BC1R11 comportamenti_t1           2.1
2   BG3M14 comportamenti_t0           3.8
93  BG3M14 comportamenti_t1           1.0
```

Abbiamo già descritto i dati, quindi possiamo passare a vedere il metodo più semplice (e **semplicistico**) per condurre un'ANOVA a misure ripetute in R, con **`ezANOVA (data, soggetti, Y, X a misure ripetute)`** del package **`ez`**: questa funzione gestisce anche ANOVA between groups, e ci sarà utilissima nell'ANOVA fattoriale, con più X e disegni non bilanciati (capitolo 13), Quando affronteremo il caso delle misure ripetute in una X a più di due livelli (capitolo 12), accenneremo a un altro metodo (i ***mixed models***, oggetto del Capitolo 15), e il suo output sarà integrato da informazioni necessarie per la **verifica dei prerequisiti** in un modello a misure ripetute con livelli $k > 2$.

Gli argomenti di ezAnova per un disegno within con un solo predittore sono: **`data=`** **dataframe in long format**, **`dv=`** ***Y*** (dependent variable), **`wid=`** **variabile che identifica il soggetto**, **`within=`** ***X*** **a misure ripetute**. Tra gli argomenti opzionali, specifichiamo **`detailed= TRUE`** (di default **`= FALSE`**) per visualizzare anche la riga della b_0 del modello lineare e le SS. L'output, nel caso di una X a due livelli, è assai stringato: riporta solo i test di significatività di b_0 e b_1 (effetto di X), una **misura di intensità dell'effetto** e le SS_M e SS_R con relativi df (ma non le MS):

```
ezANOVA(data = melt_s,wid = sogg, dv = comportamenti,within = rilevazioni,detailed = TRUE)
Warning: You have removed one or more Ss from the analysis. Refactoring "sogg" for ANOVA.
$ANOVA
       Effect DFn DFd       SSn       SSd        F            p p<.05       ges
1 (Intercept)   1  90 698.30907 100.23593 626.9989 2.481106e-42     * 0.8274142
2 rilevazioni   1  90  34.20445  45.42055  67.7755 1.349480e-12     * 0.1901716
                ↑   ↑       ↑         ↑                                    ↑
               dfM dfR     SSM       SSR                          indice di effect size
```

```
ezANOVA(data = melt_s,wid = sogg, dv = comportamenti,within = rilevazioni,detailed = FALSE)
Warning: You have removed one or more Ss from the analysis. Refactoring "sogg" for ANOVA.
$ANOVA
       Effect DFn DFd         F       p p<.05       ges
2 rilevazioni   1  90  67.7755 1.349480e-12     * 0.1901716
```

Già sapevamo che il $p - value$ del rapporto F cade nella regione di rifiuto di H_0. **ges** sta per **generalized eta squared**: è una misura di **effect size**, che, per **una sola X**, è interpretabile **come R^2**. **SSn** e **SSd** sono rispettivamente la SS_M (al numeratore -**n**- di F) e la SS_R(al denominatore – **d** – di F), che avevamo calcolato "a mano"; i rispettivi DFn e DFd sono i df_M e df_R:

```
SS_M; SS_R
[1] 34.20445
[1] 45.42055
```

Dato che stiamo lavorando su un subset, R ci avvisa con un *warning* (che naturalmente possiamo ignorare). Inoltre, il *warning* ricorda che la variabile `$sogg` è stata considerata come un `factor` per l'analisi: in effetti, ora **ogni soggetto** rappresenta **un livello di un fattore**, al cui interno sono raggruppate le misure di ciascuno.

10.2.2 t-test per dati appaiati (o campioni dipendenti)

Come in ANOVA a misure ripetute, l'H_0 cui si applica il t-test per dati appaiati è che le distribuzioni X_1 e X_2, derivate dagli stessi soggetti, provengano dalla **medesima popolazione**, per cui $\bar{x}_1$ e $\bar{x}_2$ sono **solo casualmente differenti**: il predittore X non ha un effetto significativo. Sono quindi valutate le **differenze in ogni coppia di osservazioni**: H_0 afferma che la stessa unità statistica otterrebbe lo stesso punteggio in entrambe le condizioni, o al più mostrerebbe solo **variazioni casuali** attorno allo stesso punteggio "vero", quello atteso nella popolazione di appartenenza. Come nel t-test per campioni indipendenti, H_1 può essere monodirezionale.

- $\boldsymbol{H_0}$: $\boldsymbol{\mu_1 = \mu_2}$
- $\boldsymbol{H_1}$: $\boldsymbol{\mu_1 \neq \mu_2}$ oppure $\boldsymbol{\mu_1 > \mu_2}$ oppure $\boldsymbol{\mu_1 < \mu_2}$

Il test consente di attribuire un $p - value$, utilizzando la distribuzione t per $\boldsymbol{df = N - 1}$, alla differenza osservata tra X_1 e X_2: quanto è probabile, se le due distribuzioni appartengono alla stessa popolazione, riscontrare una differenza quale quella osservata?

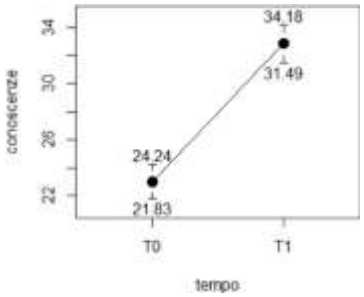
Si tratta di valutare la **differenza tra le medie** dei due livelli, **ponderata per l'errore standard della differenza tra le medie** e **corretta per la relazione tra le due misure** dello stesso soggetto: è la relazione tra le due misure dello stesso soggetto a fare la differenza con il t-test per campioni indipendenti. In quel caso, ricordiamo che lo SE della differenza fra le medie al denominatore era stato sostituito dalla somma delle radici quadrate delle stime di s^2 di X_1 e X_2 sotto **l'assunzione dell'indipendenza delle misure** – che, in **un disegno entro soggetti non regge più**: la variabilità intraindividuale, tra le prestazioni di uno stesso individuo, è minore di quella interindividuale, tra le prestazioni di individui diversi.

Quando le distribuzioni sono **dipendenti**, la **varianza della differenza** tra le distribuzioni è uguale alla **somma delle corrispondenti varianze, meno due volte la covarianza**.

$$t_{N-1} = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{s_1^2 + s_2^2 - 2cov_{x1x2}}{N}}}$$

Proviamo a dimostrarlo con un esempio. Torniamo al dataframe `sicurezza` e vediamo se le **conoscenze** sulle corrette pratiche preventive e sulle norme sono aumentate, da T_0 a T_1, in chi ha frequentato il corso sulla sicurezza. H_0 è che le conoscenze dei soggetti prima di iniziare il corso non siano significativamente diverse da quelle da loro acquisite dopo il corso. Escludiamo ancora il gruppo di controllo e ignoriamo l'informazione sull'obbligatorietà del corso.

```
conoscenze<-subset(sicurezza, sicurezza$gruppo!="controllo", select=c("codice", "conoscenze_t0", "conoscenze_t1"))
names(conoscenze)<-c("sogg","T0","T1")
str(conoscenze)
'data.frame':91 obs. Of  3 variables:
 $ sogg: Factor w/ 123 levels "BC0Y29","BC1R11",..: 1 3 ..
 $ T0  : num  34 28 32 21 7 15 19 20 20 15 …
 $ T1  : num  42 39 41 36 11 28 30 33 37 28 …
summary(conoscenze$T0); summary(conoscenze$T1)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
7.00   19.50   24.00   23.03   27.00   37.00
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
10.00   29.00   34.00   32.84   38.00   43.00
which(scale(s$comportamenti_t0)<=-2);
which(scale(s$comportamenti_t1) <= -2)
[1] 22
integer(0)
sicurezza[22,2]
[1] formazione obbligatoria
```



Le conoscenze sembrano decisamente aumentate dopo il corso. Il soggetto (riga 22) del gruppo con formazione obbligatoria che aveva nettamente meno conoscenze dei colleghi a T_0 rientra nel gruppo a T_1.

Calcoliamo la **distribuzione delle differenze tra T0 e T1**:

```
differenza<-conoscenze$T1-conoscenze$T0
```

la cui media è **uguale alla differenza fra le medie di** X_1 **e** X_2, cioè il numeratore del t-test:

```
mean(differenza)
[1] 9.802198
mean(conoscenze$T1)-mean(conoscenze$T0)
[1] 9.802198
```

Quindi, possiamo sostituire $t_{N-1} = \frac{|\bar{x}_1 - \bar{x}_2|}{SE(|\bar{x}_1 - \bar{x}_2|)}$ con $t_{N-1} = \frac{\bar{X}_{differenze}}{SE_{differenze}}$. Sappiamo ricavare senza problemi lo SE di

una distribuzione, purché questa distribuzione **sia affine alla normale**: $SE_{differenze} = \sqrt{\frac{s^2_{differenze}}{N}}$

Quindi:

```
(ES<-sqrt(var(differenza)/91))
[1] 0.4819628
(t<-mean(differenza)/ES)
[1] 20.33808
pt(t, 90, lower.tail = FALSE)
[1] 1.023078e-35
```

Vediamo allora se è vero che $t_{N-1} = \frac{|\bar{x}_1 - \bar{x}_2|}{\sqrt{\frac{s_1^2 + s_2^2 - 2cov_{x1x2}}{N}}}$:

```
numeratore<-mean(conoscenze$T1)-mean(conoscenze$T0)
denominatore<-sqrt((var(conoscenze$T0) + var(conoscenze$T1) - 2*cov(conoscenze$T0, conoscenze$T1)) / 91)
(t<-numeratore/denominatore)
[1] 20.33808
```

Sì, è vero.

Naturalmente, d'ora in poi useremo la funzione **t.test(x1, x2, paired=TRUE)** senza porci altri problemi:

```
t.test(conoscenze$T1,conoscenze$T0, paired=TRUE)
        Paired t-test
data:  conoscenze$T0 and conoscenze$T1
t = 20.338, df = 90, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:   ← CI della media delle differenze; piccolo e non contiene H0: x̄ x1−x2 = 0
  8.844695  10.759701
sample estimates:
mean of the differences          ← media delle differenze tra i valori appaiati di x1 e x2
               9.802198
```

La media delle differenze nelle conoscenze tra T_1 e T_0 è = 9.8 nel campione (a T_1 le conoscenze sono aumentate), e in popolazione possiamo aspettarci una differenza reale compresa, con il 95% di verosimiglianza, tra 8.8 e 10.8: l'intervallo è piuttosto ristretto e non contiene il valore previsto da H_0; quindi, sembra verosimile che le due distribuzioni non appartengano alla medesima popolazione.

È vero che le due misure ripetute sono correlate come ci si aspetta?

```
cor(conoscenze$T0, conoscenze$T1)
0.722087
```

Altroché. Chi aveva più conoscenze a T_0 tende ad avere più conoscenze anche a T_1.

Non dimentichiamo di chiederci quanto sia **forte** questa differenza / **l'effetto** di X. Nel t-test per dati appaiati è piuttosto controverso se per calcolare lo SE_d sia meglio utilizzare la deviazione standard *pooled* (Rosenthal, 1991) o le $\boldsymbol{s}$ di X_1 e X_2 separatamente (Dunlop, Cortina, Vaslow e Burke, 1996). Poiché, però, questi ultimi dimostrano che la s_{pooled}, corretta per la correlazione tra le misure, determina una **sovrastima dell'effect** size tanto maggiore quanto più grande è la correlazione, allora è più prudente usare le $\boldsymbol{s}$ **delle distribuzioni.** Comunque, R, fa da solo: usiamo ancora `cohen.d(x1,x2)`, aggiungendo **`paired= TRUE`**, che apporta una correzione proporzionale al grado di correlazione tra le due misure (Borenstein et al., 2009):

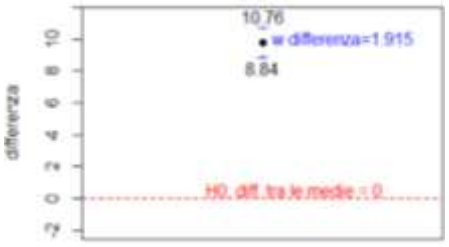
```
cohen.d(conoscenze$T0, conoscenze$T1, paired=TRUE)
Cohen's d
d estimate: -2.13201 (large)
95 percent confidence interval:
      correlazione       sup
-2.501109 -1.762912
```

L'effetto del corso (o solo del tempo? Mancano diversi elementi di controllo, come nel caso precedente) è di forte entità tanto nel campione quanto in popolazione.

Gestiremo correttamente la presenza del gruppo di controllo del dataframe `sicurezza` *nell'ANOVA fattoriale mista, ma possiamo fin d'ora anticipare quale sarà l'effetto sull'ipotesi che il corso di formazione abbia un reale effetto: verificate se e come cambino i soggetti del* ***gruppo di controllo*** *per i comportamenti a rischio e le conoscenze.*

Prima di concludere, occupiamoci del **grafico delle medie con i CI**. Concludendo il discorso sui CI nel caso dei campioni indipendenti, avevamo anticipato che usare i CI delle medie per fare inferenze sulla significatività della loro differenza è lecito nel caso di gruppi indipendenti, ma **non per campioni appaiati**, perché i due CI non catturano la covarianza tra le misure – che, come abbiamo visto, è un elemento importate del t-test. Quindi, il precedente grafico va benissimo per **descrivere** il campione nei due tempi, **ma** è **fuorviante** per **inferire** sulla significatività della differenza tra i tempi: a questo scopo dovremo, invece, **plottare il CI della differenza tra le medie**.

```
MeanCI(differenza)
     mean    lwr.ci    upr.ci
 9.802198  8.844695 10.759701
wd<-abs(10.759701- 8.844695)
wd
[1] 1.915006
```



Il margine di errore della differenza tra le medie, w_{2_d}, **è sensibile alla correlazione tra le misure**, dato che l'ampiezza del CI della differenza (al quadrato) è dato dalla somma delle ampiezze dei CI delle medie di ogni misura ripetuta (al quadrato), meno due volte la correlazione tra le due misure:

$$\boldsymbol{w_d^2 = w_A^2 + w_B^2 - 2rw_Aw_B}$$

Sappiamo che H_0, nel caso di misure ripetute, prevede che nel $95\%CI$ della media delle differenze sia compreso 0, e che un intervallo più ampio ha maggiori probabilità di intercettare il valore 0. Se la correlazione tra le misure è $r = 0.0$, dalla somma delle ampiezze al quadrato dei CI di A e B si sottrae 0: $-2 \times 0 \times w_A \times w_B = 0$; quindi l'ampiezza del CI della media delle differenze è direttamente ricavabile dall'ampiezza dei due CI nel plot delle medie. Nel caso di covarianza diversa da zero, questa inferenza diretta è impossibile: tanto più le misure sono positivamente correlate, tanto più **piccolo** è il CI della media delle differenze, perché da $w_{2_A}^2 + w_{2_B}^2$ viene **sottratta** una quantità rilevante, fino a due volte il prodotto di $w_{2_A} \times w_{2_B}$: $-2 \times 1 \times w_{2_A} \times w_{2_B} = -2 \times w_{2_A} \times w_{2_B}$; è quindi **più probabile** che la differenza $X_A - X_B$ risulti **significativa**. Tanto più le misure sono negativamente correlate, tanto più **grande** è il CI della media delle differenze, perché a $w_{2_A}^2 + w_{2_B}^2$ viene **aggiunta** una quantità rilevante, fino a due volte il prodotto di $w_{2_A} \times w_{2_B}$: $-2 \times -1 \times w_{2_A} \times w_{2_B} = +2 \times w_{2_A} \times w_{2_B}$. È quindi **meno probabile** che la differenza tra A e B risulti **significativa**.

Vediamo un esempio: quattro misure ripetute (A, B, C, D), in cui calcoliamo le differenze A-B, A-C, A-D:

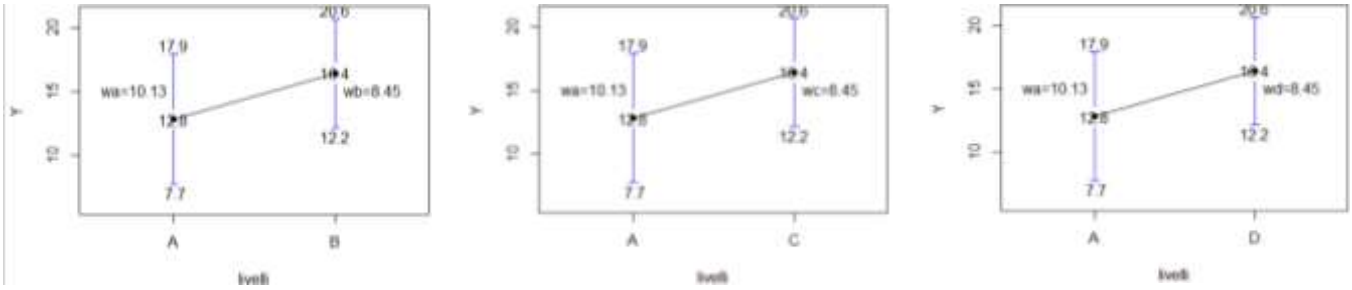
```
A<-c(2,4,8,10,11,14,17,18,20,24)
B<-c(7,8,12,15,16,19,22,19,24,22)
C<-c(22,24,19,22,19,16,15,12,8,7)
D<-c(8,19,22,24,7,22,16,19,12,15)
```

Osserviamo le correlazioni tra A e le altre misure con cui sarà confrontata:

```
ABCD<-matrix(c(A,B,C,D), nrow = 10, ncol = 4); colnames(ABCD)<-c("A", "B", "C", "D")
round(cor(ABCD),3
       A      B      C     D
A  1.000  0.952 -0.943 0.021
B  0.952  1.000 -0.870 0.011
C -0.943 -0.870  1.000 0.135
D  0.021  0.011  0.135 1.000
```

A e B hanno una forte correlazione positiva, A e C una forte correlazione negativa, A e D sono indipendenti.

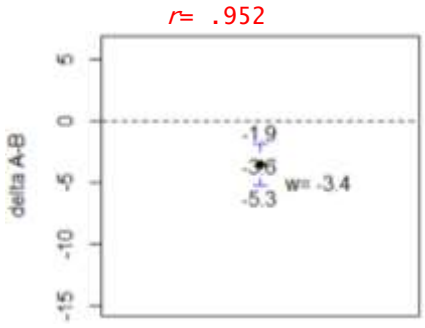
Rappresentiamo i tre confronti tra medie nel plotmeans con i CI, che non dà informazioni sulla loro covarianza.



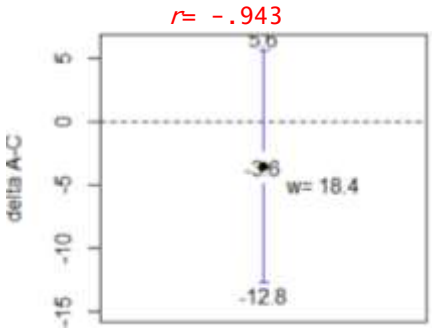
I CI delle medie di B, C e D sono identici: quindi, le conclusioni sulla significatività della differenza che trarremmo dal *proportional overlap* tra A e una qualsiasi altra misura dovrebbero essere identiche per tutti i confronti:

```
MeanCI(A); MeanCI(B); MeanCI(C); MeanCI(D)
    mean    lwr.ci   upr.ci
12.80000  7.73268 17.86732 → (wa<-17.86-7.73)  → 10.13
    mean    lwr.ci   upr.ci
16.40000 12.17192 20.62808 → (wb<-20.62-12.17) → 8.45
    mean    lwr.ci   upr.ci
16.40000 12.17192 20.62808 → (wc<-20.62-12.17) → 8.45
    mean    lwr.ci   upr.ci
16.40000 12.17192 20.62808 → (wd<-20.62-12.17) → 8.45
(wm<-(10.13+8.45)/2); (overlap<-17.87-12.17)
[1] 9.29
[1] 5.7
(proportional_overlap<-5.7/9.29)
[1] 0.613563
```

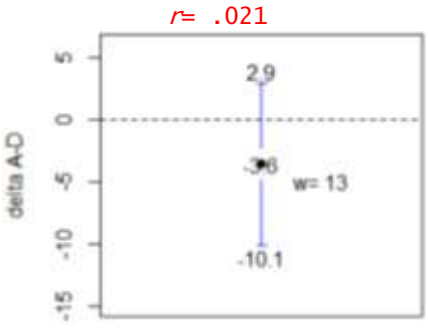
Il *proportional overlap* è $> .50$: le differenze $A - B$, $A - C$, $A - D$ non sembrano significative per $\alpha \leq .05$. Vediamo, però, cosa succede **rappresentando il CI della media delle differenze** tra le misure nei tre confronti e ricordando la correlazione tra le variabili in analisi:



La differenza tra le medie **è significativa**



La differenza tra le medie **non è significativa**



La differenza tra le medie **non è significativa**

A parità di differenza media tra le misure, quando si prende in considerazione anche l'informazione sulla covarianza l'inferenza sulla significativa della differenza cambia. Infatti, dato che $w^2_{2_{delta}} = w^2_{2_A} + w^2_{2_B} - 2rw_{2_A}w_{2_B}$:

```
sqrt(wa^2+wb^2-(2*.952*wa*wb))
[1] 3.386042                          ← w2_AB
sqrt(wa^2+wc^2-(2*(-.943)*wa*wc))
[1] 18.33555                          ← w2_AC
sqrt(wa^2+wd^2-(2*.021*wa*wd))
[1] 13.0                              ← w2_AD
```

Rispetto al $w_{2_{AD}}$, che vede le misure come indipendenti e non significativamente differenti, il $w_{2_{AC}}$, tra misure con una forte correlazione **negativa**, è più grande e abbraccia con più decisione il valore previsto da H_0, mentre $w_{2_{AB}}$, tra misure con una forte correlazione positiva, rimpicciolisce molto, tanto che il $CI_{delta_{AB}}$ non contiene più il valore previsto da H_0. Il plot dei CI delle medie ci avrebbe tratto in inganno.

10.2.3 Test non parametrici o robusti

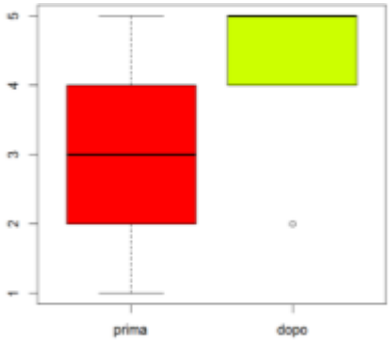
ANOVA a misure ripetute con X a due livelli e t-test per dati appaiati richiedono che siano soddisfatti i requisiti dei modelli lineari; in caso di violazione, o se Y è almeno ordinale, possiamo nuovamente decidere se **trasformare non linearmente la distribuzione Y** (ammesso che funzioni) o **sostituire** i test parametrici con i test **non parametrici** o **robusti**. Vedremo il **test di Wilcoxon per dati appaiati** e altre **statistiche robuste di Wilcox**.

Il test di **Wilcoxon** (**Wilcoxon** e **Wilcox**, 1964) **per dati appaiati** considera le differenze tra due condizioni X_1 eX_2, ma utilizzando i ranghi delle differenze tra le condizioni, considerati nel loro valore assoluto; conserva praticamente tutta la potenza del t-test per dati appaiati (.955, quando la distribuzione è normale).
Ogni coppia di osservazioni dà luogo a una **differenza**: le **differenze = 0 sono escluse dai successivi calcoli**. A questa **distribuzione di differenze**, in **valore assoluto**, è **assegnato un rango** (o un rango **medio**, in caso di *ties*); successivamente, sono **sommati separatamente** i **ranghi** assegnati alle differenze con **segno +** e quelli assegnati alle **differenze** con **segno -** . Il test, perciò, in primo luogo tiene conto solo dell'intensità della differenza, e in secondo luogo considera anche la direzione della differenza. Secondo H_0, le differenze con segno positivo ($x_{1i} > x_{2i}$) equivalgono a quelle con segno negativo.

- H_0: $mediana_{X1} = mediana_{X2}$
- H_1: $mediana_{X1} \neq mediana_{X2}$ oppure $mediana_{X1} > mediana_{X2}$ oppure $mediana_{X1} < mediana_{X2}$

Il test si applica perlopiù a piccoli campioni, come molti test non parametrici. Tuttavia, esistono alcuni vincoli: con una soglia $\alpha = .\mathbf{05}$, se H_1 è monodirezionale per rifiutare H_0 servono almeno cinque coppie di osservazioni, in esse non si deve registrare alcuna $differenza = 0$ e si rifiuta H_{01} solo se le differenze hanno tutte lo stesso segno. Se H_1 è bidirezionale, per rifiutare H_0 servono almeno sei coppie di dati e tutte le differenze devono avere la medesima direzione. Vediamo i passaggi del test usando dati inventati. Per qualche anno ha conosciuto parecchia popolarità, anche se non è stato successivamente confermato, il cosiddetto "Effetto Mozart": ascoltare musica di Mozart comporterebbe un incremento, anche a lungo termine, delle abilità cognitive, soprattutto visuo-spaziali. Per puro spirito di contraddizione, verifichiamo se anche **l'ascolto di musica gothic metal** (X; sceglieremo i Lacrimosa) produce un effetto sulla **creatività (Y)**, operazionalizzata come **numero di soluzioni alternative** prodotte al test di Christensen. Scegliamo 10 volontari, completamente a digiuno di buona musica, e li sottoponiamo alle forme parallele del test un'ora prima (X_1) e un'ora **dopo** (X_2) l'ascolto dell'intera discografia del duo.

```
prima<-c(3,4,3,1,5,2,3,4,1,3)
dopo<-c(5,5,2,5,4,5,5,4,5,5)
median(prima);median(dopo)
[1] 3
[1] 5
mean(prima);mean(dopo)
[1] 2.9
[1] 4.5
sd(prima);sd(dopo)
[1] 1.286684
[1] 0.971825
```



Il numero di risposte creative aumenta dopo l'ascolto, nel senso previsto dall'ipotesi alternativa ($med_{x1} < med_{x2}$); la distribuzione, prima decisamente più variabile da una sostanziale ottusità a una buona creatività, dopo l'ascolto si compatta, tranne un caso irrecuperabile. Calcoliamo le differenze per ogni soggetto:

```
(differenze<-prima-dopo)
[1] -2 -1  1 -4  1 -3 -2  0 -4 -2
```

Le differenze negative sono la gran maggioranza, e ci sono diversi ties, cui sarà assegnato il rango medio. Dobbiamo comunque **escludere il soggetto che non è cambiato**, quello con $\boldsymbol{differenza = 0}$:

```
(differenze<-differenze[differenze!=0])
[1] -2 -1  1 -4  1 -3 -2 -4 -2
```

Ora **assegniamo i ranghi** alle **differenze** in **valore assoluto**:

```
ranghi_differenze<-rank(abs(differenze))
ranghi_differenze
[1] 5.0 2.0 2.0 8.5 2.0 7.0 5.0 8.5 5.0
```

Separiamo i ranghi assegnati alle differenze negative da quelli assegnati alle differenze positive. Usiamo la funzione **`sign(x)`**, con restituisce un vettore che contiene i segni degli elementi della distribuzione:

```
sign(differenze)
[1] -1 -1  1 -1  1 -1 -1 -1 -1
```

Moltiplicando il vettore dei ranghi per quello dei segni delle differenze, otteniamo il **vettore dei ranghi con segno** (**signed ranks**):

```
(ranghi_differenze_segno<-ranghi_differenze*sign(differenze))
[1] -5.0 -2.0  2.0 -8.5  2.0 -7.0 -5.0 -8.5 -5.0
```

Ora possiamo **sommare separatamente i ranghi** delle differenze positive ($> \mathbf{0}$) e negative ($< \mathbf{0}$):

```
ranghi_positivi<-sum(ranghi_differenze_segno[ranghi_differenze_segno>0])
ranghi_negativi<-abs(sum(ranghi_differenze_segno[ranghi_differenze_segno<0]))
ranghi_positivi; ranghi_negativi
[1] 4
[1] 41
```

Come nel test di Wilcoxon / Mann – Whitney, è la **sommatoria più piccola** tra le due ($\Sigma_+ = 4$) a essere confrontata con i valori critici tabulati per il test, al fine di assegnare il $p - value$. La statistica del test (V) segue una distribuzione approssimativamente normale con grandi campioni, cioè indicativamente con $N > 25$, ma l'approssimazione è già buona per 14-15 coppie di dati.

Con R torniamo a usare `wilcox.test(x1,x2, paired=TRUE)`:

```
wilcox.test(prima, dopo, paired = TRUE)
        Wilcoxon signed rank test with continuity correction
data:  prima and dopo
V = 4, p-value = 0.0316
alternative hypothesis: true location shift is not equal to 0

Warning messages:
1: In wilcox.test.default(prima, dopo, paired = TRUE) :
impossibile calcolare p-value esatto in presenza di ties
2: In wilcox.test.default(prima, dopo, paired = TRUE) :
  impossibile calcolare p-valu esatti in presenza di zeri
```

Il calcolo del CI esatto per la statistica V (`conf.int=TRUE`) richiede che siano eliminati tutti i casi che producono una $differenza = 0$ (in caso contrario ne viene prodotta una stima inesatta). Nel nostro esempio, dobbiamo eliminare il soggetto 8:

```
prima<-prima[-8]
dopo<-dopo[-8]
wilcox.test(prima, dopo, paired= TRUE, conf.int=TRUE)
        Wilcoxon signed rank test with continuity correction
data:  prima and dopo
V = 4, p-value = 0.0316
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -3.000037 -0.499940
sample estimates:
(pseudo)median
     -1.999976
```

Come nel caso per campioni indipendenti:

```
hodges_lehmann<-outer(prima,dopo,"-")
median(hodges_lehmann)
[1] -2
```

Per la stima dell'entità dell'effetto da associare al test di Wilcoxon per dati appaiati, possiamo ricorrere al **coefficiente di correlazione biseriale applicato ai ranghi** delle due misure (il coefficiente delta di Cliff, ricordiamo, si applica invece a gruppi) con `rank_biserial(x= misura1, y=misura2, paired= TRUE)` di `effectsize`. Varia da -1 a +1: il segno indica se tendono a essere più numerosi i segni positivi ($X_1 > X_2$) o i segni negativi ($X_1 > X_2$). Nell'esempio dei Lacrimosa, in cui i punteggi dopo l'ascolto aumentano, dovremmo attenderci un coefficiente negativo:

```
rank_biserial(prima, dopo, paired = TRUE)
r (rank biserial) |         95% CI
---------------------------------
-0.82             | [-0.95, -0.43]
```

In effetti, la differenza è intensa e nella direzione prevista.

Tra le **misure robuste di Wilcox** del package **WRS2**, l'analogo robusto di ANOVA a misure ripetute è `rmanova(y, x, blocks= variabile che identica il soggetto)`, che si applica su **medie *trimmed*** con dataframe in ***long format***; la ritroveremo anche nel Capitolo 12, applicata a X a misure ripetute con $k > 2$ livelli.

```
rmanova(y=melt_conosc$conoscenze, melt_conosc$tempo, blocks=melt_conosc$sogg, trim=.2)
Call:
rmanova(y=melt_conosc$conoscenze,melt_conosc$tempo,blocks=melt_conosc$sogg, trim=.2)

Test statistic: 182.8665
Degree of freedom 1: 1.94
Degree of freedom 2: 104.54
p-value: 0
```

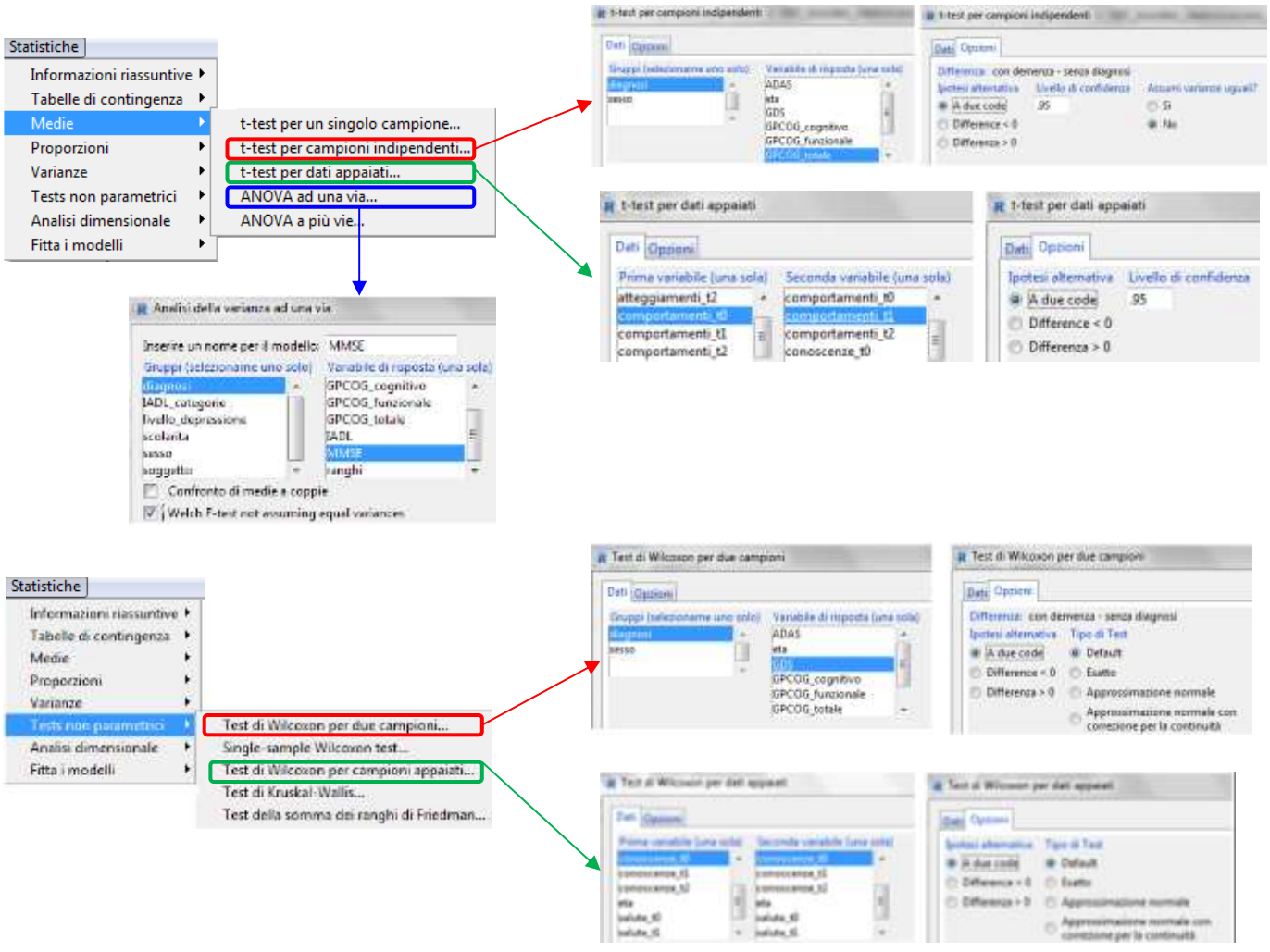
Ribadiamo che le conoscenze del corso sulla sicurezza aumentano in maniera significativa.

Invece, l'analogo del t-test per dati appaiati è il **test di Yuen per dati appaiati**, per medie tronche: `yuend(x1, x2, trim= proporzione di dati tronchi a ogni coda)`:

```
yuend (s$comportamenti_t0, s$comportamenti_t1, trim=.2)
Call:
yuend (x= comportamenti_t0, s=s$comportamenti_t1, trim=.2)

Test statistic: 9.0202 (df = 54), p-value = 0
Trimmed mean difference: 1.01273
95 percent confidence interval:
.7876        1.2378
```

Di nuovo, i comportamenti a rischio diminuiscono in maniera significativa dopo la frequenza del corso.

In `Rcommander`, i test parametrici per una X a due livelli sono nel menu Statistiche → Medie, quelli non parametrici nel menu Statistiche → test non parametrici:



Capitolo 11
Regressione lineare multipla

In questi paragrafi useremo il dataframe ***attaccamento*** *pubblicato su Elly e già noto: apritelo e ripassatene la descrizione*

Quando nel modello lineare inseriamo più predittori X, valutiamo l'effetto congiunto dei predittori su una sola Y: in questo capitolo trattiamo il caso di predittori continui, nei prossimi di predittori categoriali, ma solo per semplificare l'esposizione: in effetti, si possono inserire in un solo modello lineare predittori continui e categoriali.

Come nella regressione semplice, $\boldsymbol{H_0}$ è che in popolazione Y sia **indipendente da tutte le X inserite nel modello** → tra le X e Y **non esiste** una relazione di predittività: conoscendo il valore $x_{1i}, x_{2i}, \dots x_{ki}$, **non** possiamo predire quale sarà il corrispettivo valore y_i; $\boldsymbol{H_1}$ è che in popolazione le X e Y siano **dipendenti** → tra le X e Y esiste una **relazione di predittività** → al variare dei predittori, il criterio varia in maniera predicibile: **conoscendo i valori $x_{1i}, x_{2i}, \dots x_{ki}$, possiamo predire quale sarà il corrispettivo valore y_i, con un margine di errore minimo (il più piccolo possibile).**

Poiché passiamo a uno **spazio almeno tridimensionale**, la retta di regressione non sarà più il nostro modello grafico: parleremo invece di un **piano di regressione.**

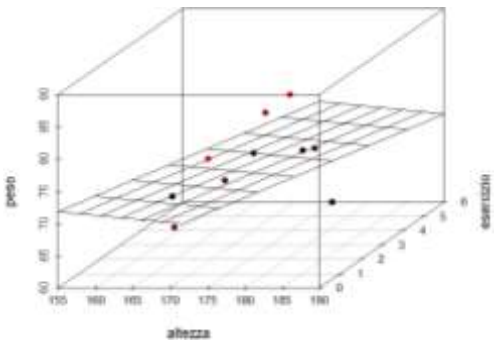
Possiamo visualizzare la dispersione multidimensionale installando `scatterplot3d` e usando la funzione **`scatterplot3d(y=Y,x=X1,z=X2)`**. L'oggetto creato dalla funzione `scatterplot3d` contiene un elemento, `grafico3d$plane3d`, che consente di aggiungere un piano tridimensionale al grafico a dispersione tridimensionale. In `$plane3d` va specificato il nome del modello di regressione multipla e l'argomento `lty="solid"`: `grafico3d$plane3d(modello, lty= "solid")`. Quando il modello prevede solo tre variabili, una Y e due X, la lettura del piano tridimensionale è piuttosto semplice, ma con più variabili è meglio lasciar perdere.

Per esempio, questa è la rappresentazione grafica di **peso ~altezza + ore di esercizio fisico a settimana** che abbiamo costruito raccogliendo in **altri 10 soggetti** rispetto a quelli usati nel Capitolo 3**,** per verificare la relazione tra peso (Y), altezza (X_1) ed esercizio fisico (X_2):

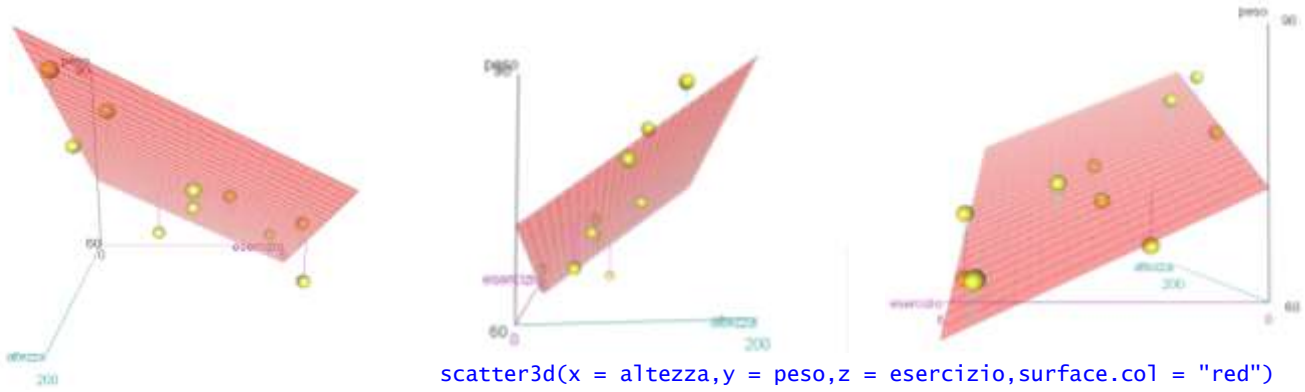
```
peso<-c(80,68,72,75,70,65,62,60,85,90)
altezza<-c(175, 171, 170, 181, 169, 165, 155, 175, 180, 186)
esercizio<-c(0,6,4,3,3,2,5.5,6,1,0)
modello2<-lm(peso~altezza + esercizio)
```

Costruiamo il grafico e aggiungiamo il piano di regressione:

```
grafico2<-scatterplot3d(x = altezza,y = esercizio,z =
  peso,pch=19, highlight.3d = TRUE)
grafico2$plane3d(modello2, lty="solid")
```



La funzione **`scatter3d(y=Y,x=X1,z=X2)`** di `car` usa anche le funzionalità del package `rgl` per creare piani di regressione interattivi: il grafico viene prodotto in una nuova finestra `RGL`, e può essere orientato nello spazio cliccandovi sopra con il mouse e spostandolo per perfezionarne la leggibilità.



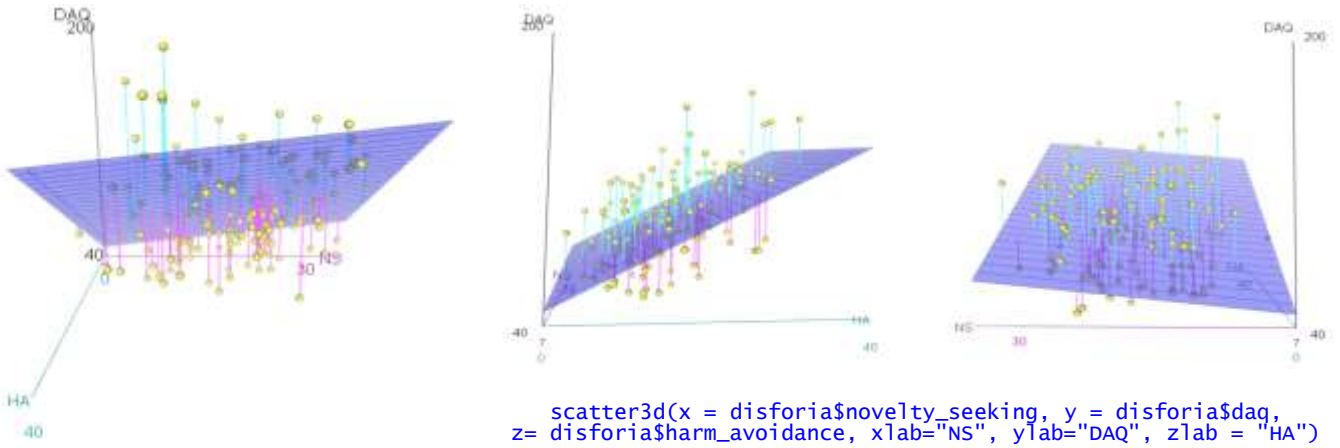
```
scatter3d(x = altezza,y = peso,z = esercizio,surface.col = "red")
```

Lo *slope* della relazione tra altezza e peso sembra positivo, mentre quello della relazione tra peso ed esercizio sembra negativo (attenti alla direzione dell'asse Z). Possiamo anticipare i due coefficienti angolari, solo per verificarne il segno:

```
modello2$coefficients[2:3]
```

```
   altezza  esercizio
 0.4884444 -2.5312032
```

Un altro esempio: vediamo, nel dataframe `disforia`, la relazione tra Y = disforia e X= Novelty Seeking più Z= Harm Avoidance.




```
scatter3d(x = disforia$novelty_seeking, y = disforia$daq,
z= disforia$harm_avoidance, xlab="NS", ylab="DAQ", zlab = "HA")
```

Entrambi i coefficienti angolari sembrano positivi, ma lo slope della NS è decisamente inferiore:

```
ns_ha<-lm(disforia$daq~disforia$novelty_seeking+disforia$harm_avoidance)
ns_ha$coefficients[2:3]
```

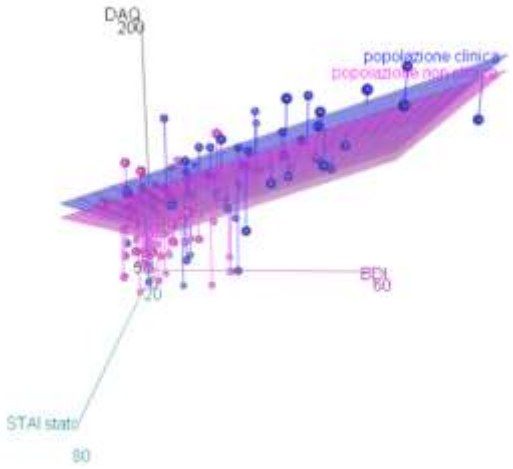
```
disforia$novelty_seeking  disforia$harm_avoidance
               0.7090589                2.3900588
```

Infatti.

La funzione consente anche di mostrare piani di regressione / modelli in subset del dataframe, opzione molto utile per verificare se la relazione tra Y e i predittori è la medesima in tutti i livelli di un fattore: basta aggiungere l'argomento `groups= fattore di raggruppamento`.

Verifichiamo se la relazione tra Y = disforia e X= BDI + Z= STAI stato è la medesima nella popolazione clinica e in quella non clinica:

```
scatter3d(x = disforia$novelty_seeking,y = disforia$daq,
z=disforia$harm_avoidance,  groups  =  disforia$gruppo,
xlab="BDI", ylab="DAQ", zlab = "STAI stato")


lm(disforia$daq~disforia$BDI+disforia$STAI_stato,  subset
= disforia$gruppo=="popolazione clinica")
Coefficients:
        (Intercept)      disforia$BDI  disforia$STAI_stato
            63.8695            1.1197                0.7586
lm(disforia$daq~disforia$BDI+disforia$STAI_stato,  subset
= disforia$gruppo=="popolazione non clinica")
Coefficients:
        (Intercept)      disforia$BDI  disforia$STAI_stato
            46.5195            0.6496                1.1555
```



Sì, la relazione sembra decisamente simile almeno rispetto alla direzione, anche se nella popolazione clinica sulla disforia sembra pesare di più la componente depressiva, mentre in quella non clinica sembra più incisiva la componente d'ansia aspecifica. Inoltre, per una depressione e un'ansia pari a 0 (intercetta), la disforia della popolazione clinica è più alta.

Matematicamente parlando, il modello lineare della relazione tra i tre vettori è solo un più complesso di quello con un solo predittore, ma di facile comprensione:

$$\boldsymbol{dato\ reale_{y_i} = (modello\ lineare) + e_i}$$
$$\downarrow$$
$$\boldsymbol{y_i = (\beta_0 + \beta_1 X_1 + \beta_2 X_2) + e_i}$$

Il dato osservato Y per il caso i (y_i) è predetto dal punto in cui il piano di regressione intercetta Y (β_0) più l'effetto del predittore X_1 (β_1) per il valore in X_1 di i, più l'effetto del predittore X_2 (β_2) per il valore in X_2 di i, più la quota **di errore** e_i **commessa dal modello per** $\boldsymbol{i}$**.**

Questo modello lineare è di tipo **additivo**: sono verificati **solo gli effetti principali** di ogni X, ovvero l'effetto di ogni X tenendo costanti le altre → **effetto di ogni** $\boldsymbol{X}$ **indipendentemente dall'effetto delle altre** (l'effetto dell'altezza indipendentemente dal numero di ore di esercizio; l'effetto dell'esercizio indipendentemente dall'altezza). Più complessi sono i **modelli lineari con interazioni**, in cui si valuta l'**effetto di** X_1 **su** $\boldsymbol{Y}$ **indipendentemente da** $\boldsymbol{X_2}$ **(effetto di moderazione)**. Nell'esempio di peso, altezza ed esercizio, valuteremmo l'effetto dell'altezza sul peso a seconda della quantità di ore di esercizio, oppure, a seconda dell'ipotesi, l'effetto della quantità di esercizio sul peso a seconda dell'altezza del soggetto: il peso potrebbe aumentare in funzione dell'altezza, ma solo per chi fa poco esercizio; oppure, fare più esercizio diminuisce il peso, ma solo se si è bassi.

Il modello con interazione diventa:

$$\boldsymbol{y_i = (\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2) + e_i}$$

Ci concentreremo per ora soprattutto sui modelli additivi, più semplici, facendo qualche esempio dei modelli con interazione, che approfondiremo nel Capitolo 13 con X categoriali.

Come nella regressione semplice, la SS_M esprime la distanza complessiva tra i **valori predetti dal modello e la media di Y**, la SS_R rappresenta la distanza complessiva tra i **valori predetti dal modello e gli** $\boldsymbol{Y}$ **valori effettivamente osservati**, e la SS_T, somma delle precedenti, rappresenta la distanza complessiva gli Y osservati e la media di Y. SS_M e SS_R sono divise per df_M e df_R ottenendo MS_M e MS_R, che danno il rapporto $\boldsymbol{F}$. Con più X, il calcolo delle tre SS è più complesso di quello eseguito nella regressione semplice, ma, poiché R lo farà per noi, non ce ne dovremo preoccupare. La **quantità di variabilità di** $\boldsymbol{Y}$ **spiegata da tutti i predittori nel modello** è ancora quantificata da $\boldsymbol{R^2_M}$ (il rapporto tra SS_M e SS_T), ora definito $\boldsymbol{R^2}$ **multiplo** (così è chiamato in tutti gli output di `lm`, in effetti): il suo valore non coinciderà più con il coefficiente di determinazione bivariato, e resterà il **quadrato del coefficiente di correlazione tra i valori** $\boldsymbol{Y}$ **e i**

valori Y predetti dal modello come nella regressione semplice. Possiamo facilmente verificarlo: nel modello di regressione semplice `peso~altezza` avremo:

```
modello1<-lm(peso~altezza)
cor(peso, altezza)^2
[1] 0.5737895
cor(peso, predict(modello1))^2
[1] 0.5737895
summary(modello1)
[omissis]
Multiple R-squared:  0.5738,     Adjusted R-squared:  0.5205
F-statistic: 10.77 on 1 and 8 DF,  p-value: 0.01116
```

Invece, nel modello con due predittori `peso~altezza+esercizio` troviamo:

```
cor(peso,predict(modello2))^2
[1] 0.8258365
summary(modello2)
[omissis]
Multiple R-squared:  0.8258,     Adjusted R-squared:  0.7761
F-statistic:  16.6 on 2 and 7 DF,  p-value: 0.002205
```

11.1 Model selection (model specification)

Avendo a disposizione diversi predittori di Y, che ci si augura inseriti nella ricerca in base a ipotesi discendenti da una teoria piuttosto precisa sulla rete di relazioni che lega dipendente e predittori in popolazione, il processo di **model selection** (o ***model specification***, o *model* building...**)** consiste nell'individuazione della combinazione di predittori che meglio avvicina il modello osservato nei dati al cosiddetto **modello generatore dei dati**. Il modello generatore dei dati è quello **che descrive le relazioni tra Y e le X in popolazione**, e che dovrebbe essere rispecchiato nel modello osservato nel campione.

Diciamo che abbiamo una Y e quattro X, con cui possiamo costruire diversi modelli di predizione: $Y{\sim}X_1$, $Y{\sim}X_2$, $Y{\sim}X_3$, $Y{\sim}X_4$, $Y{\sim}X_1+X_2$, $Y{\sim}X_1+X_3$, $Y{\sim}X_1+X_2+X_3$, $Y{\sim}X_1\times X_2$, eccetera eccetera. La model selection cerca di individuare, tra tutti i possibili modelli ottenibili con i predittori a disposizione, il modello osservato che più si avvicina al modello generatore dei dati. Avremo quindi un **set di modelli alternativi** (**model class** o **model set**), composti da diversi predittori in diverse combinazioni, che **competono per essere selezionati** come **migliore** approssimazione al modello generatore dei dati, **per uno specifico insieme di dati e all'interno del set di modelli**. Attenzione, quindi: la model selection **non** individua il miglior modello **in assoluto**: se il modello generatore non è presente nella model class (per esempio, nella ricerca è stato ignorato l'eccellente predittore X_5), o se i soggetti sono inadeguati (non rappresentativi, pochi, con casi influenti non corretti...), la model selection sceglierà il meno peggiore tra i modelli possibili, non il migliore possibile. Per questo motivo, il processo deve essere sorretto in prima istanza da una forte base teorica, più che meramente statistica.

In linea generale, la selezione del modello di regressione ottimale deve essere guidata da due principi: la **parsimonia** (*parsimony*) richiede che il modello sia il più semplice possibile, cioè che **non contenga b_1 ridondanti** (predittori non esplicativi), e deve essere bilanciata con **l'adeguatezza**, ovvero con la capacità del modello di descrivere **un'adeguata porzione di variabilità di Y**. Quindi, un **modello realmente efficiente è quello che spiega** il **massimo della variabilità di Y con il minimo numero di predittori necessari**; in omaggio alla parsimonia, inoltre, ove possibile si preferiranno relazioni lineari tra Y e X a modelli non lineari, e spiegazioni semplici a spiegazioni complesse. Un modello ben specificato (**well specified model**) include tutte le X che agiscono su Y (ma questa condizione è praticamente irrealizzabile nella realtà) ed esclude tutte le X che non hanno una relazione significativa con Y. Dato che è sostanzialmente irrealistico ottenere un modello perfettamente stimato, i nostri modelli possono essere più o meno

under fit / **sotto-specificati** (sono stati inserite meno X di quelle necessarie), oppure **over fit** / **sovra-specificati** (sono state inserite più X di quelle necessarie).

Il modo più "grezzo" di costruire un modello è **per blocchi**: tutti i **predittori sono inseriti simultaneamente nel modello**, rinunciando a definirne l'ordine d'importanza corrispondente a una specifica ipotesi sulla capacità predittiva dei costrutti che rappresentano. Proprio per questo motivo, è considerato un metodo piuttosto primitivo (e raramente ottimale), ma, d'altro canto, è anche molto solido e fornisce con più probabilità risultati replicabili, se il modello viene testato su altri campioni.

Nell'impossibilità di raggiungere il modello perfetto, è decisamente più opportuno **confrontare modelli riferiti alla stessa Y e costituiti da un diverso numero di fattori**, **penalizzando quelli in cui un maggior numero di X non determina un efficace incremento d'informazione**. Secondo questa logica, in linea generale, un predittore viene mantenuto nel modello solo se la sua rimozione dal modello provoca un **significativo decremento nella devianza spiegata**.

Occorre, quindi, trovare un quantificatore di fit adeguato alla model selection.

Nella regressione semplice, abbiamo usato R^2 per quantificare la devianza spiegata dal predittore; nella regressione multipla, però, R^2_M pone un problema **quando si confrontano i fit di modelli riferiti a una stessa Y, ma composti da un diverso numero di X:** più predittori aggiungiamo al modello, più la variabilità di Y spiegata dal modello aumenta, anche se di poco, dato che ciascuno aggiunge la sua piccola pietruzza alla quantità complessiva. Saremmo tentati di definire come modello migliore quello che, in riferimento a una stessa Y e con un diverso numero di X, presenta il maggior R^2_M, cioè spiega la maggior quantità di variabilità di Y, ma questo va contro il principio di parsimonia, che privilegia l'economia nei predittori bilanciata per variabilità spiegata. Notate che, a differenza di R^2_M, $\boldsymbol{R^2_{adj}}$ **aumenta solo quando il nuovo predittore incrementa il fit in maniera significativa**: le X che aggiungono più errore che spiegazione non fanno crescere R^2_{adj}, anzi, ne aumentano la differenza con l'R^2_M del modello (l'avevamo anticipato nella regressione semplice).

Indicatori di fit più adeguati alla model selection sono i cosiddetti ***information criteria***, una numerosa **famiglia di indicatori di fit** nati all'interno del metodo della **massima verosimiglianza** (*Maximum Likelihood* o $\boldsymbol{ML}$: stima parametri di modelli lineari e non lineari con un approccio diverso dal metodo dei minimi quadrati – *Ordinary Least Squares* o $\boldsymbol{OLS}$). Affronteremo più in dettaglio la ML nei capitoli 14 e 15, mentre in questo capitolo vedremo gli *information criteria* adattati al caso della regressione lineare secondo OLS. Tra i molti *information criteria* a disposizione, i più diffusamente utilizzati sono probabilmente l'**Akaike Information Criterion** ($\boldsymbol{AIC}$; Akaike, 1974) e il **Bayesian Information Criterion** ($\boldsymbol{BIC}$; Schwarz, 1978), che è più prudente nella stima della variazione del fit e va **preferito ad $\boldsymbol{AIC}$ quando i parametri nel modello sono molti**. La logica che riflettono è la stessa: esprimono la **quantità di errore nel modello** (perciò, più piccoli sono, migliore è il modello) **penalizzata per il numero di parametri** (predittori) del modello. Diciamo che il modello A ($Y \sim X_1 + X_2$) e il modello B ($Y \sim X_1 + X_2 + X_3$) hanno la stessa quantità di SS_R: al modello B sarà **aggiunto un handicap maggiore**, perché ha tre predittori contro i due di A, e quindi l'information criterion di B sarà più grande dell'information di A. Preferiremo, quindi, il modello A, che (non) spiega quanto B, ma con meno predittori.

$$AIC = N \times \log\left(\frac{SS_R}{N}\right) + 2k \qquad\qquad BIC = N \times \log\left(\frac{SS_R}{N}\right) + 2k \times \log(N)$$

N = numero di osservazioni, log= logaritmo in base naturale; k= numero di parametri del modello.

Un modello con tre X ottiene un $R^2_M = .455$; Aggiungiamo una quarta X: questa dovrebbe ridurre SS_R, aumentando il fit, ma se in realtà X_4 non aggiungesse ***nulla*** *al fit, la prima parte dell'equazione (in blu) non cambierebbe, mentre la seconda (in rosso) sì: ne risulterà che l'AIC del modello a 4 predittori* ***sarebbe maggiore dell'AIC del modello a 3*** *predittori,* ***a parità di R^2_M****. La parte penalizzante di BIC (in rosso) aumenta ulteriormente la penalizzazione rispetto a quella di AIC, nella medesima direzione*.

Useremo anche l'**Akaike Information Criterion Corrected** (***AICC***, *finite-sample AIC corrected*; Hurvich e Tsai, 1989); la formula si riferisce a modelli lineari univariati (una Y) con errori normalmente distribuiti.

$$AIC_C = N \times \log\left(\frac{SS_R}{N}\right) + 2\frac{k \times (k-1)}{N-k-1}$$

È sostanzialmente equivalente ad AIC per grandi N e da preferire per campioni piccoli, definibili come quei campioni in cui il rapporto tra N e k - parametri del modello è < 40, purché le distribuzioni non siano troppo leptocurtiche.

La **parte in blu** di AIC, BIC e $AICc$ è la $-2LL$ ($-2\,log-likelihood$) o **devianza** del modello costruito secondo il metodo della ML, che corrisponde interpretativamente alla SS_R nel metodo OLS: la ritroveremo nel capitolo 14.
Approfondiremo questi indici nei capitoli 14 e 15: per ora può essere sufficiente ricordate che, per tutti gli *information criteria*, un **minor *AIC* / *BIC* / *AICc*** corrispondono a un **miglior fit** (la devianza di errore, quella residua, diminuisce), **corretto per il numero dei parametri inseriti nel modello**. Inoltre, non esiste una soglia di "buon fit" o "cattivo fit" assoluto: sono **indici di fit differenziali** (o **incrementali**), dotati di senso interpretativo solo nel confronto fra modelli.
La model selection può quindi usare gli *information criteria* come piuttosto efficienti quantificatori di fit nella scelta del modello ottimale; sottolineeremo successivamente i loro - non trascurabili – difetti.
In R, AIC e BIC sono riportati di default negli output di diverse analisi; possono comunque essere richiesti con `AIC(modello)` e `BIC(modello)`.
Vediamo un esempio di valutazione del **fit di un modello costruito a blocchi**, cogliendo l'occasione di commentare e rappresentare un modello con **interazione**: confrontiamo il fit del modello additivo `peso~altezza+esercizio` e del modello con interazione `peso~altezza*esercizio`, in cui attribuiamo il ruolo di moderatore della relazione `peso~altezza` all'esercizio. Costruiamo i due modelli:

```
additivo<-lm(peso~altezza+esercizio)
interazione<-lm(peso~altezza*esercizio)
```

Confrontiamo prima i due summary:

```
summary(additivo)
Call:
lm(formula = peso ~ altezza + esercizio)

Residuals:
    Min      1Q  Median      3Q     Max
-6.597 -1.796   1.002   3.430   4.147

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.9342    37.2780  -0.106   0.9189
altezza        0.4884     0.2078   2.350   0.0511
esercizio     -2.5312     0.7953  -3.183   0.0154
---
Residual standard error: 4.663 on 7 degrees of freedom
Multiple R-squared:0.8258,Adjusted R-squared:  0.7761
F-statistic:  16.6 on 2 and 7 DF,  p-value: 0.002205
```

```
summary(interazione)
Call:
lm(formula = peso ~ altezza * esercizio)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7712 -1.6678 -0.4766  2.4409  4.0870
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -106.15289   49.25013  -2.155  0.07455
altezza            1.06446    0.27658   3.849  0.00847
esercizio         26.71195   11.59888   2.303  0.06085
altezza:esercizio -0.16726    0.06625  -2.525  0.04501

Residual standard error: 3.508 on 6 degrees of freedom
Multiple R-squared: 0.9155,Adjusted R-squared:  0.8733
F-statistic: 21.68 on 3 and 6 DF,  p-value: 0.001275
```

Pare che l'aggiunta del termine di interazione, significativo, migliori la capacità predittiva (R^2_M da .826 a .916), e cambia la sorte della significatività degli effetti principali, anche se questo può essere facilmente attribuito alla ridotta numerosità delle distribuzioni, che rende instabili i $p-value$. Vediamo gli AIC dei due modelli:

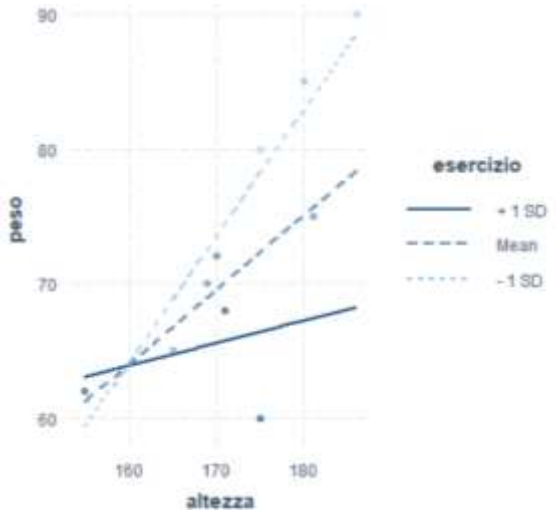
```
AIC(additivo); AIC(interazione)
[1] 63.60726
[1] 58.36934
```

L'AIC del modello con interazione è più piccolo, quindi contiene meno errore: il **fit del modello con interazione è migliore**. L'altezza, indipendentemente dall'esercizio, esercita un effetto positivo sul peso (a maggiore altezza corrisponde un maggior peso) in entrambi i modelli; l'esercizio passa dall'esercitare un effetto negativo (più esercizio, minor peso) a un effetto positivo, anche se a soglia; l'interazione è significativa: l'effetto dell'altezza sul peso è diverso

a seconda del numero di ore di esercizio. Il segno negativo spiega in che modo si eserciti la moderazione: per chi fa meno esercizio, la relazione positiva tra peso e altezza è più forte – o, se preferite, per chi fa più esercizio la relazione positiva tra peso e altezza è meno forte. Insomma, non sorprende che se ti alleni molto la tua altezza diventa meno determinante sul tuo peso, mentre se non fai movimento la tua sola altezza diventa più predittiva del tuo peso. L'effetto di interazione – o moderazione – diventa più chiaro se rappresentato in un grafico: possiamo usare `interact_plot(modello, predittore= X, modx= variabile di moderazione)` di `interactions`. Gli argomenti opzionali della funzione sono molti; almeno per ora, aggiungiamo solo `plot.points = TRUE` , per visualizzare i valori osservati. Nel grafico sono rappresentate tre rette/modelli: la relazione tra Y e X_1 indipendentemente dall'effetto del moderatore X_2 (`Mean`: soggetti con valori compresi tra $+1$ e -1 sd dalla media di X_2), la relazione tra Y e X_1 nei soggetti con valori < 1 sd dalla media di X_2 (punteggi bassi) e la relazione tra Y e X_1 nei soggetti con valori > 1 sd dalla media di X_2. Se l'effetto di interazione non è significativo, le tre rette si presentano parallele, perché la relazione tra Y e X_1 è la stessa per i valori medi, alti o bassi di X_2. La significatività dell'effetto di interazione si manifesta in rette non parallele, e il loro orientamento spiega il senso dell'effetto di moderazione di X_2. Vediamo il nostro caso:

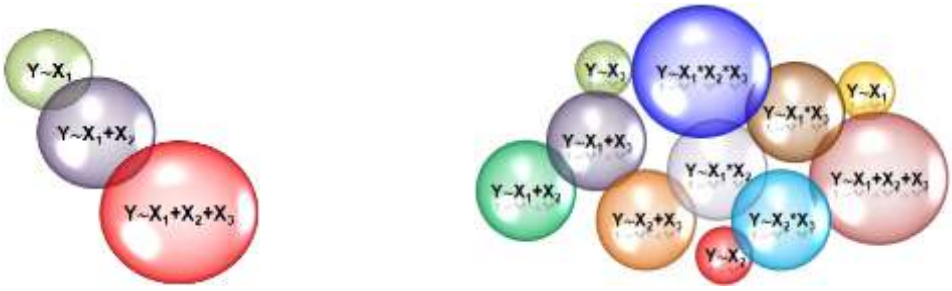
```
interact_plot(interazione, pred = altezza, modx = esercizio, plot.points = TRUE x.label =
  "altezza", y.label = "peso",)
```

Per chi ha valori medi di esercizio (linea tratteggiata blu), la relazione tra peso e altezza ha un chiaro slope positivo. Per chi fa molto esercizio ($> +1sd$, linea continua blu scuro), la relazione tra peso e altezza è ancora positiva, ma decisamente più debole: l'altezza ha meno potere nel determinare il peso. Per chi fa poco esercizio ($< -1sd$, linea azzurro chiaro), la relazione positiva tra peso e altezza è molto più forte: l'altezza è più potente nel determinare il peso. L'esercizio ha quindi un chiaro effetto di moderazione nella relazione tra peso e altezza: più se ne fa, meno è forte la capacità dell'altezza di determinare il peso.



*Rappresentate graficamente il modello con interazione peso ~altezza*esercizio in cui il ruolo di moderatore è attribuito all'altezza, e commentatelo.*

Passiamo ora da una regressione a blocchi alla selezione del modello ottimale da un set di modelli alternativi.
Il processo di selezione può applicarsi a una **model class di modelli nested (nidificati)** o **non nidificati**.



Data una Y e tre X, le model class nested (a sinistra) e non nested sono piuttosto diverse...

Nel **primo caso** (**order selection**), confrontiamo modelli **nested**: i modelli più **piccoli sono sempre casi speciali di modelli più grandi**, ad esempio: $M_1 = Y{\sim}X_1$ *versus* $M_2 = Y{\sim}X_1 + X_2$ *versus* $M_3 = Y{\sim}X_1 + X_2 + X_1X_2$, e **NON** $M_1 = Y{\sim}X_1$ *versus* $M_2 = Y{\sim}X_2$.

La variazione del fit di questi modelli nidificati può essere quantificata dall'opportuno *information criterion*; resta il problema di decidere **l'ordine** con cui i modelli sono costruiti, che deve seguire l'ordine di importanza esplicativa di X → viene prima il predittore che è maggiormente correlato a Y, indipendentemente dalle altre X. L'ordine di importanza esplicativa può **discendere dal presupposto teorico alle spalle della ricerca**, e quindi essere impostato a priori rispetto all'analisi: questo tipo di *order selection* determina una **regressione gerarchica** (§11.1.1), in cui si dovrebbero prima inserire i predittori "sicuri", poi quelli di cui il ricercatore deve ancora dimostrare la capacità predittiva su Y. A loro volta, queste *new entry* possono essere inserite in un solo blocco (per blocchi), o per passi (vedi sotto), o a loro volta gerarchicamente. In un diverso caso, l'ordine di importanza dei predittori non è associato a una teoria esplicativa, e l'ordine della loro selezione è affidata alla cieca statistica, che sceglie tra i possibili X indicati solo **quelli che apportano un cambiamento rilevante nel fit del modello**, stimato come variazione nell'AIC del modello: **regressione per passi** (§11.1.2). Nella regressione per passi, l'unica decisione a priori del ricercatore è specificare la **direzione** di inserimento dei predittori, scegliendo tra:

1) **per passi in avanti** (**forward**): si parte dal **modello nullo**, il cui parametro **è solo l'intercetta**. Il valore assunto da Y è dato solo dalla *grand mean* (intercetta), più una quota d'errore → $y_i = b_0 + e_i$. Per creare il **primo modello**, l'elaboratore (per noi R) sceglie dalla lista di predittori la X con **il maggior coefficiente r con Y**: se il **nuovo modello aggiunge fit rispetto al modello nullo** (AIC / BIC si abbassa), cioè la X predice meglio della sola media di Y, il **predittore è mantenuto** nel modello ($\boldsymbol{X_1}$) e si cerca la seconda X da inserire ($\boldsymbol{X_2}$). Il **secondo predittore** è quello che spiega la **maggior parte della varianza di Y rimasta dopo aver tolto quella spiegata dal primo**. Quindi, se X_1 determina $R^2 = .40$ (spiega da solo il 40% della varianza di Y), resta il 60% della varianza non spiegata: si sceglie come secondo predittore la X che spiega la maggior parte di quel 60%[92]. Se l'aggiunta di X_2 determina un **modello con fit migliore del precedente**, anche X_1 **è mantenuta**. Si prosegue, quindi, accumulando X: la procedura si ferma o quando sono stati inseriti tutti i predittori previsti (se tutti hanno relazioni significative con Y), o quando nuovi predittori non aggiungono fit al modello precedente, cioè se, aggiungendoli, AIC / BIC non si abbassa.

2) **Per passi all'indietro** (**backward**): si parte da un **modello che contiene tutti i predittori previsti**, e che è quindi probabilmente ridondante (*over-fit*). Da questo modello ipertrofico si **tolgono**, uno per volta, i predittori non correlati con Y, cioè **quelli la cui eliminazione abbassa $\boldsymbol{AIC}$ / $\boldsymbol{BIC}$**, partendo da quelli la cui eliminazione comporta il miglioramento più sensibile nel fit. A ogni passo, si riparte dal nuovo AIC / BIC e si procede alla rimozione della peggiore tra le X rimaste. La procedura s'interrompe quando la rimozione di ulteriori X determinerebbe un decremento del fit (AIC aumenta): i predittori rimasti definiscono il miglior modello.

3) **Both [in R] o stepwise**: si parte con l'ordinamento forward, quindi da un modello nullo, ma ogni volta che viene inserita una nuova X i predittori inseriti sono rivalutati e possono essere rimossi, se non contribuiscono più al fit del modello.

Attenzione, nel fittare molti modelli allo stesso corpus di dati, a un fenomeno che abbiamo già trovato correlando molte variabili a coppie: sarà possibile trovare X statisticamente significative in un modello e non in un altro, dato che ciascun test in **ogni modello ha un proprio tasso di errore di I tipo**.

[92] ovvero la X che ha la maggior correlazione **semi-parziale** con Y

Infine, si può valutare una **model class non (solo) nidificata**, in cui si creano **tutti i possibili modelli derivabili dalle combinazioni dei X**, o almeno quelli per cui si possono porre ipotesi sensate: dal modello nullo al modello completo, con effetti additivi e/o di interazione tra le X. Viene quindi generata una "classifica" dei modelli, basata sulla miglior variazione del fit del modello e della sua verosimiglianza. Oltre al modello migliore, è possibile identificare anche un **confidence set di modelli**, analogo all'usuale CI per una statistica campionaria, al cui interno si trovano modelli alternativi al migliore e ragionevolmente plausibili in popolazione. Vedremo i dettagli della procedura nel §11.1.3. Ovviamente, il numero di predittori possibili determina un numero di modelli esponenzialmente crescente, e il vero problema, come nella regressione per passi, è la giustificazione **a posteriori** dei predittori inseriti nel miglior modello identificato: non sempre la giustificazione statistica e quella teorica sono in accordo.

Insomma, se è possibile, è meglio adottare un inserimento che replichi statisticamente un fondato modello teorico: se il fit risultasse cattivo, si valuterà la possibilità **a posteriori** di aggiustamenti al modello. Se la regressione si applica a un modello per cui le ipotesi esplicative a priori sono poco fondate, ad esempio in campi su cui la letteratura è ancora scarsa, si può usare un inserimento per blocchi, che garantisce una buona replicabilità. Se proprio si decide di usare un inserimento per passi, meglio allora usare l'ordinamento backward, che riduce maggiormente l'errore di II tipo (escludere un predittore che potrebbe essere utile) rispetto al metodo forward.

11.2 Un esempio di regressione gerarchica

Facciamo un esempio di regressione multipla gerarchica usando pochi dati semplici: dopo aver visto la relazione tra peso e altezza nel capitolo 9, ora vediamo modelli più completi, aggiungendo prima il nuovo predittore ***ore di esercizio***, *p*oi il predittore ***età***: l'ordine è determinato dalla mia personale ipotesi che l'esercizio agisca sul peso meno dell'altezza, ma più dell'età, e che l'età agisca sul peso, ma meno di altezza ed esercizio.

```
peso<-c(80,68,72,75,70,65,62,60,85,90)
altezza<-c(175,171,170,181,169,165,155,175,180,186)
esercizio<-c(0,6,4,3,3,2,5.5,6,1,0)
eta<-c(25,31,33,41,54,38,42,34,53,40)
```

Avremo quindi tre modelli da confrontare:

- **Modello 1:** $Y_{peso}, X_{altezza} \rightarrow y_i = (b_0 + b_1X_1) + e_i$
- **Modello 2:** $Y_{peso}, X_{1_{altezza}}, X_{2_{esercizio}} \rightarrow y_i = (b_0 + b_1X_1 + b_2X_2) + e_i$
- **Modello 3:** $Y_{peso}, X_{1_{altezza}}, X_{1_{esercizio}}, X_{3_{età}} \rightarrow y_i = (b_0 + b_1X_1 + b_2X_2 + b_3X_3) + e_i$

Ogni **modello è nidificato** (**nested**) nel precedente: contiene gli **stessi parametri più (almeno) un parametro nuovo**. Il **succedersi dei modelli** segue un ordine **gerarchico**, e in ogni modello i predittori sono inseriti in blocco**:** se non troveremo una **variazione significativa del fit** del modello 2 rispetto al modello 1, e/o del fit del modello 3 rispetto al modello 2, vuol dire che la X / le X inserite nel nuovo modello non aggiungono capacità predittiva al modello.

```
modello1<-lm(peso~altezza)
modello2<-lm(peso~altezza+esercizio)
modello3<-lm(peso~altezza+esercizio+eta)
```

Per velocizzare la costruzione dei modelli, si può usare `update(modello, .~. + nuovo elemento)`: **`.~.`** indica a R: "mantieni tutto uguale tranne" l'aggiunta (**+**) del nuovo elemento (**predittore**):

```
modello2<-update(modello1, .~.+esercizio)
modello3<-update(modello2, .~.+eta)
```

Vediamo i modelli nel loro complesso, prima; poi approfondiremo i singoli predittori dei tre modelli.

```
summary(modello1)
[omissis]
Residual standard error: 6.824 on 8 degrees of freedom
Multiple R-squared:  0.5738,    Adjusted R-squared:  0.5205
F-statistic: 10.77 on 1 and 8 DF,  p-value: 0.01116

summary(modello2)
[omissis]
Residual standard error: 4.663 on 7 degrees of freedom
Multiple R-squared:  0.8258,    Adjusted R-squared: 0.7761
F-statistic:  16.6 on 2 and 7 DF,  p-value: 0.002205

summary(modello3)
[omissis]
Residual standard error: 4.958 on 6 degrees of freedom
Multiple R-squared:  0.8312,    Adjusted R-squared: 0.7469
F-statistic: 9.851 on 3 and 6 DF,  p-value: 0.009824
```

Da sola (`modello1`), l'**altezza** predice il **57.4%** della variabilità del peso, il 52.05% in popolazione. Il rapporto tra MS_M e MS_R è $=$ 10.8 ed è significativo. Aggiungendo **l'esercizio** all'altezza, la variabilità spiegata di Y fa un balzo rilevante a $R^2_M = 82.6$ nel campione, $R^2_{adj} = 77.6$ in popolazione. Il rapporto F aumenta, indicando che la variabilità spiegata da X_1 e X_2 è ancora più grande dell'errore (come conferma la radice quadrata di MS_R, che diminuisce). Aggiungendo **l'età** ad altezza ed esercizio, le cose non vanno altrettanto bene. R^2_M aumenta da 82.6 *a* 83.1 (cioè poco), ma in popolazione la stima $R^2_{adj} = 74.7$ si abbassa ulteriormente, *ergo* l'intervallo tra R^2_M e R^2_{adj} aumenta: abbiamo detto nel Capitolo 9 che quando la differenza tra R^2_M e R^2_{adj} è grande, nel modello qualcosa non va. Altri segnali negativi sono il rapporto F, che diventa addirittura più piccolo che nel modello con un solo predittore, e il *residual standard error*, che cresce rispetto al modello 2: aggiungendo l'età, **abbiamo introdotto più errore che capacità esplicativa**.

Verifichiamolo con l'AIC:

```
AIC(modello1); AIC(modello2); AIC(modello3)
[1] 70.55665
[1] 63.60726
[1] 65.29186
```

Passando dal modello 1 al modello 2 AIC si abbassa, quindi il fit migliora; dal modello 2 al modello 3, AIC si alza, quindi il fit **peggiora**. Valutare un modello sulla base del solo R^2_M, quindi, porterebbe a conclusioni ingannevoli.

È anche possibile applicare un **test di significatività alla variazione di R^2_M**, che è, ancora una volta, un **rapporto F**, per valutare se la variazione di R^2_M da un modello all'altro sia **significativa**, ovvero che non rappresenti una casuale oscillazione del fit.

Il rapporto F che **esprime la differenza nella variazione di R^2_M da un modello nullo** (in cui $R^2_M = 0$) al modello in cui è inserito **almeno un** predittore (non nullo) è dato da:

$$F_{R^2_0 - R^2_1} = \frac{(N-k-1)R^2_1}{k(1-R^2_1)}$$

in cui N = il numero di osservazioni, k = numero di predittori nel modello **non nullo**, R^2_M = R^2_M del modello non nullo. Nel nostro esempio:

```
R2_1<-.5738
R2_2<-.8258
R2_3<-.8312
(F1<- ((10-1-1)*R2_1)/(1*(1-R2_1)))
[1] 10.77053
```

Come nella regressione semplice, i df di F sono $df_M = numero\ di\ parametri - 1$, $df_R = N - numero\ b_1 - 1$.

Quindi, possiamo chiedere:

```
pf(F1,df1 =  1, df2 = 10-2, lower.tail = FALSE)
[1] 0.01115697
```

... e ricordare che questa è appunto la statistica F che abbiamo letto nel `summary` del `modello1`, come espressione della significatività dell'unico predittore $Altezza$:

```
F-statistic: 10.77 on 1 and 8 DF,  p-value: 0.01116
```

Per valutare la **variazione del fit dal modello 2 al modello 3**, invece, consideriamo: $F_{R_1^2-R_2^2} = \frac{(N-k_2-1)diff_{R_1^2-R_2^2}}{diff_{k_1-k_2}(1-R_2^2)}$, in cui: k_2 = il numero di predittori nel modello 2, $diff_{R_1^2-R_2^2}$ = differenza (in valore assoluto) tra i coefficienti R_M^2 del modello 1 e del modello 2, $diff_{k_1-k_2}$ = differenza (in valore assoluto) nel numero di predittori del modello 2 e del modello 3, R_2^2 = coefficiente R_M^2 del modello 2.

Avremo allora:

```
(F2<-((10-2-1)*abs(R2_1-R2_2))/(abs(1-2)*(1-R2_2)))
[1] 10.12629
pf(F2,df1 =  2, df2 = 7, lower.tail = FALSE)
[1] 0.008588479
```

La variazione di R_M^2 da $.574$ del modello 1 a $.826$ è anch'essa significativa: aggiungere il predittore $Esercizio$ all'altezza migliora significativamente il fit del modello.

Infine, vediamo la variazione dal modello 2 al modello 3:

```
(F3<-((10-3-1)*abs(R2_2-R2_3))/(abs(2-3)*(1-R2_3)))
[1] 0.1919431
```

Poiché $F_{modello\,3} < 1$, possiamo anche evitare di chiedere il $p-value$, che sarà certamente $p > .05$: aggiungere il predittore $Età$ non migliora significativamente il modello.

In R, la verifica della variazione del fit può essere gestita da **anova**, che è una funzione piuttosto eclettica (la ritroveremo anche nella regressione logistica): applicata a **un solo modello** restituisce la **tavola di *SS* e *MS*** del modello: **`anova(modello)`**. Per esempio:

```
anova(modello1)
Analysis of Variance Table
Response: peso
          Df Sum Sq Mean Sq F value  Pr(>F)
altezza    1 501.55  501.55   10.77 0.01116 ← df_M, SS_M, MS_M, F, p − value
Residuals  8 372.55   46.57                 ← df_R, SS_R, MS_R
```

Oppure:

```
anova(modello3)
Analysis of Variance Table
Response: peso
             Df   Sum Sq   Mean Sq   F value     Pr(>F)
 altezza      1   501.55    501.55   20.4007   0.004032
 esercizio    1   220.31    220.31    8.9613   0.024209  ← df_M, SS_M , MS_M, F, p − value
 eta          1     4.73      4.73    0.1923   0.676394
 Residuals    6   147.51     24.58                      ← df_R, SS_R, MS_R,
```

Quando gli argomenti di `anova` sono modelli *nested*: `(modello`$_k$`,modello`$_{k+1}$`,…, modello`$_{ennesimo}$`)`, la funzione **confronta ogni modello con il precedente**:

```
anova(modello1, modello2, modello3)
Analysis of Variance Table[93]

Model 1: peso ~ altezza
Model 2: peso ~ altezza + esercizio
Model 3: peso ~ altezza + esercizio + eta
  Res.Df      RSS     Df   Sum of Sq         F    Pr(>F)
1      8    372.55                                      ← df_R, SS_R
2      7    152.24      1     220.314    8.9613   0.02421  ← df_R, SS_R , df_M, SS_M, F di X2 Esercizio
3      6    147.51      1       4.727    0.1923   0.67639  ← df_R, SS_R , df_M, SS_M, F di X3 Età
```

Aggiungere l'esercizio determina una variazione significativa rispetto a `modello1`, mentre l'aggiunta dell'età non rappresenta una modifica significativa del fit.

D'altronde, questa funzione è **del tutto equivalente** a `summary(aov(modello3))`**:**

```
summary(aov(modello3))
            Df Sum Sq Mean Sq F value  Pr(>F)
altezza      1  501.5   501.5  20.401 0.00403
esercizio    1  220.3   220.3   8.961 0.02421
eta          1    4.7     4.7   0.192 0.67639
Residuals    6  147.5    24.6
```

Infatti (lo vedremo anche nel Capitolo 13), `aov` esegue una partizione della devianza totale perfettamente corrispondente a quella seguita nell'inserimento gerarchico: il primo predittore (non mostrato in `anova`) spiega quel che può del 100% della variabilità di Y, $\boldsymbol{X_2}$ spiega quel che può della varianza di Y **non spiegata da** X_1, $\boldsymbol{X_3}$ spiega la varianza di Y **non spiegata dal primo e dal secondo predittore**, ecc. (ecco, anche, perché i valori F di `anova` e del test F calcolato passo passo non coincidono perfettamente).

Passiamo ora ai tre b_1 dei predittori in `modello3`:

```
summary(modello3)
[omissis]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.18179   40.80155  -0.201   0.8477
altezza      0.49401    0.22134   2.232   0.0671
esercizio   -2.47748    0.85438  -2.900   0.0273
eta          0.07987    0.18216   0.438   0.6764
```

Per ogni cm in più, il peso aumenta di .5 kg, indipendentemente da esercizio ed età, e l'effetto è alle soglie della significatività (pochi soggetti, forse). Per ogni ora di esercizio in più, il peso cala di 2.5kg, indipendentemente da altezza ed età, e questo effetto non è casuale. Per ogni anno di età in più, il peso aumenta di 8 gr, indipendentemente da altezza ed esercizio: in realtà, questa variazione è da considerare solo casuale, dato che l'età non ha un effetto significativo.

Abbiamo **definito come migliore il modello a due predittori**. Di questo modello, quindi, dobbiamo interpretare l'output, completarlo con le informazioni che non vi sono comprese e valutarne l'accuratezza.

```
summary(modello2)
Residuals:
   Min     1Q Median     3Q    Max
-6.597 -1.796  1.002  3.430  4.147

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.9342    37.2780  -0.106   0.9189
altezza       0.4884     0.2078   2.350   0.0511
esercizio    -2.5312     0.7953  -3.183   0.0154
---
Residual standard error: 4.663 on 7 degrees of freedom
Multiple R-squared:  0.8258,    Adjusted R-squared:  0.7761
F-statistic:  16.6 on 2 and 7 DF,  p-value: 0.002205
```

[93] RSS = SS_R dei tre modelli; Sum of Sq: SS_M del predittore inserito nei passi

```
confint(modello2)
                   2.5 %     97.5 %
(Intercept) -92.08259913 84.2142570
altezza      -0.00300737  0.9798961
esercizio    -4.41172278 -0.6506837
```

Il vero calo ponderale in popolazione, per ogni ora di esercizio, in più sta **tra -4 kg e -7 etti**: se questi dati fossero reali, sarebbe una potenziale fregatura ☺.

Nella regressione semplice (`modello1`), il coefficiente standardizzato β_1 corrisponde al coefficiente di correlazione r tra Y e X, come già sappiamo dal Capitolo 9.

```
modello1$coefficients[2]*(sd(altezza)/sd(peso))
  altezza
0.7574889
round(cor(altezza,peso),3)
[1] 0.757
```

Nella regressione multipla (`modello2`), i $b_{1_{Altezza}}$ e $b_{1_{Esercizio}}$ sono analoghi a **coefficienti di correlazione parziale di primo ordine** tra peso e altezza, al netto dell'esercizio, e tra peso e altezza, al netto dell'altezza, come dimostra la probabilità loro attribuita. Si definiscono, infatti, anche **coefficienti angolari parziali** (**partial slope**).

Per mettere a confronto la loro capacità di influenzare la variazione di Y, dato che X_1 e X_2 hanno unità di misura differenti, si possono **standardizzare in coefficienti beta**:

$$\beta_1 = b_1 \frac{s_x}{s_y}$$

```
(beta_altezza<-modello2$coefficients[2]*(sd(altezza)/sd(peso)))
  altezza
0.4390032
(beta_esercizio<-modello2$coefficients[3]*(sd(esercizio)/sd(peso)))
esercizio
-0.594542
```

Si può anche usare `lm.beta(modello)` del package `lm.beta`:

```
lm.beta(modello2)
Standardized Coefficients::
  (Intercept)       altezza      esercizio
   0.0000000         0.4390032   -0.5945420
```

Per un'unità di deviazione standard in più in altezza, il peso aumenta di meno di mezza deviazione standard; per una unità di deviazione standard in più nell'esercizio, il peso cala di -.6 deviazioni standard: **l'effetto dell'esercizio sulla variazione del peso è più forte** dell'effetto dell'altezza.

Grazie ai β_1, possiamo scindere la variabilità spiegata di Y in **quote di varianza attribuite a ciascuno dei due predittori**: R^2 multiplo è uguale alla somma dei prodotti dei coefficienti β_1 per il coefficiente r tra Y e il rispettivo predittore.

$$R^2_{multiplo} = \beta_{X_1} r_{yX_1} + \beta_{X_2} r_{yX_2}$$

```
((quota_variabilità_altezza<- beta_altezza * cor(peso, altezza)))
[1] 0.3325401
((quota_variabilita_esercizio<- beta_esercizio * cor(peso,esercizio)))
[1] 0.4932964
```

L'altezza spiega il 33.3% della variabilità del peso, indipendentemente dall'esercizio, mentre l'esercizio ne spiega il 49.3%, indipendentemente dall'altezza: è quindi maggiormente determinante. La somma di queste due ripartizioni dà, naturalmente, l'$R^2_M = .826$ visto nell'output:

```
quota_variabilità_altezza+quota_variabilita_esercizio
[1] 0.8258365
```

Il commento al `modello2` dovrebbe quindi avere suppergiù questa forma:

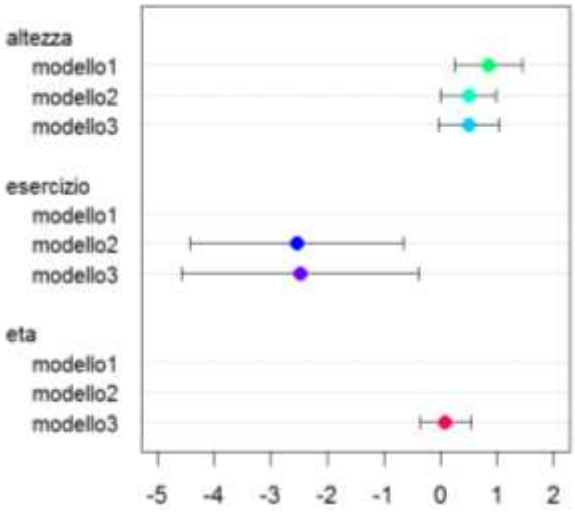
Gli errori del modello vanno da una sottovalutazione massima di 6.6kg a una sopravvalutazione massima di 4.1Kg: l'errore mediano è di circa 1kg, quindi accettabile. I due predittori, insieme, spiegano oltre l'83% della variabilità del peso, stima che si abbassa in popolazione a circa il 78%. Il modello è nel suo complesso significativo: la varianza spiegata dai predittori è significativamente maggiore di quella attribuibile all'errore. L'altezza, indipendentemente dall'esercizio, esercita un effetto positivo e non casuale sul peso, quantificabile in circa mezzo chilo in più per ogni cm di altezza in più; tuttavia, in popolazione, la predizione è piuttosto imprecisa, in quanto con il 95% di verosimiglianza va da pochi grammi a quasi un kg in più per cm. La relazione tra esercizio fisico e peso, al netto dell'altezza, altrettanto significativamente diversa da 0, è invece negativa: per ogni ora di esercizio in più si prevede un calo di circa 2.5 kg; nuovamente, però, in popolazione la stima è imprecisa, in quanto il calo previsto oscilla tra poco più di 6hg e quasi 4.5kg. Il confronto tra i due coefficienti standardizzati indica che l'effetto dell'esercizio sul peso è più forte di quello esercitato dall'altezza, dato che comporta una variazione in valori assoluti di quasi 0.6 deviazioni standard versus 0.4 deviazioni standard. Infatti, la scomposizione dell'R^2 multiplo dice che l'altezza, al netto del peso, spiega il 33.3% della variabilità del peso, e l'esercizio, al netto dell'altezza, il 49.3%.

Per aiutarci a confrontare le informazioni essenziali di più modelli usando una sola funzione, possiamo sfruttare `TMod(modello1, modello2, modello3...)` di `DescTools`: le **caratteristiche più importanti dei modelli sono tabulate** con una modalità simile a quella che vedremo nel §11.4, consentendo un'efficace valutazione parallela:

```
modello3<-lm(peso~altezza+esercizio+eta)
TMod(modello1,modello2, modello3)
```

	coef	modello1	modello2	modello3	
1	(Intercept)	-72.851	-3.934	-8.182	← b_0
2	altezza	0.843 *	0.488 .	0.494 .	← b_1 e significatività
3	esercizio	-	-2.531 *	-2.477 *	
4	eta	-	-	0.080	
5	---				
6	r.squared	0.574	0.826	0.831	← R^2 e R^2_{adj}
7	adj.r.squared	0.521	0.776	0.747	
8	sigma	6.824	4.663	4.958	← è il *residual standard error* dell'output
9	logLik	-32.278	-27.804	-27.646	← LL e $-2LL$: le vedremo nel capitolo 14, per ora ignoratele
10	deviance	372.551	152.236	147.510	
11	AIC	70.557	63.607	65.292	← information criteria
12	BIC	71.464	64.818	66.805	
13	numdf	1	2	3	← df del modello (`numdf`) e di errore (`dendf`)
14	dendf	8	7	6	
15	N	10	10	10	← N, numero di variabili e numero di parametri nei modelli
16	n vars	1	2	3	
17	n coef	2	3	4	
18	F	10.770	16.596	9.851	← F e significatività del modello
19	p	0.011	0.002	0.010	
20	MAE	4.762	3.479	3.412	← Mean Absolute Error, Mean Absolute Percentage Error, Means Squared Error, Root Means Square Error: sono misure di accuratezza del modello (*accuracy*), importanti nel modeling – ma non entrano nel nostro programma.
21	MAPE	0.068	0.050	0.050	
22	MSE	37.255	15.224	14.751	
23	RMSE	6.104	3.902	3.841	

Salvando `TMod(mod1, mod2,....)` come oggetto, si può **plottare**: il risultato è un bel grafico dei coefficienti angolari di ogni predittore nei modelli, con relativo CI:




```
confronto<-TMod(modello1,modello2, modello3)
plot(confronto, pch=19, cex=1.5, col=rainbow(15))
```

`Tmod` si applica a qualsiasi tipo di modello lineare, generale e generalizzato, anche con predittori categoriali.

Prima di vedere un esempio di regressione per passi, soffermiamoci sulla **centratura** dei predittori. Abbiamo già visto nella regressione semplice che per alcune X un valore $X = 0$ è irrealistico, per cui l'interpretazione di b_0 (valore in Y per $X = 0$) è priva di senso. Nel nostro esempio, un giovane adulto non può assumere $X_{Altezza} = 0$ o $X_{Età} = 0$, anche se può certamente fare $X_{Esercizio} = 0$ ore di esercizio fisico, tanto che il valore 0 è effettivamente presente nella distribuzione `esercizio`): quindi l'intercetta del modello con altezza ed età non ha un vero significato. Possiamo, però, **traslare i predittori come scarti attorno alla media della distribuzione X**, cioè **centrarli sulla media di X**, e inserire queste distribuzioni di scarti come predittori nel modello: in questo modo, uno **scarto $X = 0$ corrisponde al valore medio** della distribuzione → quindi, l'intercetta del modello con predittori centrati corrisponderà al **valore assunto in Y per il valore medio di X**.

Per esempio:

```
altezza_cen<-altezza-mean(altezza); esercizio_cen<-esercizio-mean(esercizio); eta_cen<-eta-
mean(eta)
mean(altezza); mean(eta); mean(esercizio)
[1] 172.7
[1] 39.1
[1] 3.05
round(mean(altezza_centrato),1); round(mean(eta_centrato),1); round(mean(esercizio_centrato),1)
[1] 0
[1] 0
[1] 0
```

C'è una scorciatoia: nel capitolo 4 abbiamo usato `scale(distribuzione)` per standardizzare una variabile in punti z, lasciando di default i suoi argomenti `center= TRUE` e `scale= TRUE` e ottenendo così una variabile centrata **e** scalata (divisa per la deviazione standard della distribuzione grezza). Ora vogliamo **solo centrare**, perciò lasciamo `center= TRUE` e indichiamo `scale= FALSE`:

```
altezza_cen<-scale(altezza, center=TRUE, scale=FALSE)
esercizio_cen<-scale(esercizio, scale=FALSE)
eta_cen<-scale(eta, scale=FALSE)
modello3_centrato<-lm(peso~altezza_cen + esercizio_cen + eta_cen)
```

Mettiamo a confronto i modelli *overall* con predittori centrati e non centrati: **sono naturalmente identici,** dato che la traslazione è una trasformazione lineare che non altera i rapporti tra Y e le X.

```
summary(modello3)
Residual standard error:4.958 on 6 degrees of freedom
Multiple R-squared:0.8312,Adjusted R-squared: 0.7469
F-statistic: 9.851 on 3 and 6 DF,  p-value: 0.009824
```

```
summary(modello3_centrato
Residual standard error:4.958 on 6 degrees of freedom
Multiple R-squared:0.8312,Adjusted R-squared:  0.7469
F-statistic: 9.851 on 3 and 6 DF,  p-value: 0.009824
```

Invece, nei coefficienti del modello:

```
summary(modello3)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.18179   40.80155  -0.201   0.8477
altezza      0.49401    0.22134   2.232   0.0671
esercizio   -2.47748    0.85438  -2.900   0.0273
eta          0.07987    0.18216   0.438   0.6764
```

```
summary(modello3_centrato)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   72.70000    1.56796  46.366 6.74e-09
altezza_cen    0.49401    0.22134   2.232   0.0671
esercizio_cen -2.47748    0.85438  -2.900   0.0273
eta_cen        0.07987    0.18216   0.438   0.6764
```

I **coefficienti angolari sono immutati, l'intercetta cambia**: nel modello centrato $b_0 = 72.2$ esprime il peso atteso di un giovane adulto di altezza media (172.7 cm), di età media (39.1 anni) e che fa una quantità di esercizio medio (3.1 ore a settimana).

Se scegliessimo di **centrare solo altezza ed età**, dato che l'esercizio prevede uno 0 sensato:

```
summary(lm(peso~altezza_centrato+esercizio+eta_centrato))
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      80.25632    3.04122  26.389 1.95e-07
altezza_centrato  0.49401    0.22134   2.232   0.0671
esercizio        -2.47748    0.85438  -2.900   0.0273
eta_centrato      0.07987    0.18216   0.438   0.6764
```

$\boldsymbol{b_0 = 80.2}$ esprime il peso atteso di un giovane adulto di altezza media (172.7 cm), di età media (39.1 anni) e che fa **zero ore di esercizio a settimana**; notate come cambia il peso atteso per $X = 0$ dal modello precedente, in cui b_0 originava da un esercizio medio di 3.1.ore a settimana, a parità degli altri predittori).

11.3 Un esempio di selezione per passi

Ora vediamo un esempio di regressione per passi, con inserimento backward e forward, su dati veri: la depressione dei *caregiver* (dataframe `attaccamento`, che già conosciamo) è determinata / predetta dal carico assistenziale? E, più precisamente, da quali dimensioni del carico assistenziale?

Poiché i lunghi nomi delle variabili danno fastidio nell'output della regressione, le estraiamo e le rinominiamo:

```
a<-data.frame(depressione = attaccamento$BDI_II_depressione, restrizione_tempo =
   attaccamento$CBI_burden_restrizione_tempo, blocco = attaccamento$CBI_burden_blocco_evolutivo,
   fisico = attaccamento$CBI_burden_fisico, ruolo = attaccamento$CBI_burden_conflitto_ruolo,
   emotivo = attaccamento$CBI_burden_emotivo)

head(a, 4)
  depressione restrizione_tempo blocco fisico ruolo emotivo
1          23                 5      5      4     2       3
2          24                 9     17      8    14      17
3           9                14     14      3    11       6
4          35                20     18     14    10       4
```

Non facciamo ipotesi a priori su quale tra carico emotivo, fisico, restrizione di tempo personale, blocco evolutivo e conflitto di ruolo pesi di più sul malessere del caregiver: consideriamo il campione complessivo e usiamo un metodo di inserimento per passi backward. La funzione per impostare un inserimento per passi è **`step`**, in questo caso applicata a un modello lineare[94], con **direzione all'indietro**, e quindi: **`step(lm(Y~X1+X2+...Xk), direction="backward")`**;

[94] La vedremo anche applicata a un modello lineare generalizzato nella regressione logistica multipla, Capitolo 14

possiamo anche omettere direction, che di default è "backward". L'output è lungo, perché spiega passo per passo quale predittore viene eliminato in ogni verifica, fino ad arrivare al modello con i migliori predittori. Lo spezziamo per seguirne il ragionamento.

```
step(lm(a$depressione ~ a$restrizione_tempo + a$blocco + a$fisico + a$ruolo + a$emotivo), direction
   = "backward")
```

```
Start:  AIC=165.33
```

Il fit del modello con tutti i predittori è = 165.33. Togliendo un predittore ("-"), come cambia il fit?

```
a$depressione ~ a$restrizione_tempo + a$blocco + a$fisico + a$ruolo + a$emotivo

                      Df Sum of Sq    RSS    AIC
- a$ruolo              1     11.74 1860.0 163.58
- a$blocco             1     76.18 1924.5 164.94
<none>                             1848.3 165.33
- a$emotivo            1    117.57 1965.9 165.79
- a$restrizione_tempo  1    199.43 2047.7 167.42
- a$fisico             1   1138.27 2986.6 182.52
```

Togliendo $ruolo, AIC si abbassa, il fit migliora

Togliendo $blocco, AIC si abbassa, il fit migliora

Togliendo $emotivo, AIC si alza, il fit peggiora

Togliendo $restrizione, AIC si alza, il fit peggiora

Togliendo $fisico, AIC si alza, il fit peggiora

Togliere ruolo e blocco migliora il fit: tra le due, è l'eliminazione di ruolo a determinare il miglioramento maggiore, quindi la togliamo dal modello.

```
Step:  AIC=163.58
```

Si riparte da un fit = 165.33. Togliendo un predittore ("-"), come cambia il fit?

```
a$depressione ~ a$restrizione_tempo + a$blocco
+ a$fisico + a$emotivo

                      Df Sum of Sq    RSS    AIC
- a$blocco             1     70.71 1930.8 163.07
<none>                             1860.0 163.58
- a$emotivo            1    107.11 1967.2 163.82
- a$restrizione_tempo  1    220.99 2081.0 166.07
- a$fisico             1   1335.74 3195.8 183.23
```

Togliendo $blocco, AIC si abbassa, il fit migliora

Togliendo $emotivo, AIC si alza, il fit peggiora

Togliendo $restrizione, AIC si alza, il fit peggiora

Togliendo $fisico, AIC si alza, il fit peggiora

Togliere blocco migliora il fit: facciamolo

```
Step:  AIC=163.07
a$depressione ~ a$restrizione_tempo + a$fisico +
a$emotivo

                      Df Sum of Sq    RSS    AIC
- a$emotivo            1     53.21 1984.0 162.16
<none>                             1930.8 163.07
- a$restrizione_tempo  1    224.37 2155.1 165.47
- a$fisico             1   1429.06 3359.8 183.23
```

Si riparte da un fit = 165.07. Togliendo un predittore ("-"), come cambia il fit?

Togliendo $emotivo, AIC si abbassa, il fit migliora

Togliendo $restrizione, AIC si alza, il fit peggiora

Togliendo $fisico, AIC si alza, il fit peggiora

Togliere emotivo migliora il fit: facciamolo

```
Step:  AIC=162.16
a$depressione ~ a$restrizione_tempo + a$fisico

Df Sum of Sq  RSS    AIC
<none>                       1984 162.16
- a$restrizione_tempo  1    251.04 2235 164.93
- a$fisico             1   1639.99 3624 184.26
```

Togliendo $restrizione, AIC si alza, il fit peggiora

Togliendo $fisico, AIC si alza, il fit peggiora

Togliere restrizione e fisico peggiora il fit, l'unica soluzione vincente è non fare nulla: il procedimento si interrompe e R dà i parametri del modello migliore

```
Call:
lm(formula = a$depressione ~ a$restrizione_tempo + a$fisico)

Coefficients:
        (Intercept)  a$restrizione_tempo             a$fisico
            10.5074              -0.5074               1.8206
```

A questo punto, si procede a esaminare il modello migliore e a cercare di spiegare (a posteriori) perché **una maggior depressione sia spiegata** solo **da una minore restrizione del tempo per sé** (relazione negativa) e d**a un maggior onere fisico** dell'assistenza (relazione positiva).

Volendo fare una regressione in avanti, si usa ancora **step**, ma la formula $Y \sim X$ è diversa, dato che ora si parte da un modello nullo, in cui y_i è predetto dalla sola intercetta, per arrivare (**scope= ~**) a un modello che contenga i predittori indicati procedendo in avanti (direction= "forward"). Il modello nullo è indicato a R come **$Y \sim 1$** , in cui "1" sta per b_0.

Avremo: step(lm(Y~1), direction= "forward", scope= ~X_1 + X_2 + X_3 +... X_K).

```
step(lm(a$depressione ~ 1), scope= ~ a$restrizione_tempo + a$blocco + a$fisico + a$ruolo +
a$emotivo, direction = "forward")
Start:  AIC=182.71
a$depressione ~ 1
```

Il fit del modello nullo è = 182.71. Aggiungendo un predittore ("+"), come cambia il fit?

	Df	Sum of Sq	RSS	AIC	
+ a$fisico	1	1429.97	2235.0	164.93	*Aggiungendo $fisico, AIC si abbassa, il fit migliora*
+ a$ruolo	1	523.88	3141.1	178.54	*Aggiungendo $ruolo, AIC si abbassa, il fit migliora*
+ a$blocco	1	409.66	3255.3	179.97	*Aggiungendo $blocco, AIC si abbassa, il fit migliora*
+ a$emotivo	1	267.65	3397.3	181.68	*Aggiungendo $emotivo, AIC si abbassa, il fit migliora*
<none>			3665.0	182.71	
+ a$restrizione_tempo	1	41.02	3624.0	184.26	*Aggiungendo $restrizione, AIC si alza, il fit peggiora*

Aggiungere fisico, ruolo, blocco ed emotivo migliora il fit: tra tutte, è l'aggiunta di fisico a determinare il miglioramento maggiore, quindi la inseriamo nel modello.

```
Step:  AIC=164.92
a$depressione ~ a$fisico
```

	Df	Sum of Sq	RSS	AIC	
+ a$restrizione_tempo	1	251.044	1984.0	162.16	*Aggiungendo restrizione, AIC si abbassa, il fit migliora*
<none>			2235.0	164.93	
+ a$emotivo	1	79.891	2155.1	165.47	*Aggiungendo emotivo, AIC si alza, il fit peggiora*
+ a$blocco	1	12.283	2222.7	166.70	*Aggiungendo blocco, AIC si alza, il fit peggiora*
+ a$ruolo	1	0.849	2234.2	166.91	*Aggiungendo ruolo, AIC si alza, il fit peggiora*

L'unica mossa che migliora il fit è aggiungere la restrizione, quindi la inseriamo nel modello.

```
Step:  AIC=162.16
a$depressione ~ a$fisico + a$restrizione_tempo
```

	Df	Sum of Sq	RSS	AIC	
<none>			1984.0	162.16	
+ a$emotivo	1	53.214	1930.8	163.07	*Aggiungendo emotivo, AIC si alza, il fit peggiora*
+ a$blocco	1	16.811	1967.2	163.82	*Aggiungendo blocco, AIC si alza, il fit peggiora*
+ a$ruolo	1	0.504	1983.5	164.15	*Aggiungendo ruolo, AIC si alza, il fit peggiora*

Aggiungere emotivo, blocco o ruolo peggiora il fit, l'unica soluzione vincente è non fare nulla: il procedimento si interrompe e R dà i parametri del modello migliore

```
Call:
lm(formula = a$depressione ~ a$fisico + a$restrizione_tempo)
Coefficients:
        (Intercept)             a$fisico  a$restrizione_tempo
            10.5074               1.8206              -0.5074
```

Come spesso – ma non sempre – succede, i due metodi hanno prodotto lo stesso modello a due predittori come migliore.

Per l'inserimento ***both***, si usa step applicata a un lm(Y~X1+X2+X3...), con direction = "**both**". Per la procedura ***all subset***, si può usare il package **leaps**, funzione **leaps(data= Y~X1+X2+X3..., nbest= 1** [numero di modelli migliori]**, nvmax= NULL** [nessun limite al numero di variabili]**, method= "exhaustive")**.

Richiedete e commentate tutti gli elementi interpretativi del modello depressione ~ restrizione + fisico.

11.4 Un esempio di selezione con modelli non nidificati

Per questa model selection ci servirà la funzione **`model.sel(modelli)`** del package **`MuMIn`**, che torneremo a usare nel capitolo 15. La funzione restituisce una tabella in cui **i modelli sono ordinati dal migliore al peggiore** secondo **AIC_C** .
Costruiamo una *model class* di modelli relativi alla relazione tra depressione e dimensioni di burden. Per limitare i tanti possibili modelli e facilitare la lettura dell'output, che comprende un po' di novità, usiamo solo alcune dimensioni, in particolare quelle che sappiamo essere significative dal paragrafo precedente: restrizione del tempo libero e carico fisico.
Costruiamo la model class con tutte le combinazioni dei possibili X: modello nullo, con un predittore, con due predittori indipendenti (modello additivo), con due predittori e la loro **interazione**.

```
nullo<-lm(a$depressione~1)
restrizione<-lm(a$depressione~a$restrizione_tempo)
fisico<-lm(a$depressione~a$fisico)
restrizione_fisico<-lm(a$depressione~a$restrizione_tempo+a$fisico)
restrizione_per_fisico<-lm(a$depressione~a$restrizione_tempo*a$fisico)
```

Notate che non tutti i modelli di questo set sono nidificati in modelli di ordine superiore (restrizione e fisico, per esempio, non sono *nested* l'uno nell'altro), ma li confronteremo lo stesso.
Inseriamo tutti i modelli come argomenti di `model.sel`; l'ordine è indifferente:

```
model.sel(nullo, restrizione, fisico, restrizione_fisico, restrizione_per_fisico)
```

Model selection table	*Informazioni sui coefficienti dei modelli*					*Informazioni sul fit*			
	b_0 (Int)	b_1 restrizione a$rst	b_1 fisico a$fsc	b_1 interazione a$fsc:a$rst	df	logLik	AICc	delta	weight
restrizione_fisico	10.510	-0.5074	1.821		4	-134.837	**278.8**	**0.00**	**0.573**
restrizione_per_fisico	14.860	-0.8603	0.840	0.06879	5	-134.378	280.5	1.70	0.245
fisico	6.649		1.446		3	-137.220	281.1	2.29	0.182
nullo	17.020				2	-147.111	298.5	19.73	0.000
restrizione	14.770	0.1745			3	-146.886	300.4	21.62	0.000

La tabella presenta i **modelli in ordine di fit decrescente**: il modello migliore è `restrizione_fisico`. Nelle colonne da (`Int`) a `df` troviamo informazioni sul modello: le stime di $\boldsymbol{b_0}$ (`Intrc`), e $\boldsymbol{b_1}$**b** (indicato nell'intestazione con il nome del predittore, nei modelli che lo prevedono) e i **gradi di libertà del modello** (`df`), che corrispondono al **numero di parametri liberi,** ovvero calcolati dai dati. Sono $df = 2$ nel modello nullo (b_0 e varianza di e_i,), $df = 3$ nei due modelli con un solo b_1, $df = 4$ nel modello additivo (a b_0 ed e_i si aggiunge un b_1 per ogni X) e $df = 5$ nel modello con interazione (un b_1 per ogni X più il b_1 dell'effetto di interazione).
Nella sezione successiva, troviamo gli **indicatori di fit**: la log-likelihood ($-2LL$: ricordiamo che corrisponde a SS_R, quindi meno è, meglio è) e il coefficiente $AICc$ presentati sopra.
`delta` indica la **differenza tra il modello migliore** (quello con AIC più basso), e **ciascuno degli altri**: una sua piccola elaborazione fornisce informazioni utili. Infatti, l'**esponziale di $-.5 \times delta$** [$exp(-.5 \times \Delta)$] è la **verosimiglianza relativa** (***relative likelihood***) del modello, una quantificazione della **plausibilità di ogni modello di essere la migliore** approssimazione al modello "vero", alla luce ai dati.

La r*elative likelihood* può essere **normalizzata**, diventando l'**Akaike's weight** (w_i, l'ultima colonna della tabella): ciascuna *relative likelihood* viene divisa per la somma delle *relative likelihood* di tutti i modelli a confronto:

$$w_i = \frac{exp(-.5 \times \Delta)}{\sum_{rl=1}^{RL} exp(-.5 \times \Delta)}$$

```
delta<-c(0,1.7, 2.29,19.73, 21.62)
relative_likelihood<-exp(-.5*delta)
denominatore<-sum(relative_likelihood)
weight<-relative_likelihood/denominatore
round(weight,3)
[1] 0.573 0.245 0.182 0.000 0.000
```

Il *weight* di ogni modello può essere interpretato come la **probabilità che il modello in oggetto sia il migliore, alla luce dei dati e dei modelli con cui è confrontato**. Nel nostro esempio, il modello `restrizione+fisico` ha una probabilità di essere il modello migliore oltre due volte più grande della probabilità di essere il migliore attribuita al modello `restrizione*fisico` ($.573/.245 = 2.34$), e una probabilità di essere il migliore oltre tre volte più grande di quella del modello che comprende solo il predittore `fisico` ($.573/.182 = 3.15$). Potremmo quindi ragionevolmente fidarci di aver fatto la scelta migliore, **tra quelle disponibili**, individuando il modello additivo come più adatto per rappresentare la realtà.

Inoltre, grazie ai w_i, è possibile **determinare un confidence set** dei modelli candidati, il cui significato è analogo a quello dei *confidence intervals* attorno a una statistica campionaria: avremo quindi il modello migliore secondo $AICc$ e un **set di modelli alternativi e ragionevolmente plausibili** in popolazione. Una regola pratica[95] consiglia di **includere nel confidence set i modelli il cui w_i sia $\geq 10\%$ del w_i più grande**. Nell'esempio, questo includerebbe nel confidence set, oltre al migliore, tutti i modelli con una plausibilità (w_i) maggiore o uguale al 10% di $.573$, ovvero con un $w_i \geq .573 \times .10 \geq .057$: il modello con interazione e il modello con il solo predittore `fisico` hanno un $w_i > .057$, quindi rientrano nel confidence set, da cui sono esclusi gli altri due.

Infine, i w_i possono essere usati anche per **stimare l'importanza relativa dei singoli b_1**: basta sommare, per ogni predittore, i w_i di ogni modello che lo contenga. Nell'esempio:

b_1	*Restrizione + Fisico* $w_i = .573$	*Restrizione × Fisico* $w_i = .245$	*Solo Fisico* $w_i = .182$	*Solo Restrizione* $w_i = .0$	**Importanza relativa**
Fisico	.573	.282	.182	--	`.573+.282+.182` `[1] 1.037`
Restrizione	.573	.282	--	.0	`.573+.282+0` `[1] 0.855`

Il carico fisico dell'assistenza è un predittore della depressione $1.037/.855 =$ **1.21 volte più plausibile** della restrizione del tempo libero, **alla luce dei dati e dei modelli costruiti**.

Guardiamo ora i parametri dei modelli: b_0 e b_1 variano a seconda del modello, e quindi potrebbe venire la curiosità di definire **quale sia il vero valore** di Y per $X = 0$ e della variazione unitaria in Y per unità di X, in popolazione. Naturalmente, si può prendere la decisione *tranchant* di considerare solo i parametri definiti dal modello migliore, ma quando il *confidence set* dei modelli comprende plurimi modelli ragionevolmente verosimili, questa decisione potrebbe essere inutilmente miope. Si potrebbe allora fare una semplice media dei parametri dei modelli (almeno quelli verosimili), ma in questo elimineremmo del tutto l'informazione sulla verosimiglianza del modello - e sappiamo che, comunque, alcuni sono migliori di altri. Possiamo allora usare **l'informazione di w_i per ponderare la stima dei parametri** e dei loro errori standard e combinarli nei cosiddetti ***averaged parameters*.** Non sarà richiesto all'esame, ma nella vita può sempre servire: trovate le indicazioni per il calcolo degli *averaged parameters* e relative funzioni nell'**Appendice IV**.

[95] Royall (1997) suggerisce un cut off minimo di 8 (1/8) per valutare la forza dell'evidenza

11.5 Verifica dei casi influenti e dei prerequisiti del modello lineare multiplo

Anche per la regressione multipla valgono le cautele relative all'influenzabilità del modello da parte di casi influenti e alla generalizzabilità del modello che abbiamo visto per la regressione semplice.
Outliers multivariati e casi con alto valore di leverage si rilevano come visto nel §9.2; analogamente, i requisiti per una regressione lineare semplice si ritrovano in una regressione lineare multipla:

Requisiti relativi alla relazione tra le variabili	Requisiti relativi agli errori
Y: metrica, continua e non tronca; X: quantitative o categoriali	Normalità
Indipendenza dei valori Y	Omoschedasticità
Relazione assente tra le X e altre covariate esterne al modello	Indipendenza
Relazione multivariata lineare tra Y e le X	

Si aggiunge però un ulteriore, importante requisito relativo alle **relazioni tra i predittori**: il modello lineare multiplo richiede **assenza di multicollinearità**. La **perfetta collinearità** si ha quando almeno una X è una perfetta **combinazione lineare** delle altre X, cioè è una funzione lineare degli altri predittori: $X_1 = \alpha X_2$. **Alti livelli di collinearità** non sono esattamente corrispondenti a correlazioni elevate tra i predittori, anche se può essere facile concettualizzarli così: quello che importa è **l'associazione** tra una o più X, **condizionata alle altre variabili X** nel modello. In pratica, si hanno problemi di multicollinearità quando, una volta noto l'effetto di una X su Y, conoscere l'effetto di un altro predittore aggiunge poco al fit del modello, cioè al suo potere predittivo. Un esempio tipico in psicometria è l'inserimento in un modello delle sottoscale (X_1 e X_2) e del loro punteggio totale (X_3) di un test: conoscere il punteggio totale, note le sottoscale, non aggiunge nulla al fit del modello. Ad ogni modo, predittori multicollineari danno problemi perché **aumentano gli SE dei b_1,** che quindi sono **più variabili** tra i campioni: questo rende i modelli **meno affidabili rispetto alla popolazione**, **allarga i CI** e **allontana la probabilità di rifiutare l'**H_0 relativa ai b_1 (ricordate le formule...). Inoltre, **complicano la comprensione dell'apporto di ogni X al modello**: se X_1 condivide molta varianza (cioè è molto correlata) con le altre, il suo contributo individuale **unico** alla predizione di Y sarà limitato; se invece la sua varianza condivisa è scarsa, il suo contributo unico alla variazione di Y sarà chiaramente interpretabile. Quando i predittori non sono correlati, il disegno è **ortogonale**: la variabilità attribuita a un dato predittore è costante, ovvero indipendente dall'ordine in cui i fattori sono inseriti o rimossi dal modello. Al contrario, in presenza di multicollinearità, cioè in disegni **non ortogonali**, la porzione di variabilità di Y attribuita a una X dipende dall'ordine in cui sono inseriti. Come vedremo in dettaglio nel capitolo 13, in cui affronteremo il problema della multicollinearità con X categoriali, questo problema porta alla scelta fra diversi tipi di partizione della SS_M tra i predittori per ovviare al problema; per fortuna, la funzione `lm` adotta di default il tipo di partizione (partizione di tipo III) più prudente nel caso di disegni non ortogonali, per cui potremo rimandare il problema tra un paio di capitoli.
Intanto, per quantificare la multicollinearità possiamo usare due statistiche. La **Tolleranza (T_1) varia da 0 a 1** e indica quanta **proporzione di varianza di una X non è spiegata** dalle altre X. Ci auguriamo perciò **valori alti**: se ci sono X con $T_1 < .2$ **o** $T_2 < .1$, il modello ha problemi di multicollinearità. il **variance influence factor (VIF)** equivale a $1/T_1$; quindi, dovrebbe essere **il più basso possibile**: indicativamente, valori $VIF \geq 10$ dovrebbero essere ritenuti allarmanti, perché segnale di una forte correlazione la X e gli altri i predittori. In R, possiamo usare usiamo `vif(modello)` del package `car`, oppure `VIF(modello)` di `DescTools`, o ancora `check_collinearity(modello)` di `performance`, che nell'output suggerisce anche il livello del problema nel modello:

```
vif(depressione_caregiver)
a$restrizione_tempo            a$fisico
           1.382045            1.382045


1/vif(depressione_caregiver)
a$restrizione_tempo            a$fisico
          0.7235654           0.7235654.03

VIF(depressione_caregiver)
a$CBI_burden_restrizione_tempo
                      1.382045
           a$CBI_burden_fisico
                      1.382045

check_collinearity(depression_caregiver)
# Check for Multicollinearity
Low Correlation
           Parameter  VIF Increased SE
 a$restrizione_tempo 1.38         1.18
            a$fisico 1.38         1.18
```

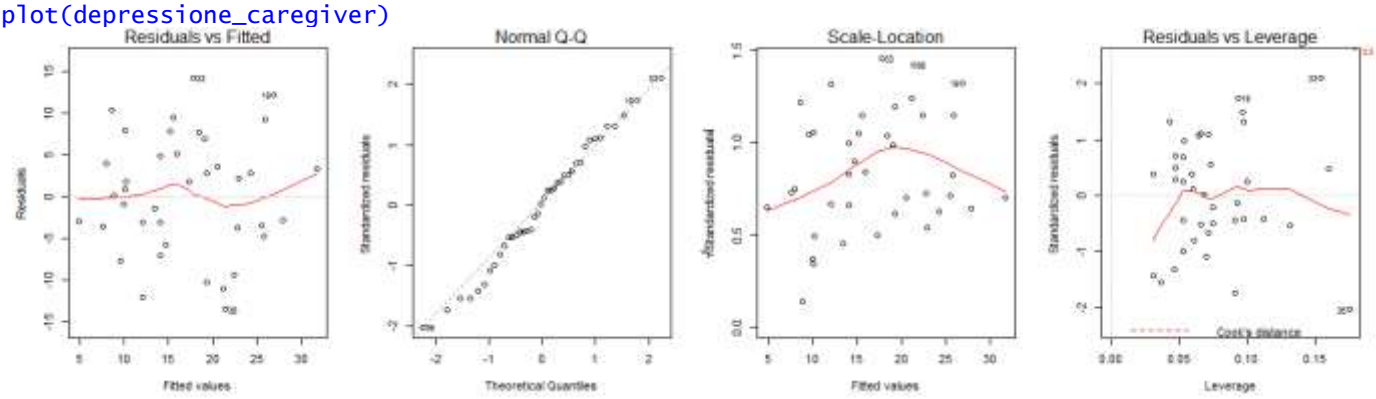
I predittori non presentano problemi di multicollinearità: secondo T_1, il 72.4% della varianza della restrizione non risente dell'effetto del carico fisico, e il 72.4% (ovviamente) della varianza del carico fisico è indipendente dalla restrizione del tempo libero.

Vediamo stabilità e generalizzabilità del modello `depressione~restrizione+fisico`, risultato migliore secondo la procedura per passi.

```
depressione_caregiver<-lm(a$depressione~a$restrizione_tempo+a$fisico)
```

Prima i plot diagnostici:

```
par(mfrow = c(1,4))
plot(depressione_caregiver)
```



A prima vista, non sembrano esserci problemi di normalità e omoschedasticità degli errori né di linearità della relazione, né casi influenti (i pochi casi con residui standardizzati prossimi a |2| hanno una distanza di Cook <1.

Non fidiamoci degli occhi e verifichiamo:

```
shapiro.test(residuals(depressione_caregiver))
       Shapiro-Wilk normality test
data:  residuals(depressione_caregiver)
W = 0.9792, p-value = 0.66

t.test(residuals(depressione_caregiver))
       One Sample t-test
data:  residuals(depressione_caregiver)
t = -2.6765e-17, df = 39, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -2.281047  2.281047
sample estimates:
    mean of x
-3.018419e-17
```

```
dwt(depressione_caregiver)
 lag Autocorrelation D-W Statistic p-value
   1      -0.1164792      2.215425   0.388
 Alternative hypothesis: rho != 0

bptest(depressione_caregiver, studentize=FALSE)
        Breusch-Pagan test
data: depressione_caregiver
BP = 0.63212, df = 1, p-value = 0.4266
```

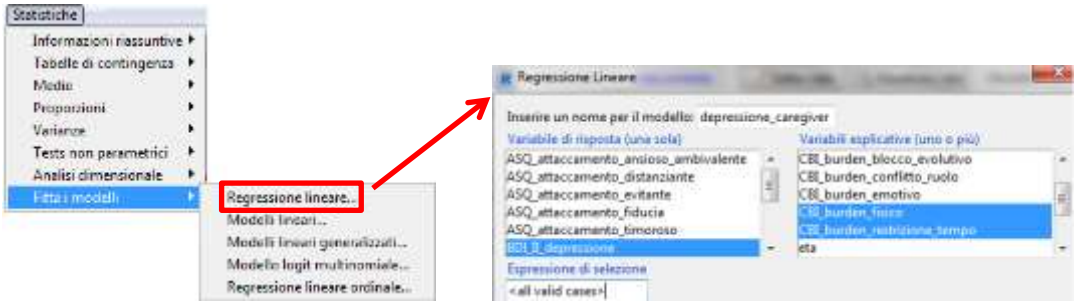
I requisiti sugli errori si confermano pienamente rispettati.

```
summary(rstandard(depressione_caregiver))
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.02000 -0.56880  0.06733  0.00146  0.69070  2.11000

summary(cooks.distance(depressione_caregiver))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
8.850e-06 3.830e-03 1.293e-02 3.315e-02 2.925e-02 2.895e-01
```

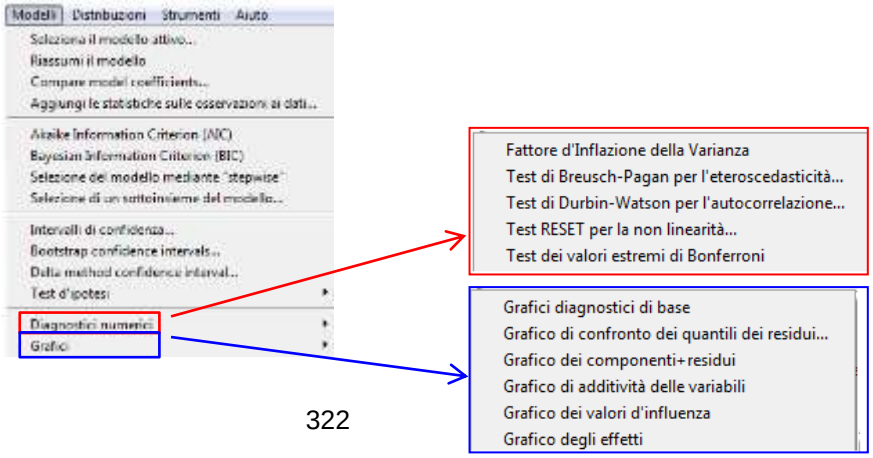
Ci sono soggetti probabilmente outliers bivariati, ma nessun caso con preoccupante valore di leverage, quindi nessun caso influente.

- *Riprendete il dataframe* ***cuccioli*** *usato nel capitolo 3: verificate se* ***l'interazione positiva*** *dei cuccioli con gli osservatori è* ***predetta*** *dalla loro personalità* ***amichevole*** *ed* ***estroversa****. Costruite il modello, completate e interpretate l'output, verificate tutti i prerequisiti e la sua stabilità.*
- *Verificate* ***quali****, tra* ***tutti i comportamenti osservati****, determinano il giudizio sul* ***nevroticismo*** *dei cuccioli degli osservatori. Individuate e costruite il modello migliore, interpretatene l'output, verificate tutti i prerequisiti e la sua stabilità.*

Se volete fare una regressione multipla con `RCommander`, prima di tutto dovete creare il modello:



Questa operazione produce l'oggetto – modello e genera il suo `summary`. Nel menu `Modelli`, ora attivo, potete trovare (quasi) tutto quello che vi può servire: diagnostiche sui casi, CI, grafici, inserimento per passi...



Capitolo 12
Regressione con un predittore categoriale a più di due livelli

*In questo capitolo useremo il dataframe **coppie** (rinominato **cp**) e il dataframe **epde** pubblicati su Elly: apriteli e leggetene la descrizione prima di proseguire*

Dovrebbe essere sufficientemente chiaro come trattare il **confronto tra due medie** adattando un modello lineare (capitolo 10). In questo capitolo vedremo invece come adattare un modello lineare al caso in cui le **medie** da confrontare siano **più di due**, derivanti dai livelli $k > 2$ di un solo predittore X applicato a una sola Y (almeno ordinale).
Avendo, ad esempio, tre medie corrispondenti a tre livelli indipendenti a, b, c di una X, potremmo essere tentati di metterle a confronto con tre confronti a coppie **indipendenti**: X_a *versus* X_b, X_a *versus* X_c, X_b*versus* X_c. Così facendo, però, incorreremmo nel ***family-wise error rate*** già visto nel §8.4.1: **l'incremento della probabilità di commettere un errore di I tipo in una qualsiasi famiglia di test** quando H_0 è assunta per vera in ciascun caso. Nel capitolo 8 la famiglia di test si riferiva a una matrice di coefficienti di correlazione, qui è costituita da una serie di test a coppie (t-test o test non parametrici, per campioni indipendenti o dati appaiati). Resta il problema che per i tre confronti a coppie ipotizzati la probabilità overall di non incorrere in almeno un errore di I tipo nell'intero set di confronti è uguale a:

```
.95*.95*.95
[1] 0.857375
```

e quindi quella di farne almeno uno è:

```
1.-.857375
[1] 0.142625
```

Per questo motivo, preferiamo creare un unico modello lineare *overall* per descrivere la relazione di predizione di X su Y; questo tipo di analisi si definisce **analisi della varianza** (**ANOVA**) quando il predittore è categoriale: come nei precedenti modelli lineari, stabiliremo la significatività del predittore dal **rapporto** $\boldsymbol{F}$ tra la MS_M e la MS_R. Però, potremo comunque avere bisogno di confrontare i livelli tra loro, ad esempio se il modello *overall* risulta significativo (quindi se i diversi livelli di X dipingono una diversa conformazione dei dati), oppure se l'ipotesi sperimentale è specificamente legata alla differenza tra solo alcuni livelli di X: ovvieremo, allora, al *family-wise error rate* in due modi:

a) **Pianificando nel modello overall solo** i **confronti strettamente necessari** all'ipotesi (§§12.1.1 - 12.1.2): **contrasti a priori** A.K.A. pianificati o **pesati** (ponderati, *weighted*). I contrasti a priori possono essere **semplici**, cioè applicati a coppie di singole medie (X_a *versus* X_b), o **multipli** (A.K.A. complessi), cioè applicati alle medie di coppie di livelli accorpati (X_a *versus* X_{b-c}, X_{a-b} *versus* X_{c-d}, ecc.). Se si vuole mantenere costante l'α prescelto ed evitare il family-wise error, con k gruppi ($df_M = k - 1$) si devono fare **solo $k - 1$** contrasti a priori, nonostante il numero teorico di confronti a coppie possibili sia decisamente più alto: per una X a 3 livelli, ad esempio, i confronti a priori che rispettano l'indipendenza dei $p - value$ (**ortogonali**) sono solo 2 $\rightarrow k - 1$, mentre i confronti a coppie tra i livelli, semplici o complessi, sono molti di più (X_a *vs* X_b, X_a *vs* X_c, X_b *vs* X_c , X_a vs X_{bc}, X_b *vs* X_{ac}, X_{ac} *vs* X_b). I contrasti sono ortogonali quando il risultato di un contrasto non fornisce informazioni sull'esito degli altri: così impostati, sono **più potenti** dei confronti post hoc. Ripasseremo l'ortogonalità, vista nella correlazione, approfondendo i contrasti.

b) **Conducendo confronti post – hoc** (solo se l'effetto nel modello *overall* è significativo) tra tutti i livelli a coppie, **correggendo l'alfa** in ragione del numero di confronti o secondo altri criteri, più articolati, che vedremo nel §12.6.3. A differenza dei contrasti ortogonali, i confronti post hoc non sono indipendenti: ad esempio, se il confronto $X_a > X_b$ è significativo, e se $X_b > X_c$ è significativo, allora sarà significativo anche $X_A > X_c$.

Come nel capitolo 10, suddivideremo l'esposizione secondo la natura del disegno (campioni indipendenti e misure ripetute) e secondo la natura di Y (test parametrici e non parametrici). Cominciamo con i disegni between groups.

12.1 Test per disegni between groups: ANOVA per un solo predittore a più di due livelli

La tabella "**tradizionale**" dell'ANOVA per una X a più di due livelli è del tutto analoga a quella di ANOVA per una X a due livelli: SS_M e SS_R, i loro df, MS_M e MS_R, F e $p - value$, che esprime la significatività dell'effetto di X su Y: è l'output mostrato da `aov(Y~X)`.

Vediamolo con un esempio: nel dataframe **coppie** (→ **cp**) sono raccolti i dati relativi a ***offenders*** (solo **uomini**, in carcere per crimini estremamente violenti commessi contro la partner), **vittime** (solo **donne**, seguite da Centri antiviolenza in quanto oggetto di atti violenti/persecutori da parte di un partner) e **controlli** (**uomini e donne**, reclutati tra persone con relazioni di coppia, stabili da almeno tre anni, in cui non si sono verificate interazioni violente tra i partner). I predittori sono gruppo, genere, età (variabile continua) e livello di istruzione. Le Y sono molte, relative a dimensioni di personalità, stile di attaccamento, dipendenza affettiva, capacità empatica, schemi o stili di coping disfunzionali. Scegliamo **come Y il disturbo di personalità schizoide** (test *Millon Clinical Multiaxial Test*; espressi in punti T, sono clinicamente rilevanti punteggi ≥ 65), e come predittore il **gruppo** ($X_{a\ Controlli}$, $X_{b\ Offenders}$, $X_{c\ Vittime}$): indipendentemente da tutti gli altri predittori possibili, l'appartenenza all'uno o all'altro gruppo determina una diversa elevazione del disturbo di personalità schizoide? Rinomiamo, per brevità, la Y in `cp$schizoide` e vediamo cosa risponde la funzione `aov(Y~X)`:

```
cp$schizoide<- cp$millon_personalita_schizoide
summary(aov(cp$schizoide~o$gruppo))
             Df    sum Sq   Mean Sq   F value     Pr(>F)
cp$gruppo     2     10851      5426     10.42   0.000108
Residuals    71     36974       521
```

Sì, il gruppo ha un effetto significativo sull'elevazione della personalità schizoide. L'output non fornisce direttamente R^2, ma contiene gli elementi per calcolarlo dalle SS, e, un po' più faticosamente, anche quelli per calcolare R^2_{adj}: $R^2_{adj} = 1 - (1 - R^2)\frac{N-1}{df_{errore}}$:

```
round(10851/(10851+36974),3)
[1] 0.227
1-(1-.227)*(73/71)
[1] 0.2051119
```

Il gruppo spiega il 22.7% della variabilità della personalità schizoide nel campione, generalizzabile al 20.5% in popolazione. Alternativamente, possiamo usare **`EtaSq(aov(formula))`** di **`DescTools`** (ci potrà servire anche per i test post hoc), che fornisce il coefficiente **eta quadrato** (η^2), *alias* R^2:

```
EtaSq(aov(cp$schizoide~cp$gruppo))
              eta.sq  eta.sq.part[96]
```

[96] Eta quadrato parziale (η^2_p ovvero R^2_p): lo ritroveremo nell'ANOVA fattoriale, con più di una X. È un **coefficiente di intensità dell'effetto relativo**, che **esprime l'impatto di un predittore X su Y rispetto a quello degli altri predittori nel modello**. Con una sola X, ovviamente, coincide con η^2.

```
cp$gruppo 0.2268924    0.2268924
```
La stessa funzione, aggiungendo l'argomento **anova= TRUE** produce contemporaneamente anche l'output di aov:

```
EtaSq(aov(cp$schizoide~cp$gruppo), anova= TRUE)
              eta.sq  eta.sq.part        SS  df         MS          F            p
cp$gruppo 0.2268924    0.2268924  10851.15   2  5425.5741  10.41857  0.0001077752
Residuals 0.7731076           NA  36973.95  71   520.7598         NA           NA
```

Questa sintesi fornita da aov "nasconde" il lavoro sui contrasti a priori all'opera nell'ANOVA, esplicitato invece nella funzione lm: vediamola applicata a questo modello, e procediamo a spiegarne la logica.

```
summary(lm(cp$schizoide~cp$gruppo))
Residuals:
   Min     1Q Median     3Q    Max
-51.19 -12.21  -1.00  17.29  42.79

Coefficients:
                  Estimate Std. Error t value  Pr(>|t|)
(Intercept)         42.214      3.521  11.98    < 2e-16
cp$gruppooffender   30.598      6.704   4.564  2.05e-05
cp$gruppovittima     8.973      6.704   1.338     0.185

Residual standard error: 22.82 on 71 degrees of freedom
Multiple R-squared:  0.2269,     Adjusted R-squared: 0.2051
F-statistic: 10.42 on 2 and 71 DF,  p-value: 0.0001078
```

Ritroviamo R^2_M **multiplo**, dato dal rapporto tra SS_M (somma delle differenze tra le medie dei gruppi e la *grand mean*, moltiplicate per la N di ogni gruppo) e la SS_T (somma delle differenze tra ogni y_{ij} e la *grand mean*): nel complesso, il gruppo di appartenenza spiega il 22.7% di variabilità dei punteggi di personalità schizoide nel campione, che cala al 20.5% in popolazione. Il *residual standard error* è sempre la radice quadrata della SS_R (somma delle differenze tra ogni y_{ij} e media del gruppo j di appartenenza). F è naturalmente dato dal rapporto tra MS_M ($SS_M/(k-1)$, dove k è il numero di contrasti - b_1) e MS_R ($SS_R/(N-k-1)$).
L'intercetta b_0 rappresenta al solito il valore di Y quando $X = 0$, quindi la media del gruppo di riferimento (il primo in ordine alfabetico): il gruppo dei soggetti di controllo ha un punteggio medio = 42.2 nella scala del disturbo schizoide. I b_1 **rappresentano i due contrasti** che ora andremo ad approfondire: controlli *versus* offenders (primo b_1) e controlli *versus* vittime (secondo b_1).

Nei contrasti a priori, le **medie dei livelli interessati** sono **moltiplicate per pesi** (**coefficienti**), differenti da confronto a confronto, e successivamente **sommate**: la **somma pesata** (*weighted*) così ottenuta è **confrontata con la somma prevista** per quel confronto **dall'ipotesi nulla**, secondo cui la **somma pesata è uguale a zero**: $H_0 = \bar{X}_{w_a} + \bar{X}_{w_b} = 0$.
Questa H_0 è verificata con test F o t-test (R usa t-test, altri software no), confronto per confronto. In questo metodo, i contrasti sono pianificati secondo le ipotesi che hanno guidato l'impostazione della ricerca, e che potrebbero non essere interessate a tutti i confronti tra tutti i livelli, o all'effetto complessivo di X su Y, ma solo ad alcuni di questi confronti. Si possono, quindi, condurre i contrasti a priori limitatamente ad alcuni livelli, **invece di** fare il modello lineare - ANOVA, oppure impostare i contrasti desiderati per la variabile X e poi ricavare anche il modello complessivo, dal quale avremo più informazioni. Il tipo di contrasti impostati non ha alcun effetto sul rapporto tra MS_M e MS_R, e quindi sull'effetto principale di X, dato che i **contrasti producono diverse ripartizioni della sola** MS_M, lasciando intatta la quota di errore del modello.
Vediamo prima i contrasti a priori semplici, che verificano la differenza tra le medie di due livelli, poi quelli multipli, in cui almeno una parte del contrasto è costituita dalla media delle medie di più livelli accorpati.

12.1.1 Contrasti a priori semplici

L'ipotesi nulla che guida un contrasto a priori semplice è che la **somma della media del livello X_a e della media del livello X_b, entrambe opportunamente pesate, sia uguale a 0**; gli **altri gruppi** presenti in X sono **ignorati** nel contrasto. Facciamo un esempio con una X a tre livelli (X_a, X_b, X_c): il primo contrasto vuole verificare la differenza tra la media del gruppo X_a e la media del gruppo X_b; perciò, l'ipotesi nulla è che $\bar{X}_a = \bar{X}_b \rightarrow \bar{\boldsymbol{X}}_{\boldsymbol{wa}} + \bar{\boldsymbol{X}}_{\boldsymbol{wb}} = \mathbf{0}$.

Costruiamo la **variabile di contrasto C_1**, in cui la somma delle medie di tutti i livelli, compresi quelli non interessati da H_0 (X_c) corrisponde a quanto previsto da X_0: per far questo, assegniamo gli **opportuni pesi** alle medie dei livelli. Da $\bar{X}_a = \bar{X}_b$ avremo quindi:

$$C_1 \rightarrow H_0: -\mathbf{1} * \bar{\boldsymbol{X}}_{\boldsymbol{a}} + \mathbf{1} * \bar{\boldsymbol{X}}_{\boldsymbol{b}} + \mathbf{0} * \bar{\boldsymbol{X}}_{\boldsymbol{c}} = \mathbf{0},$$

oppure, in modo del tutto equivalente:

$$C_1 \rightarrow H_0: 1 * \bar{X}_a + (-1) * \bar{X}_b + 0 * \bar{X}_c = 0.$$

Abbiamo quindi assegnato segno diverso alle medie che intendiamo confrontare, 0 al gruppo che viene ignorato nel contrasto, e la somma dei pesi assegnati nel contrasto è $= \mathbf{0}$.

Se nel secondo contrasto si vuole verificare la differenza tra X_a e X_C, per cui H_0: $\bar{X}_a = \bar{X}_c$ → $\bar{\boldsymbol{X}}_{\boldsymbol{wa}} + \bar{\boldsymbol{X}}_{\boldsymbol{wc}} = \mathbf{0}$, si imposta la **seconda variabile di contrasto C_2**, seguendo le stesse regole:

$$\boldsymbol{C_2} \rightarrow H_0: -\mathbf{1} * \bar{\boldsymbol{X}}_{\boldsymbol{a}} + \mathbf{0} * \bar{\boldsymbol{X}}_{\boldsymbol{b}} + \mathbf{1} * \bar{\boldsymbol{X}}_{\boldsymbol{c}} = \mathbf{0}, \text{ oppure } \boldsymbol{C_2} \rightarrow H_0: +\mathbf{1} * \bar{\boldsymbol{X}}_{\boldsymbol{a}} + \mathbf{0} * \bar{\boldsymbol{X}}_{\boldsymbol{b}} + (-\mathbf{1}) * \bar{\boldsymbol{X}}_{\boldsymbol{c}} = \mathbf{0}$$

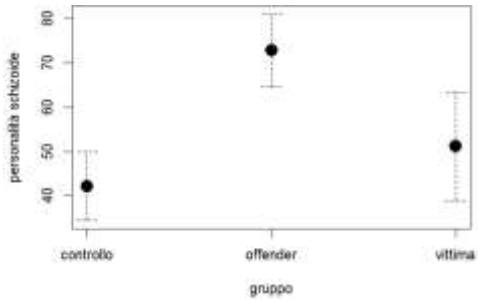
Una volta costruite le variabili di contrasto, per ciascuna di esse la **differenza tra le medie pesate** viene testata contro l'ipotesi nulla che sia $= 0$, con un test inferenziale che usa la distribuzione t (così in R) o la distribuzione F.

Riprendiamo l'esempio dell'effetto del gruppo sull'elevazione del profilo schizoide e vediamo come si è arrivati all'output del `lm` di qualche pagina fa: in particolare, siamo interessati ai **confronti tra il gruppo di controllo e ciascuno dei gruppi clinici**, per cui abbiamo **due ipotesi nulle**: $\bar{X}_a = \bar{X}_b$ e $\bar{X}_a = \bar{X}_c$.

Vediamo le medie dei gruppi:

```
tapply(cp$schizoide, cp$gruppo, mean)
controllo  offender   vittima
 42.21429  72.81250  51.18750

tapply(cp$schizoide, cp$gruppo, sd)
controllo  offender   vittima
 24.91110  15.36758  23.07732
```

Diremmo che il confronto si annuncia significativo tra controlli e offenders, e non significativo tra controlli e vittime.



Per lavorare con tutti i decimali delle **medie** ed evitare errori di trascrizione nei calcoli successive, salviamole: basta assegnare `tapply` a un oggetto che chiamiamo **`schizo`**, ciascuno dei cui elementi è una media:

```
schizo<-tapply(cp$schizoide, cp$gruppo, mean)
schizo[1];schizo[2]; schizo[3]
controllo
 42.21429
offender
 72.8125
vittima
 51.1875
```

Costruiamo le due variabili di confronto C_1 e C_2, corrispondenti alle H_0; naturalmente, attenzione all'ordine dei livelli dei `factor` in R, che, come sappiamo, è di default alfanumerico. L'ordine dei segni è irrilevante:

✓ **C_1** confronto controlli X_a e offenders X_b → H_0: $-1 * \bar{X}_a + 1 * \bar{X}_b + 0 * \bar{X}_c = 0$

```
contrasto1 <- (-1*schizo[1]) + (1*schizo[2])  + (0*schizo[3])
```

✓ **C_2** confronto controlli X_a e vittime X_c → H_0: $-1 * \bar{X}_a + 0 * \bar{X}_b + 1 * \bar{X}_c = 0$

```
contrasto2 <- (-1*schizo[1]) + (0*schizo[2])  + (1*schizo[3])
```

Le differenze tra le medie dei gruppi così pesate sono:

```
contrasto1
30.59821
contrasto2
8.973214
```

La differenza $C_1 = 30.59821$ tra controlli e offenders è significativamente diversa da 0? La differenza $C_2 = 8.973214$ tra controlli e offenders è significativamente diversa da 0? Possiamo verificarlo con due t-test che, come sempre, hanno al numeratore la differenza pesata tra le medie (C_1 o C_2), e al denominatore l'errore standard, oppure, e più sensatamente, costruendo i CI attorno alla differenza pesata, nuovamente usando l'errore standard.

Con sollievo, R ci esonera, una volta compresa la logica del contrasto, dal calcolare a mano tutto quanto, C_1 e C_2 compresi, perché `lm` gestisce il modello lineare **inserendovi come predittori proprio le variabili di contrasto** così impostate: avremo quindi una **regressione multipla, in cui i predittori sono costituiti dalle variabili di contrasto**, il cui numero deve essere **uguale al numero k di livelli di $X - 1$**. Oltre alle due variabili di contrasto – predittori, con relativo t-test, potremo così avere anche informazioni sul modello complessivo (F, R^2, R^2_{adj}), e calcolare facilmente i CI dei contrasti con `confint(modello)`.

R, **di default**, attribuisce a tutte le variabili di classe `factor` **non ordinato** (*not ordered*, §3.2.1) una tipologia di costruzione di contrasti definita **contrasts treatment**: `contr.treatment(numero di gruppi= , livello di riferimento = )`. In questi contrasti, che sono quelli usati nell'esempio precedente, un gruppo è confrontato con ciascuno degli altri, ed R assegna da sé i pesi a ciascuna media, in modo che la loro somma sia $\sum = 0$ in ogni confronto. Se l'utente non modifica il livello di riferimento, quello di default è il primo gruppo in ordine alfanumerico, come sempre. Possiamo verificare come R è pronto a trattare i contrasti del fattore con `contrasts(fattore)`:

```
contrasts(cp$gruppo)
```

	offender	vittima
controllo	0	0
offender	1	0
vittima	0	1

il primo livello in ordine alfabetico è il gruppo di riferimento in tutti i confronti

I contrasto: *vs* offender — II contrasto: *vs* vittima

Il livello cui è assegnato 0 in tutti i contrasti è il **gruppo dei controlli**. Questo livello viene confrontato **nel primo contrasto con gli offenders [1],** mentre il gruppo delle vittime è ignorato [0]. Nel **secondo contrasto**, il gruppo dei controlli è **confrontato con le vittime [1]**, mentre il gruppo offenders è ignorato [0]. Per ora, diciamo che il gruppo di controllo ci va bene come livello di riferimento e rivediamo le medie dei gruppi:

```
schizo
controllo  offender   vittima
 42.21429  72.81250  51.18750    ← in rosso: b0
schizo[2]-schizo[1]
offender
30.59821                         ← b1 primo contrasto
schizo[3]-schizo[1]
 vittima
8.973214                         ← b1 secondo contrasto
```

Ancora prima di vedere il `summary`, dovremmo poter anticipare che avremo **un modello con una b_0** = media del **gruppo di controllo** (**41.21**) **e due b_1**, che rappresentano rispettivamente la differenza tra le medie dei gruppi nel I contrasto (controllo versus offender: $b_{1_{C1}} = 30.59$) e nel II contrasto (controlli versus vittime: $b_{1_{C2}} = 8.97$)

```
summary(lm(cp$schizoide~cp$gruppo))
Residuals:
   Min     1Q Median     3Q    Max
-51.19 -12.21  -1.00  17.29  42.79

Coefficients:
                  Estimate Std. Error t value  Pr(>|t|)
(Intercept)         42.214      3.521  11.98    < 2e-16
cp$gruppooffender   30.598      6.704   4.564  2.05e-05
cp$gruppovittima     8.973      6.704   1.338     0.185

Residual standard error: 22.82 on 71 degrees of freedom
Multiple R-squared:  0.2269, Adjusted R-squared: 0.2051
F-statistic: 10.42 on 2 and 71 DF,  p-value: 0.0001078

confint(lm(cp$millon_personalita_schizoide~cp$gruppo))
                       2.5 %   97.5 %
(Intercept)       35.193158 49.23541
cp$gruppooffender 17.230384 43.96604
cp$gruppovittima  -4.394616 22.34104
```

Naturalmente, confermiamo le anticipazioni e i valori di C_1 e C_2 prima calcolati. Aggiungiamo che gli errori minimi e massimi sono davvero imponenti (dovremmo controllare la presenza di outlier multivariati), e che la differenza tra controlli e offender (I contrasto) è significativa, anche se il CI in popolazione è davvero molto ampio (la differenza sta tra 17 e 43 punti), mentre la personalità schizoide di controlli e vittime non è significativamente differente (II contrasto).

Un **diverso gruppo di riferimento** rispetto a quello previsto di default è definito assegnando ai contrasti di X la funzione `contr.treatment`, in cui si specifica il nuovo livello di baseline: `contrasts(fattore)<-contr.treatment(n= numero di gruppi, base= nuovo livello di riferimento)`.

Poiché siamo particolarmente interessati al gruppo degli offenders e non al gruppo dei controlli, impostiamo questo livello (X_b) come nuovo gruppo di riferimento:

```
contrasts(cp$gruppo)<-contr.treatment(n = 3, base = 2)
contrasts(cp$gruppo)
```

```
           1  3
controllo  1  0
offender   0  0   -►il livello offender è il gruppo di riferimento in tutti i confronti
vittima    0  1
```

I contrasto: *1 vs* livello 2 — II contrasto: 3 *vs* livello 2

Cioè:

```
schizo[1]-schizo[2]
controllo
-30.59821
schizo[3]-schizo[2]
vittima
-21.625
```

Rifacciamo il modello lineare con i nuovi contrasti:

```
summary(lm(cp$millon_personalita_schizoide~cp$gruppo))
[omissis]
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    72.812      5.705  12.763  < 2e-16
cp$gruppo1    -30.598      6.704  -4.564 2.05e-05
cp$gruppo3    -21.625      8.068  -2.680  0.00914
```

I punteggi di personalità schizoide degli offenders sono significativamente **maggiori** di quelli delle coppie di controllo (I contrasto: 30.6 di differenza; lo sapevamo già) e di quelli delle vittime (II contrasto: 21.6 punti di differenza).

Il **modello overall non è affatto cambiato**:

```
Residual standard error: 22.82 on 71 degrees of freedom
Multiple R-squared:  0.2269,     Adjusted R-squared:  0.2051
F-statistic: 10.42 on 2 and 71 DF,  p-value: 0.0001078
```

Infatti, abbiamo detto che **cambiare i contrasti non modifica affatto la ripartizione tra MS_M e MS_R**, dato che i contrasti riguardano la partizione della sola variabilità attribuita al predittore. Volendolo, possiamo **creare i contrasti** definendoli uno per uno, il che consente anche di assegnare loro nomi più intuibili una volta letti nell'output: facciamolo con i contrasti di questo secondo esempio, mantenendo gli offenders come riferimento:

```
c1_offender_controlli<-c(1,0,0)
c2_offender_vittime<-c(0,0,1)
```

Assegniamoli al fattore gruppo con `cbind` e verifichiamo:

```
contrasts(cp$gruppo)<-cbind(c1_offender_controlli,c2_offender_vittime)
contrasts(cp$gruppo)
          c1_offender_controlli c2_offender_vittime
controllo                     1                   0
offender                      0                   0
vittima                       0                   1
```

```
summary(lm(cp$millon_personalita_schizoide~cp$gruppo))
[omissis]
Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                      72.812      5.705  12.763  < 2e-16
cp$gruppoc1_offender_controlli  -30.598      6.704  -4.564 2.05e-05
cp$gruppoc2_offender_vittime    -21.625      8.068  -2.680  0.00914
```

Oltre a `contr.treatment`, R dispone di altre funzioni per gestire i contrasti semplici: ad esempio, per confrontare **l'ultimo livello con ciascuno degli altri** si assegna ai contrasti di X la funzione **`contr.SAS(n=)`**, in cui `n=` è il numero di **livelli**.

Formalizziamo, prima di procedere oltre.

Qualunque serie di contrasto si consideri, abbiamo costruito un modello lineare **additivo**, con **una intercetta** (**gruppo di riferimento**) e **due coefficienti angolari parziali**, corrispondenti alle **due variabili dummy** / contrasti:

$y_i = (b_0 + b_1X_1 + b_2X_2) + e_i$. Quindi, per un soggetto i del gruppo di controllo X_a: $y_{livello\ \boldsymbol{Xa}} = (b_0 + b_1 \cdot 0 + b_2 \cdot 0) + e_i = b_0 + e_i$, il punteggio è dato da b_0 (media del gruppo di controllo) più l'errore commesso dal modello per il soggetto i.

Usiamo un esempio con il modello:

```
schizo_gruppo<-lm(cp$millon_personalita_schizoide~cp$gruppo)
```

Vediamo qual è il punteggio osservato del soggetto 10 che appartiene al gruppo di controllo, $y_{sogg.10}$:

```
((S10_controlli <- cp[10,29]))
[1] 20
```

Il punteggio del soggetto 10 è dato dall'intercetta del modello più l'errore del modello per il soggetto 10, cioè:

```
schizo_gruppo$coefficients[1]+residuals(schizo_gruppo)[10]
20
```

Oppure, se preferite i calcoli espliciti:

```
schizo_gruppo$coefficients[1]
(Intercept)
   42.21429
residuals(schizo_gruppo)[10]
       10
-22.21429
```

```
42.21429 + (-22.21429)
[1] 20
```

Vediamo un soggetto del gruppo Offender: $y_{livello\ \boldsymbol{Xb}} = (b_0 + \boldsymbol{b_1} \cdot \mathbf{1} + b_2 \cdot 0) + e_i = b_0 + b_1 + e_i$, il cui punteggio è dato da $b_0 + b_1$ del I contrasto (differenza tra le medie del gruppo di controllo e del gruppo X_b) moltiplicato per il valore del livello X_b nella dummy ($X_b = 1$), più l'errore commesso dal modello per il soggetto i.

Prendiamo il soggetto$_{50}$:

```
(S50_offender <- cp[50,12])
[1] 88
```

Il punteggio del $soggetto_{50}$ appartenente al gruppo X_b è:

```
schizo_gruppo$coefficients[1]+schizo_gruppo$coefficients[2]+(residuals(schizo_gruppo)[50])
88
```

O, se vi è più chiaro:

```
schizo_gruppo$coefficients[2]
offenders$gruppooffender
                30.59821
residuals(schizo_gruppo)[50]
     50
15.1875
42.21429+(30.59821*1)+15.1875
[1] 88
```

Infine, il punteggio di un soggetto del gruppo Vittime: $y_{livello\ \boldsymbol{Xc}} = (b_0 + b_1 \cdot 0 + \boldsymbol{b_2} \cdot \mathbf{1}) + e_i = b_0 + b_2 + e_i$ è dato da $b_0 + b_1$ del II contrasto (differenza tra le medie del gruppo di controllo e del gruppo X_c) moltiplicato per il valore del livello X_c nella dummy ($X_c = 1$), più l'errore commesso dal modello per il soggetto i. Per il $soggetto_{60}$:

```
((S60_vittime <- cp[60,12]))
[1] 2
```

abbiamo:

```
schizo_gruppo$coefficients[1]+schizo_gruppo$coefficients[3]+(residuals(schizo_gruppo)[60])
2
```

Cioè:

```
schizo_gruppo$coefficients[3]
offenders$gruppovittima
               8.973214
schizo_gruppo$residuals[60]
      60
-49.1875
42.21429+(8.973214*1)+(-49.1875)
[1] 2.000004
```

$\boldsymbol{b_0}$ è il termine comune ai soggetti di tutto il modello: pertanto, viene definita anche **costante**.

Concludiamo riprendendo il concetto di ortogonalità visto nel capitolo 8, perché **i vettori dei contrasti semplici non sono ortogonali**, il che comporta che l'esito di uno non è indipendente dal risultato degli altri. I vettori **ortogonali**, ricordiamo, sono vettori **linearmente indipendenti** che formano un angolo **retto** ($cos_{XY} = 0$); abbiamo visto la loro rappresentazione grafica bidimensionale, e verificato che possiamo verificare l'ortogonalità usando il ***dot product*** dei vettori: **moltiplichiamo ogni elemento del primo vettore per il corrispettivo del secondo** e facciamone la somma:

- se =0, i vettori sono ortogonali,
- se ≠ 0, i vettori non sono ortogonali.

Per i primi contrasti costruiti:

- $C_1 \rightarrow H_0$: $\mathbf{-1 * \bar{X}_a + 1 * \bar{X}_b + 0 * \bar{X}_c = 0}$ e $C_2 \rightarrow H_0$: $\mathbf{-1 * \bar{X}_a + 0 * \bar{X}_b + 1 * \bar{X}_c = 0}$. Quindi, abbiamo: $C_1[-1, 1, 0]$ (controlli versus offender) e $C_2[-1,\ 0, 1]$ (controlli versus vittime). Il *dot product* di questi due contrasti è: $\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = (-1 \times -1) + (1 \times 0) + (0 \times 1) = 1 + 0 + 0 = \mathbf{1} \rightarrow$ **i contrasti non sono ortogonali.**

Anche se la somma dei pesi in ogni contrasto è $\sum = 0$, la somma dei prodotti dei pesi nei due contrasti **non è** $\boldsymbol{\sum = 0}$. L'indipendenza è violata indipendentemente dal gruppo scelto come riferimento: infatti, nei due contrasti in cui abbiamo posto gli offenders come riferimento troviamo:

- $C_1 \rightarrow H_0$: $\mathbf{-1 * \bar{X}_a + 1 * \bar{X}_b + 0 * \bar{X}_c = 0}$ e $C_2 \rightarrow H_0$: $\mathbf{0 * \bar{X}_a + (-1) * \bar{X}_b + 1 * \bar{X}_c = 0}$. Quindi, abbiamo: $C_1[-1, 1, 0,]$ (controlli versus offender) e $C_2[0,\ -1, 1]$ (vittime versus offenders). Il *dot product* di questi due contrasti è: $\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = (-1 \times 0) + (1 \times -1) + (0 \times 1) = 0 - 1 + 0 = \mathbf{-1} \rightarrow$ **i contrasti non sono ortogonali.**

Ricordate? Quando i vettori grezzi **non** sono **ortogonali**, i vettori centrati possono essere o non essere perpendicolari, cioè possono essere o non essere correlati. I nostri contrasti sono **non ortogonali e correlati**:

```
c1<-c(-1,1,0)          c1<-c(-1,1,0)
c2<-c(0,-1,1)          c2<-c(-1,0,1)
cor(c1,c2)             cor(c1,c2)
[1] -0.5               [1] 0.5
```

Quando i vettori dei contrasti sono correlati, quindi dipendenti l'uno dall'altro, l'interpretazione dei $p - value$ assegnati ai quantili t dei contrasti non è libera dal *family-wise error rate*: vedremo una possibilità per gestire questo problema con il test Q di Dunnet (§6.1.3), ed esempi di contrasti ortogonali nel prossimo paragrafo.

12.1.2 Contrasti a priori multipli

Come anticipato, i contrasti pianificati **multipli o complessi** consentono di creare **variabili dummy accorpando più livelli,** confrontati con un solo livello o con altri livelli accorpati. Per esempio, si può ipotizzare che l'effetto dell'appartenenza al gruppo sulla **personalità dipendente** sia diverso in chi vive relazioni affettive non gravemente conflittuali (controlli) e in chi ha vissuto, sia pure da prospettive opposte, relazioni intime gravemente disfunzionali. La SS_M sarebbe allora ripartita in:

- $\boldsymbol{contrasto_1}$: X_a controlli ***versus*** $X_b X_c$ disfunzionali (offenders + vittime)
- $\boldsymbol{contrasto_2}$: X_b offenders ***versus*** X_c vittime, ignorando il gruppo X_a.

Oppure, si può ipotizzare che la personalità dipendente sia differente tra chi non ha messo in atto comportamenti francamente antisociali (controlli e vittime) rispetto a chi ha commesso gravi reati contro la persona (offenders):

- $\boldsymbol{contrasto_1}$: $X_a X_c$ (controlli+vittime) ***versus*** X_b offenders
- $\boldsymbol{contrasto_2}$: X_a controlli ***versus*** X_c vittime, ignorando il gruppo X_b offenders

Infine, si può ipotizzare una differenza nella personalità dipendente sia tra i controlli e gli offender, sia tra i controlli e le vittime: è questa è la forma di contrasto semplice che abbiamo già visto:

- $\boldsymbol{contrasto_1}$: X_a ***versus*** X_b, ignorando il gruppo X_c
- $\boldsymbol{contrasto_2}$: X_a ***versus*** X_c, ignorando il gruppo X_b

Nuovamente, in tutti i modelli ipotizzabili, la bipartizione $SS_M - SS_R$ non cambia: i contrasti agiscono sulla sola partizione della SS_M, nella direzione prevista dall'ipotesi a priori del ricercatore.

Come nei contrasti semplici, il loro numero dovrà essere sempre $contrasti = \boldsymbol{k-1}$; si confrontano due "pacchetti" di variabilità per volta: o singoli gruppi a due a due, o un gruppo con altri $\boldsymbol{n}$ gruppi accorpati, o $\boldsymbol{n}$ gruppi accorpati con $\boldsymbol{n}$ gruppi accorpati (meglio non esagerare con gli accorpamenti...). Una volta che un **gruppo è stato isolato in un contrasto**, **non** dovrebbe più essere usato in un **altro contrasto**, perché si **ridurrebbe o annullerebbe l'indipendenza dei contrasti**: dopo aver verificato una differenza significativa tra X_a e X_{bcd} accorpati, è sbagliato confrontare nuovamente X_a con ciascuno dei gruppi X_b, X_c o X_d, o con i gruppi X_{bcd} diversamente accorpati (X_{cd}, X_{bd}, X_{bc}), perché il primo risultato dà informazione sul secondo. Se X_a è il gruppo di controllo e X_{bcd} i gruppi sperimentali, la differenza significativa tra il gruppo di controllo e i tre gruppi sperimentali accorpati rende più probabile che risulti una differenza significativa tra il controllo e i gruppi sperimentali individualmente considerati, o diversamente accorpati.
Ome nei contrasti semplici, dobbiamo **assegnare pesi** (**coefficienti**) di **contrasto** alle medie dei gruppi interessati: per ogni contrasto, i pesi assegnati a ciascun gruppo/i sono **proporzionali** al numero di confronti, e il loro valore discende dall'ipotesi nulla di ogni confronto. Seguiamo il ragionamento con qualche esempio, prima di passare alle scorciatoie, con una X a quattro livelli (X_a, X_b, X_c, X_d), le cui medie sono: $\bar{X}_a = 3$; $\bar{X}_b = 4$; $\bar{X}_c = 6$; $\bar{X}_d = 7$.
Se l'ipotesi da verificare prevede di **accorpare i livelli $\boldsymbol{X_a}$ e $\boldsymbol{X_d}$** per **confrontarli con i livelli $\boldsymbol{X_b}$ e $\boldsymbol{X_c}$ accorpati**, l'H_0 del confronto sarà: $H_0: \bar{X}_a + \bar{X}_d = \bar{X}_b + \bar{X}_c$.
Questa H_0 equivale a dire che la **differenza** tra la media delle medie di X_a e X_{d1} e la media delle medie di X_b e X_c **è uguale a 0** $\rightarrow \boldsymbol{H_0: \frac{\bar{X}_a+\bar{X}_d}{2} - \frac{\bar{X}_b+\bar{X}_c}{2} = 0}$, cioè:

$$\boldsymbol{H_0: +\frac{1}{2} * \bar{X}_a + \frac{1}{2} * \bar{X}_d + \left(-\frac{1}{2}\right) * \bar{X}_b + \left(-\frac{1}{2}\right) * \bar{X}_c = 0}$$

Ecco i quattro coefficienti del contrasto: il gruppo X_a e X_d si vedono assegnare +.5, i gruppi X_b e X_c -.5: **+.5, +.5, -.5, -.5** $\rightarrow$ la somma algebrica corrisponde a 0, cioè il valore previsto da H_0.

```
media_a<-3; media_b<-4; media_c<-6; media_d<-7
media_a+media_d; media_b+media_c
[1] 10
[1] 10
(media_a+media_d)/2; (media_b+media_c)/2
[1] 5
[1] 5
.5*media_a + .5*media_d - .5 * media_b - .5*media_c
[1] 0
```

Evidentemente, nell'esempio il contrasto non sarà significativo.

Un altro contrasto potrebbe voler **confrontare il gruppo X_a con gli altri tre gruppi accorpati**: l'H_0 corrispondente sarebbe $H_0: \bar{X}_a = \bar{X}_b + \bar{X}_c + \bar{X}_d$.
Questa H_0 equivale a dire che la **differenza** tra la media di X_a e la media delle medie di X_b, X_c e X_d **è uguale a 0** $\rightarrow \boldsymbol{H_0: \bar{X}_a - \frac{\bar{X}_b+\bar{X}_c+\bar{X}_d}{3} = 0}$, cioè:

$$H_0: \boldsymbol{+1 * \bar{X}_a + \left(-\frac{1}{3}\right) * \bar{X}_b + \left(-\frac{1}{3}\right) * \bar{X}_c + \left(-\frac{1}{3}\right) * \bar{X}_d = 0}$$

Ecco i quattro coefficienti del contrasto: al gruppo X_a si assegna $+1$, ai gruppi X_b, X_c e X_d si assegna $-.33$: la somma algebrica dei coefficienti corrisponde a zero.

Ultimo esempio: **ignoriamo il gruppo X_a** e confrontiamo il gruppo X_b con gli altri due accorpati, per un'ipotesi nulla uguale a $H_0 = 0 * \bar{X}_a + \bar{X}_b = \bar{X}_c + \bar{X}_d$.

Questa H_0 equivale a dire che, ignorando la media di X_a, la **differenza** tra la media di X_b e la media delle medie di X_c e di X_d **è uguale a 0** $\rightarrow$ $\boldsymbol{H_0}$: $\boldsymbol{0 * \bar{X}_a + 1 * \bar{X}_b - \frac{\bar{X}_c + \bar{X}_d}{2} = 0}$, cioè:

$$H_0: \mathbf{0} * \bar{\boldsymbol{X}}_{\boldsymbol{a}} + \mathbf{1} * \bar{\boldsymbol{X}}_{\boldsymbol{b}} + \left(-\frac{\mathbf{1}}{\mathbf{2}}\right) * \bar{\boldsymbol{X}}_{\boldsymbol{c}} + \left(-\frac{\mathbf{1}}{\mathbf{2}}\right) * \bar{\boldsymbol{X}}_{\boldsymbol{d}} = \mathbf{0}$$

Assegniamo allora coefficiente $= 0$ al gruppo X_a, coefficiente $+1$ al gruppo X_b e coefficienti - .5 a X_c e X_d.

Per velocizzare l'assegnazione dei pesi senza ricorrere alle H_0, nella pratica si ricorre a **numeri interi** e alle poche regole già viste nei contrasti semplici:

- il valore assoluto del coefficiente: i **pesi** assegnati a ciascun gruppo/i della prima parte del confronto dovranno essere **uguali** al numero di gruppi della seconda, e viceversa. Quindi, confrontando X_a con X_b e X_c accorpati, invece di $[+1, -.5, -.5]$ si assegna $X_a = |\mathbf{2}|$, $X_b = |\mathbf{1}|$, $X_c = |\mathbf{1}|$; confrontando X_a con X_b, X_c e X_d accorpati, assegneremo $X_a = |\mathbf{3}|$, $X_b = |\mathbf{1}|$, $X_c = |\mathbf{1}|$, $X_d = |\mathbf{1}|$.
- **il segno $+$ o $-$**: saranno **confrontati** gruppi i cui pesi avranno **segno differente** e saranno **accorpati** gruppi i cui pesi avranno lo **stesso segno**;
- **0**: il gruppo (o i gruppi) il cui peso è impostato = 0 sarà **ignorato** nel contrasto
- $\boldsymbol{\sum = 0}$: naturalmente, la **somma algebrica dei pesi** assegnati per ciascun contrasto dovrà essere $\sum = 0$, **cioè essere corrispondente** alla differenza tra le medie oggetto del contrasto **secondo $\boldsymbol{H_0}$**;

Attenzione ai parametri del modello:

In questi contrasti, $\boldsymbol{b_0}$ non corrisponde più alla media del gruppo di riferimento, come nei contrasti semplici, ma alla **media delle medie dei gruppi**, cioè alla *grand mean*[97].

$$b_0 = \frac{\bar{X}_a + \bar{X}_b + \bar{X}_c \ldots \bar{X}_k}{k}$$

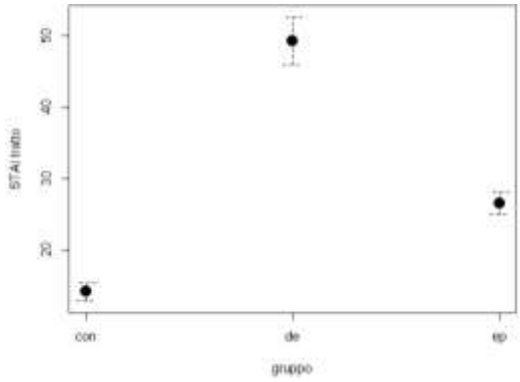
Analogamente, i **coefficienti angolari** non corrispondono più alla differenza tra le medie dei gruppi: quando un gruppo è confrontato con gruppi accorpati, il b_1 esprime la **differenza** tra la **media del gruppo singolo** e la **media delle medie dei gruppi accorpati**, **divisa** per il **numero di gruppi coinvolti** nel confronto. Quando sono confrontati due gruppi singoli e almeno un altro è escluso dal confronto, il b_1 esprime la differenza tra le due medie, divisa per il numero di gruppi coinvolti (due).

$$b_1 = \left(\bar{X}_a - \frac{\bar{X}_b + c_a + \cdots \bar{X}_k}{k_{b,c \ldots k}}\right) / k$$

Per verificare quanto detto, usiamo il dataframe **epde**, in cui i gruppi sono perfettamente **bilanciati**: c'è un ugual numero di soggetti che compongono il gruppo di controllo (**con**: nessun sintomo nella sfera sessuale) e il gruppo di pazienti con disfunzione erettile (**de**) o con eiaculazione precoce (**pe**) che si sono rivolti a un Centro di terapia per problemi relazionali e disturbi sessuali primari. È facile ipotizzare, secondo letteratura, che i livelli di ansia di tratto siano prevalenti tra i pazienti:

[97] Le formule di b_0 e $b_1$si riferiscono a **disegni bilanciati**, cioè l'optimum per l'analisi della varianza; in caso di disegni non bilanciati, le formule cambiano, ponderando il risultato in funzione della diversa numerosità dei gruppi... ma possiamo evitare eccessivi dettagli, pur non dimenticandoci di questo dato di fatto.

```
table(epde$gruppo)
con  de  ep
 29  29  29
tapply(epde$STAI_tratto, epde$gruppo, mean)
      con       de       ep
14.13793 49.31034 26.51724
mean(epde$STAI_tratto)
[1] 29.98851     ← b0 del modello lineare costruito con i contrasti multipli

49.31-14.14
[1] 35.17
26.52-14.14
[1] 12.38
```



Secondo i contrasti **semplici** di default:

```
contrasts(epde$gruppo)
     de ep
con   0  0
de    1  0
ep    0  1

lm(epde$STAI_tratto~epde$gruppo)
Coefficients:
   (Intercept)  epde2$gruppode  epde2$gruppoep
         14.14           35.17           12.38
```

, il gruppo dei controlli ha **significativamente** (guardate i CI nel plot) **meno ansia** di tratto dei pazienti **de** e dei pazienti **ep**. La loro media (14.14) coincide con l'intercetta del modello e i due b_1 corrispondono alle differenze tra de e con ($49.31 - 14.14$), e tra ep e con ($26.52 - 14.14$).

Vediamo cosa succede impostando **due contrasti multipli**: prima (C_1) impostiamo la differenza tra il gruppo di controllo e i due gruppi di pazienti accorpati (con *versus* de+ep), poi (C_2) ignoriamo il gruppo di controllo e confrontiamo de con ep. Come nel paragrafo precedente, salviamo le medie dei gruppi ottenute con `tapply` nell'oggetto **`ansia`**: il primo elemento di `ansia` è la media dei controlli, il secondo la media dei pazienti `de`, il terzo la media dei pazienti `ep`:

```
ansia<-tapply(epde$STAI_tratto, epde$gruppo, mean)
ansia[1]; ansia[2]; ansia[3]
     con
14.13793
      de
49.31034
      ep
26.51724
```

L'intercetta del modello con contrasti multiplo sarà la media delle medie dei gruppi, ovvero la *grand mean* che abbiamo già visto:

```
b0<-(ansia[1]+ansia[2]+ansia[3])/3
b0; mean(epde$STAI_tratto)
     con
29.98851
[1] 29.98851
```

Il primo b_1 del modello *overall* (primo contrasto) è dato dalla differenza tra la media di `con` e la media delle medie di `de` ed `ep`, divisa per il numero di gruppi coinvolti nel contrasto (tre). Calcoliamola (occhio alle parentesi):

```
(b1_con_epde<-(ansia[1]-((ansia[2]+ansia[3])/2))/3)
      con
-7.925287
```

Il secondo b_1 del modello *overall* (secondo contrasto) è dato dalla differenza tra la media di `de` e la media di `pe`, divisa per il numero di gruppi coinvolti (2), ignorando il gruppo di controllo:

```
(b2_ep_de<-(ansia[2]-ansia[3])/2)
      de
11.39655
```

Giusto. Ovviamente, ciò vuol dire che **per ottenere la differenza** tra controlli e i due gruppi di pazienti accorpati si dovrà moltiplicare il primo b_1 per 3, e per avere la differenza tra de ed ep si moltiplicherà il secondo b_1 per 2:

```
b1_con_epde*3; ansia[1]-((ansia[2]+ansia[3])/2)
      con
-23.77586
      con
-23.77586

b1_ep_de*2; ansia[2]-ansia[3]
     de
22.7931
     de
22.7931
```

Facciamo fare tutto a R. Stabiliamo i pesi da assegnare ai contrasti del fattore gruppo, secondo le due H_0:

$C_1 \rightarrow H_0: +\mathbf{2} * \bar{X}_a + (-\mathbf{1}) * \bar{X}_b + (-\mathbf{1}) * \bar{X}_c = \mathbf{0} \rightarrow$ `con_epde<-c(2, -1, -1)`

$C_2 \rightarrow H_0: \mathbf{0} * \bar{X}_a + \mathbf{1} * \bar{X}_b + (-\mathbf{1}) * \bar{X}_c = \mathbf{0} \rightarrow$ `ep_de<-c(0, 1, -1)`

```
contrasts(epde$gruppo)<-cbind(con_epde, ep_de)
contrasts(epde$gruppo)
    con_epde ep_de
con       2     0
de       -1     1
ep       -1    -1
```

Vediamo i parametri del modello:

```
lm(epde$STAI_tratto~epde$gruppo)
Coefficients:
        (Intercept)  epde$gruppocon_epde     epde$gruppoep_de
             29.989                -7.925               11.397
```

Quindi, mentre b_0 rappresenta la media dell'ansia del campione, indipendentemente dal gruppo di appartenenza, il primo $b_{1_{con-epde}} = -7.925$ esprime la **differenza** tra la **media del gruppo di controllo** e la media **delle medie dei gruppi dei pazienti**; questa differenza è **divisa** per **il numero di gruppi coinvolti** (**3**). Il secondo $b_{1_{ep-de}} = 11.397$ esprime la **differenza** tra le **medie del gruppo de** e **del gruppo ep**, **divisa per il numero di gruppi** coinvolti (**2**).

```
stai<-lm(epde$STAI_tratto~epde$gruppo)
summary(stai)
---
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          29.9885     0.6337   47.32   <2e-16
epde$gruppocon_epde  -7.9253     0.4481  -17.69   <2e-16
epde$gruppoep_de     11.3966     0.7761   14.68   <2e-16
---
Residual standard error: 5.911 on 84 degrees of freedom
Multiple R-squared:  0.8628,      Adjusted R-squared:  0.8596
F-statistic: 264.2 on 2 and 84 DF,  p-value: < 2.2e-16
confint(stai)
                   2.5 %   97.5 %
(Intercept)   11.955164 16.32070
epde$gruppode 32.085516 38.25931
epde$gruppoep  9.292412 15.46621
```

La differenza tra controlli e gruppi clinici accorpati è significativa, e il suo CI in popolazione piuttosto ristretto; i pazienti con disfunzione erettile hanno significativamente più ansia dei pazienti con eiaculazione precoce, e anche l'ampiezza di questa differenza è accettabile. Nel complesso, l'appartenenza al gruppo spiega una davvero grande quantità di variabilità della predisposizione all'ansia, nel campione (86.3%) quanto nella popolazione (85.9%): il suo effetto è, naturalmente, significativo.

Ora verifichiamo se abbiamo costruito una coppia di contrasti ortogonali: sono rispettivamente $C_1[+2,-1,1]$ (controlli versus offenders e vittime accorpati) e $C_2[0,1,-1]$ (offenders versus vittime), perciò:

$$\begin{bmatrix}2\\-1\\-1\end{bmatrix}\times\begin{bmatrix}0\\1\\-1\end{bmatrix}=(2\times0)+(-1\times1)+(-1\times-1)=0-1+1=0$$

Sì, i contrasti impostati sono **ortogonali**; non solo:

```
c1<-c(2,-1,-1); c2<-c(0,1,-1)
cor(c1,c2)
[1] 0
```

Sono anche non correlati: il problema del *family-wise – error rate* non si pone.

Un esempio di contrasti multipli **non ortogonali** può contenere un gruppo usato singolarmente in un contrasto e nuovamente utilizzato in un secondo contrasto: per esempio, dopo aver confrontato i controlli con offender e vittime accorpati, potremmo decidere di confrontare nuovamente i controlli con le sole vittime, che sembrano avere una differenza meno marcata. Avremo perciò: $C_1[+2,-1,-1]$ e $C_2[-1,0,1]$, e quindi:

$$\begin{bmatrix}2\\-1\\-1\end{bmatrix}\times\begin{bmatrix}-1\\0\\1\end{bmatrix}=\big(2\times(-1)\big)+(-1\times0)+(-1\times1)=-2+0-1=-3$$

```
c1<-c(2,-1,-1); c2<-c(-1,0,1)
cor(c1,c2)
[1] -0.8660254
```

I contrasti **non** sono ortogonali e sono **correlati**: applicando questi contrasti al predittore, R **restituisce comunque il modello lineare** e i contrasti richiesti, ma la loro mancata indipendenza **non elude il *family-wise error rate***.

Come per i contrasti semplici, anche per quelli multipli R dispone di **contrasti preimpostati**: ad esempio, nel **contrasto di Helmert**: `contr.helmert(n=)`, in cui `n=` è il numero dei gruppi, è possibile confrontare il primo livello con il secondo (gli altri sono ignorati); poi il primo e secondo accorpati sono confrontati con il terzo, quindi il primo-secondo-terzo accorpati sono confrontati con il quarto e così via.

Nella tabella seguente sono riassunti alcuni dei contrasti semplici e multipli di uso più corrente.

Nome	Cosa fa	Con X a 3 livelli		Con X a 4 livelli	
Semplice (iniziale) `contr.treatment(n=, base= 1)`	Confronta la media del **primo** gruppo con quella di ciascuno degli altri gruppi	C_1: 1 vs 2 C_2: 1 vs 3		C_1: 1 vs 2 C_2: 1 vs 3 C_3: 1 vs 4	

Semplice (finale)
`contr.SAS(n= gruppi)`
	Confronta la media dell'**ultimo** gruppo con quella di ciascuno degli altri gruppi	C_1: 3 vs 1
		C_2: 3 vs 2
			`contrasts(epde$gruppo)`
			`  1 2`
			`con 1 0`
			`de 0 1`
			`ep 0 0`
				C_1: 4 vs 1
				C_2: 4 vs 2
				C_3: 4 vs 3
					`contrasts(epde$attivita_sessuale_mese)`
					`  1 2 3`
					`>=7 volte 1 0 0`
					`1-2 volte 0 1 0`
					`3-4 volte 0 0 1`
					`5-6 volte 0 0 0`

Di Helmert	La media semplice o accorpata di ogni gruppo, tranne l'ultimo, è confrontata con le medie accorpate delle categorie successive. È ortogonale	C_1: 1 vs (2,3)
		C_2: 2 vs 3
			`c1<-c(2/3,-1/3,-1/3)`
			`C2<-c(0,1/2,-1/2)`
				C_1: 1 vs (2,3,4)
				C_2: 2 vs (3,4)
				C_3: 3 vs 4
					`C1<-c(3/4,-1/4,-1/4,-1/4)`
					`C2<-c(0,2/3,-1/3,-1/3)`
					`C3<-c(0,0,1/2,1/2)`

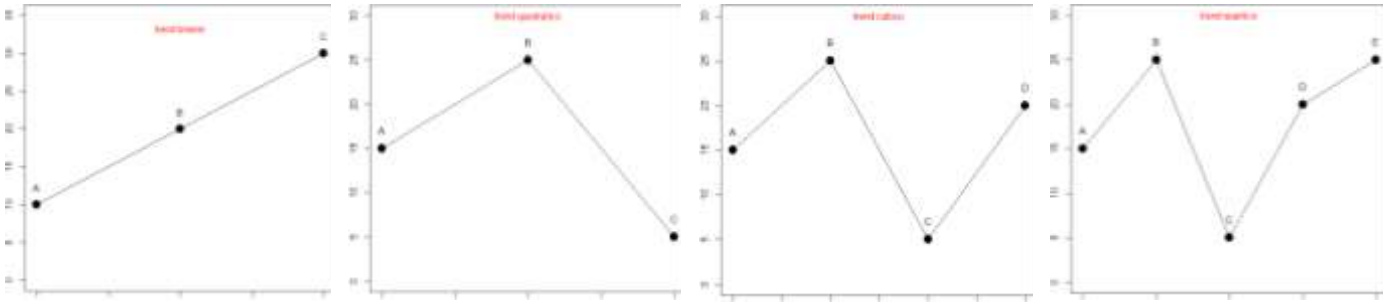
Differenza
(reverse Helmert)
`contr.helmert(n=gruppi)`
	La media semplice o accorpata di ogni gruppo, tranne il primo, è confrontata le medie accorpate della categoria precedenti	C_1: 2 vs 1 C_2: 3 vs (2,1	`contrasts(epde$gruppo)`
			`  [,1] [,2]`
			`con -1 -1`
			`de 1 -1`
			`ep 0 2`
				C_1: 2 vs 1
				C_2: 3 vs (2,1)
				C_3: 4 vs (3,2,1)
					`contrasts(epde$attivita_sessuale_mese)`
					`  [,1][,2][,3]`
					`>=7 volte -1 -1 -1`
					`1-2 volte 1 -1 -1`
					`3-4 volte 0 2 -1`
					`5-6 volte 0 0 3`

Deviazione (finale)
`contr.sum(n=gruppi)`
	La media di un dato livello è confrontata con la media delle medie di Y per ogni livello di X. L'ultimo livello non viene confrontato. È ortogonale	C_1: 1 vs $M_{1,2,3}$
		C_2: 2 vs $M_{1,2,3}$
			`contrasts(epde$gruppo)`
			`  [,1] [,2]`
			`con 1 0`
			`de 0 1`
			`ep -1 -1`
				C_1: 1 vs $M_{1,2,3}$
				C_2: 1 vs $M_{1,2,3}$
				C_2: 2 vs $M_{1,2,3}$
					`contrasts(epde$attivita_sessuale_mese)`
					`  [,1][,2][,3]`
					`>=7 volte 1 0 0`
					`1-2 volte 0 1 0`
					`3-4 volte 0 0 1`
					`5-6 volte -1 -1 -1`

Deviazione (iniziale)	Confronta l'effetto di ogni gruppo (tranne il **primo**) con con la media delle medie di Y per ogni livello di X.	C_1: 2 vs (1,2,3) C_2: 3 vs (1,2,3)		C_1: 2 vs (1,2,3,4) C_2: 3 vs (1,2,3,4) C_3: 4 vs (1,2,3,4)	

Infine, nel caso in cui il predittore X sia un **fattore ordinato** (`ordered factor`), è possibile verificare il **trend delle medie dei gruppi**, ovvero se i **gruppi sono collocabili in un qualche ordine**, per cui è possibile applicare un **contrasto polinomiale** e verificare se nel passaggio da un gruppo all'altro sia riconoscibile un **trend**: `contr.poly(n=)`, **ortogonale**, in cui `n=` è il numero dei gruppi. Anche se il contesto "naturale" di questo tipo di fattore ordinato e dei conseguenti contrasti per il trend è un disegno **within subjects**, soprattutto **longitudinale** (i livelli del fattore sono i diversi momenti di rilevazione di Y), è possibile individuare fattori ordinati anche in disegni **between groups**, come vedremo nell'esempio di seguito. In questo tipo di fattore, il **contrasto polinomiale è il contrasto impostato per default** (non i *treatment contrasts* dei fattori non ordinati). Nell'output sarà indicato se nella disposizione dei dati da un gruppo all'altro è rilevabile (significativo) un trend lineare, ovvero una crescita proporzionale tra i gruppi, e/o un trend quadratico e/o un trend cubico, et cetera:

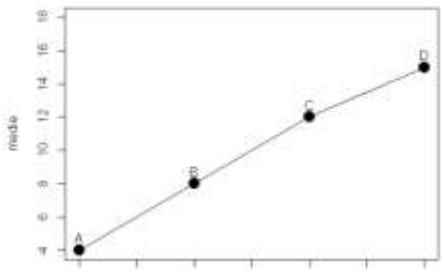
- con almeno due gruppi, possiamo rilevare un trend **lineare** (primo ordine), in cui le medie dei gruppi ordinate crescono (o diminuiscono) in modo proporzionale (cambiamento proporzionale in Y al variare di X), da un'assenza di differenza tra gruppi;
- con **almeno tre gruppi**, possiamo rilevare **anche** un trend **quadratico** (secondo ordine), in cui si verifica **un cambiamento** (una curva) nella direzione della linea;
- con **almeno quattro gruppi**, possiamo rilevare **anche** un trend **cubico** (terzo ordine), in cui si verificano **due cambiamenti** (due curve) nella direzione della linea;
- con **più di quattro gruppi**, possiamo rilevare anche trend di **ordine superiore**, in cui si verificano ***n* cambiamenti** (*n* curve) nella direzione della linea (ad esempio un trend quartico, con tre cambiamenti).



Ciascuno di questi trend ha un set di coefficienti per le variabili di contrasto nel modello: i contrasti operano quindi come precedentemente descritto, **ma** - fortunatamente – R già predispone i coefficienti per pesare le medie in ogni contrasto, senza necessità di calcolarli[98]: i pesi assegnati a ogni contrasto danno $\sum = 0$, e anche la somma dei prodotti dei pesi assegnati ai contrasti è $\sum = 0$, per cui i **contrasti polinomiali sono ortogonali**. Con due gruppi ordinati sarà valutato solo il contrasto relativo al trend lineare (`.L`); con tre gruppi saranno valutati due contrasti: per il trend lineare e per il trend quadratico (`.Q`); con quattro gruppi saranno valutati tre contrasti: per il trend lineare, per quello quadratico e per il trend cubico (`.C`), et cetera.

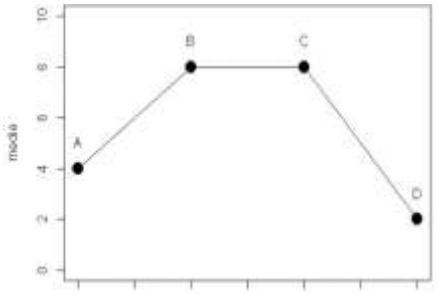
Tanto per chiarire, vediamo qualche esempio.

Contrasto per il trend lineare → esiste un trend tale per cui dal primo al secondo, dal secondo al terzo, dal terzo al quarto livello di X la tendenza delle medie è solo a salire, in modo proporzionale? $\boldsymbol{H_{0_L} = -3 - 1 + 1 + 3 = 0}$; $\boldsymbol{H_1}$**:** passando da un gruppo all'altro, le medie aumentano significativamente. Il modello, con una sola X di primo ordine, è il ben noto: $y_i = b_0 + b_1 X_{ji} + e_i$.

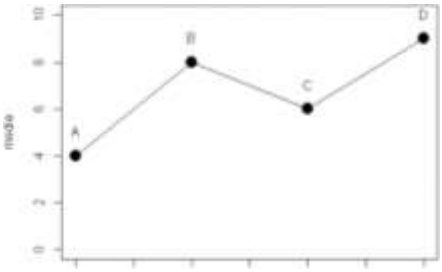


[98] Si usa algebra matriciale

Contrasto per il trend quadratico → esiste un trend tale per cui dal primo al secondo livello di X il trend sale, tra secondo e terzo si stabilizza e tra terzo e quarto scende? $\boldsymbol{H_{0_Q} = +1 - 1 - 1 + 1 = 0;}$ $\boldsymbol{H_1:}$ passando da X_a a X_b Y aumenta, da X_b a X_c Y non cambia, da X_c a X_d Y diminuisce. Le medie accorpate di X_a e X_d sono significativamente diverse dalle medie accorpate di X_b e X_c. Il modello comprende **anche** il predittore al quadrato: $y_i = b_0 + b_1 X_{ji} + \boldsymbol{b_1 X_{ji}^2} + e_i$.



Contrasto per il trend cubico → esiste un trend tale per cui dal primo al secondo livello il trend sale, tra secondo e terzo scende e tra terzo e quarto torna a salire? $\boldsymbol{H_{0_C} = -1 + 3 - 3 + 1 = 0}$; **H1:** le media di X_a e X_c accorpate sono significativamente diverse dalle medie di X_b e X_d accorpate. Il modello comprende **anche** il predittore al cubo: $y_i = b_0 + b_1 X_{ji} + \boldsymbol{b_1 X_{ji}^2} + \boldsymbol{b_1 X_{ji}^3} e_i$.



E così via, per trend / modelli sempre più complessi, anche se in genere è piuttosto raro l'uso di curve di ordine superiore al secondo.

Come esempio concreto, vediamo la soddisfazione per l'attività sessuale di coppia, rilevata nel campione con il test IIEF (International Index of Erectile Function) che valuta diverse dimensioni: funzionalità erettile, orgasmo, desiderio, soddisfazione per la qualità dei rapporti e soddisfazione complessiva, che è l'unica scala disponibile nel dataframe. È ragionevole pensare che una maggiore soddisfazione sia manifestata in coloro che dichiarano una maggiore frequenza dei rapporti di coppia, che è una variabile categoriale a quattro livelli, **non ordinati**:

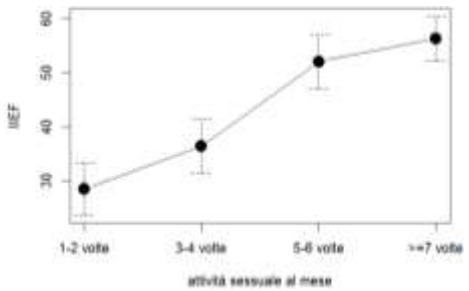
```
class(epde$attivita_sessuale_mese)
[1] "factor"

levels(epde$attivita_sessuale_mese)
[1] ">=7 volte" "1-2 volte" "3-4 volte" "5-6 volte"
```

Rendiamo il fattore ordinato in senso crescente, come abbiamo già imparato, con **ordered(fattore, livelli=)**; creiamo una nuova variabile:

```
epde$attivita_ordinato<-ordered(epde$attivita_sessuale_mese, levels=c("1-2 volte","3-4 volte","5
-6 volte",">=7 volte"))

class(epde$attivita_ordinato)
[1] "ordered" "factor"

levels(epde$attivita_ordinato)
[1] "1-2 volte" "3-4 volte" "5-6 volte" ">=7 volte"
```

Ora l'ordine dei gruppi ha un senso, esprimendo una sempre più frequente attività sessuale. Possiamo descrivere il dato e impostare il modello usando contrasti per la stima del trend; poiché abbiamo quattro gruppi, i trend possibili sono tre: lineare (variazione costante dal primo al quarto gruppo), quadratico (una sola riconoscibile variazione dell'andamento) e cubico (due variazioni riconoscibili nel trend).

```
Desc(epde$IIEF~epde$attivita_ordinato, digits = 2)
------------------------------------------------------------
        1-2 volte  3-4 volte  5-6 volte  >=7 volte
mean        28.52      36.49      52.09      56.40
median      24.00      37.00      53.00      53.50
sd          13.28      14.38       7.46       5.82
```



L'incremento della soddisfazione è rilevante dal primo al secondo gruppo, così come dal secondo al terzo: fin qui, il trend sembra lineare, anche se la differenza tra terzo e quarto gruppo è minore. Non è necessario calcolare i coefficienti dei contrasti polinomiali, che negli `ordered factors` sono già preimpostati in `contr.poly()` (possono essere assegnati ai contrasti del factor con `contr.poly(numero di livelli)`). Vediamoli per la nostra X a quattro livelli (`.L` → componente lineare, `.Q` → componente quadratica, `.C` → componente cubica):

```
contrasts(epde$attivita_ordinato)
             000000.L          .Q          .C
Xa [1,] -0.6708204        0.5      -0.2236068
Xb [2,] -0.2236068       -0.5       0.6708204
Xc [3,]  0.2236068       -0.5      -0.6708204
Xd [4,]  0.6708204        0.5       0.2236068
```

↑	↑	↑
H_0: la media di A e B accorpati è uguale alla media di C e D accorpati	*H_0: la media di A e D accorpati è uguale alla media di B e C accorpati*	*H_0: la media di A e C accorpati è uguale alla media di B e D accorpati*

```
summary(lm(epde$IIEF~epde$attivita_ordinato))
[omissis]
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                 43.373      1.581  27.432  < 2e-16
epde$attivita_ordinato.L    22.194      3.226   6.880 1.04e-09
epde$attivita_ordinato.Q    -1.830      3.162  -0.579    0.564
epde$attivita_ordinato.C    -4.233      3.097  -1.367    0.175

Residual standard error: 12.61 on 83 degrees of freedom
Multiple R-squared:  0.3902,      Adjusted R-squared:  0.3681
F-statistic:  17.7 on 3 and 83 DF,  p-value: 5.689e-09
```

Nel complesso, la soddisfazione per l'attività sessuale di coppia è significativamente differente ($F_{[3;83]} = 17.7, p < .001$) a seconda della frequenza dei rapporti, che ne spiega il 39.1% di variabilità nel campione e il 36.8% stimato in popolazione. La sola componente significativa del trend è **quella lineare**: l'incremento delle medie dal primo al quarto gruppo è quindi proporzionale, mentre non sono significativi i contrasti che verificano un polinomio quadratico e cubico.

12.1.2 Confronti a coppie a posteriori o test post hoc

I confronti a coppie *post hoc*, applicabili se l'effetto del predittore è significativo, confrontano ogni livello con ogni altro (con l'eccezione del test di Dunnet, come vedremo). I test *post hoc* per la differenza a coppie tra le medie rispecchiano la forma – base del t-test per campioni indipendenti visto nel Capitolo 4, ma **usano la varianza d'errore MS_R** del modello lineare, al posto delle s^2 associate ai gruppi, e i suoi df_R, invece di $df = N - 2$.

Gruppi non bilanciati: $$t_{post\ hoc, df_R} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{MS_R\left(\frac{1}{N_{x1}} + \frac{1}{N_{x2}}\right)}}$$

Gruppi bilanciati: $$t_{post\ hoc, df_R} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{2MS_R}{N}}}$$

Tuttavia, la vera differenza tra una serie di confronti a coppie con t-test e un set di t-test post hoc sta nel fatto che il $p - value$ di ogni t-test in questo set è corretto tenendo conto del più volte citato *family-wise error rate*. Sono **moltissimi i test post hoc proposti in letteratura,** diversi per controllo dell'errore di I e II tipo, rispondenza del modello ai requisiti di normalità e omoschedasticità, tipo di Y: vedremo qui solo quelli più frequentemente usati in letteratura, e disponibili tra le statistiche di base in R, o in package che usiamo anche per altre funzioni.

a. Test basati sul family-wise error rate

Il principio di **Bonferroni, o disuguaglianza di Bonferroni**, afferma che per effettuare k volte il confronto a coppie tra le medie con il t-test, mantenendo **costante** la probabilità di non incorrere in un errore di I tipo **nell'intero set di**

confronti (α_T), la probabilità di non incorrere in un errore di I tipo **in ogni confronto** (α_k) deve essere minore di α_T **divisa per il numero di confronti** k → **Bonferroni:** α_T/k.

Quindi, con X a tre livelli e quindi tre confronti a coppie, per $\alpha_T = .05$ l'α_k di ogni confronto sarà $\alpha_k < (.05 / 3) = 0.0167$.

In realtà, come dimostrano Dunn (1961) e Sidack (1967), nonché Sokal e Rohlf (1981), la relazione tra α_T e α_k è esponenziale → **Dunn − Sidak:** $\alpha_k = 1 - (1 - \alpha_T)^{1/k}$, il che porta le due correzioni a soglie diverse;

```
alfa_bonferroni<-.05/5
alfa_dunn_sidak<-1-(.95^(1/5))
alfa_bonferroni;alfa_dunn_sidak
[1] 0.01
[1] 0.01020622
```

Al crescere del numero dei confronti, la differenza tra le soglie α_k definite dalle due correzioni aumenta:

```
alfa_bonferroni<-.05/10
alfa_dunn_sidak<-1-(.95^(1/10))
alfa_bonferroni;alfa_dunn_sidak
[1] 0.005
[1]0.005116197
```

La correzione secondo Bonferroni porta a soglie di rifiuto di H_0 più alte, tanto da essere eccessivamente conservativa per un ampio numero di confronti (>6-7), in quanto riduce troppo la potenza del test e induce a errori di II tipo. Notate che la soglia critica α_k, con entrambi i metodi, è la stessa per tutti i confronti a coppie, **indipendentemente dalla grandezza della differenza tra le medie**: una differenza $\bar{x}_1 - \bar{x}_2 = 15$ ha la stessa soglia critica di una differenza molto più piccola, ad esempio $\bar{x}_1 - \bar{x}_2 = 5$.

In R, si possono fare *t*-test tra livelli applicando la correzione secondo Bonferroni con una delle opzioni disponibili nella funzione di base `pairwise.t.test(Y, X, p.adjust.method= "bonferroni")`, che approfondiremo nel prossimo paragrafo.

Un diverso approccio ai confronti multipli, molto diffuso in letteratura, è quello di **Tukey** (1949; 1977): **Honestly significant difference (*HSD*)** o *wholly significant difference test*. Se X ha un effetto *overall* significativo, per ogni coppia dei suoi livelli si calcola l'HSD, data dalla differenza tra le medie in rapporto alla radice quadrata della MS_R divisa per la numerosità dei gruppi n (se bilanciati).

$$HSD_{x_1-x_2} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{MS_R}{n}}}$$

Ogni HSD è quindi confrontata con il valore critico proposto da Tukey: se è maggiore del valore critico, allora la differenza tra le due medie è significativa. "*Honest*" fa riferimento all'essere un compromesso onesto, onorevole, tra la soglia di significatività del modello completo e le soglie di significatività dei confronti a coppie. Richiede che **normalità e omoschedasticità del modello siano rispettate**. Se i gruppi **non sono bilanciati**, invece della MS_R si usa una stima delle sd (metodo di **Tukey-Kramer**, 1956), che è una **procedura per passi**: le **medie sono ordinate dalla più grande alla più piccola** (per esempio, con $k = 5$, dalla 1° alla 5°) e si confrontano le due medie agli estremi (1° e 5°). Se il rapporto F del modello *overall* è significativo, la differenza tra queste due medie estreme è significativa. Successivamente, si procede dall'esterno all'interno della distribuzione delle medie, valutando la significatività della differenza più grande tra la prima media in ordine di grandezza e la penultima (1° e 4°) o la seconda media in ordine di grandezza e l'ultima (2° e 5°): questa differenza risulterà inferiore alla precedente. Se è significativa, si proseguirà verso l'interno della distribuzione delle medie, con differenze sempre più piccole: alla prima differenza che non significativa, il procedimento si ferma, perché si assume che tutte le differenze tra le medie comprese entro queste ultime due non saranno più significative. In R, la funzione di base è `TukeyHSD(aov(modello))`, il cui oggetto è il modello *overall*

costruito con **aov**; per ogni confronto post hoc, sono prodotti la differenza tra le medie dei livelli, il relativo ***CI*** (di default al 95%) e il $p - value$ corretto per il family-wise error rate.

```
dipe<-aov(cp$dipendente~cp$gruppo_quattro_livelli)
TukeyHSD(dipe)
Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = cp$dipendente ~ cp$gruppo_quattro_livelli)

$`cp$gruppo_quattro_livelli`
                          diff         lwr      upr     p adj
controllo_M-controllo_F -1.714286 -20.4569980 17.02843 0.9950497
offender-controllo_F    17.675595  -2.4782682 37.82946 0.1059310
vittime-controllo_F     29.363095   9.2092318 49.51696 0.0015213
offender-controllo_M    19.389881  -0.7639825 39.54374 0.0636421
vittime-controllo_M     31.077381  10.9235175 51.23124 0.0007188
vittime-offender        11.687500  -9.7849745 33.15997 0.4836437
```

Il **test di Scheffé** (1953, 1959; modificato da Gabriel, 1978) è piuttosto flessibile: oltre a singoli livelli, consente di **confrontare più una coppia di medie contemporaneamente**, ed è applicabile a disegni non bilanciati. Tuttavia, è **molto conservativo**, dunque poco potente: aumenta molto la probabilità di errore di II tipo; quindi, può essere preferibile solo per confronti complessi, non gestiti dagli altri test post hoc.

Applicata alle medie la formula del t-test proposta da Bonferroni (***S***), sono significative per l'alfa prescelto tutte le differenze tra due medie quando S è maggiore, in valore assoluto, del valore critico S_α: k è il numero di gruppi a confronto e F è il quantile di una distribuzione F corrispondente all'α prescelto, per $df = k-1, n-k$.

$$S_\alpha = \sqrt{(k-1) \times F_{\alpha,k-1,n-k}}$$

In R, si può trovare in `DescTools`: l'oggetto della funzione di **`ScheffeTest(aov(modello))`** è di classe `aov`. Nell'output viene mostrata la differenza tra le medie (semplici o accorpate), il loro CI e il $p - value$ corretto. Per post hoc tra le **singole** medie, è sufficiente indicare l'oggetto `aov`:

```
ScheffeTest(dipe)
Post hoc multiple comparisons of means : Scheffe Test
  95% family-wise confidence level

$ cp$gruppo_quattro_livelli
                          diff          lwr.ci     upr.ci      pval
controllo_M-controllo_F -1.714286      -22.115493   18.68692    0.9963
offender-controllo_F    17.675595       -4.261633   39.61282    0.1597
vittime-controllo_F     29.363095        7.425867   51.30032    0.0038
offender-controllo_M    19.389981       -2.547347   41.32711    0.1033
vittime-controllo_M     31.077381        9.140153   53.01461    0.0019
vittime-offender        11.687500      -11.685020   35.06002    0.5648
```

Notate che la differenza tra offenders e controlli maschi, a soglia nel test di Tukey, è qui invece chiaramente nella regione di accettazione di H_0.

Per post hoc tra **medie accorpate**, occorre aggiungere l'argomento `contrasts=`, in cui si costruisce la matrice (`matrix`) dei contrasti seguendo le usuali regole. Ad esempio, vogliamo prima confrontare i soggetti di controllo con i due gruppi clinici (**primo** contrasto**:** $\mathbf{1, 1, -1, -1}$), poi il gruppo di controllo con le vittime, ignorando gli offenders (**secondo** contrasto: $\mathbf{1, 1, 0, -2}$), infine gli offenders con le vittime, ignorando i controlli (**terzo** contrasto: $\mathbf{0, 0, 1, -1}$). **Questi contrasti semplici non sono ortogonali** (potete verificarlo costruendo la tabella dei pesi), pertanto non sfuggono al family-wise error rate, che però il test post hoc corregge:

```
ScheffeTest(dipe, contrasts= matrix(c(1,1,-1,-1,1,1,0,-2,0,0,1,-1), ncol=3))
Post hoc multiple comparisons of means : Scheffe Test
  95% family-wise confidence level
$ cp$gruppo_quattro_livelli
                                          diff      lwr.ci      upr.ci       pval
controllo_F,controllo_M-offender,vittime  -48.72598  -79.77690   -17.72905    0.00046
controllo_F,controllo_M-vittime           -60.44048  -99.28322   -21.59773    0.00053
offender-vittime                          -11.68750  -35.06002    11.68502    0.56478
```

Se il **modello è fattoriale**, `ScheffeTest` produce post hoc **per tutti i fattori previsti**, compreso il termine d'interazione (lo vedremo in ANOVA fattoriale). Ad esempio, se **aggiungiamo** il genere:

```
gruppo_istruzione<-aov(cp$dipendente~cp$gruppo_quattro_livelli+cp$istruzione)
```

, otteniamo:

```
ScheffeTest(gruppo_istruzione)
Post hoc multiple comparisons of means : Scheffe Test
  95% family-wise confidence level

$ cp$gruppo_quattro_livelli
                              diff          lwr.ci      upr.ci     pval
controllo_M-controllo_F -1.714286       -26.374685    22.94611   1.0000
offender-controllo_F    17.675595         -8.841502    44.19269   0.3992
vittime-controllo_F     29.363095          2.845998    55.88019   0.0203
offender-controllo_M    19.389981         -7.127216    45.90698   0.2930
vittime-controllo_M     31.077381          4.560284    57.59448   0.0113
vittime-offender        11.687500        -16.564537    39.93954   0.8457

$ cp$istruzione
                                        diff     lwr.ci      upr.ci      pval
laurea-diploma superiore            -3.158778  -23.84775    17.53020    0.9980
media inferiore-diploma superiore   -4.739364  -30.69369    21.21496    0.9954
media inferiore-laurea              -1.580586  -28.72426    25.56309    1.0000
```

Il modello **con interazione** produce anche i post hoc per tutte le possibili combinazioni dei livelli entro e tra X_1 e X_2; vediamo il modello **personalità dipendente ~ genere** × **livello di empatia**, che ha meno combinazioni del precedente, per semplificare un po' il già lungo elenco di combinazioni (provate a verificare i post hoc d'interazione per `cp$dipendente~cp$gruppo_quattro_livelli+cp$istruzione`...):

```
genere_empatia<-aov(cp$dipendente~cp$genere*cp$empatia_etichetta)
ScheffeTest(genere_empatia)
Post hoc multiple comparisons of means : Scheffe Test
  95% family-wise confidence level

$ cp$genere                          ← effetto principale del genere
           diff        lwr.ci       upr.ci     pval
M-F    -6.027027     -27.10619     15.05214    0.9646

$ cp$empatia_etichetta               ← effetto principale dell'empatia
                     diff      lwr.ci      upr.ci      pval
bassa-alta      8.5487080   -21.50642    38.60384    0.9654
norma-alta      8.2610720   -20.52991    37.05205    0.9641
norma-bassa    -0.2876361   -23.90812    23.33285    1.0000

$ cp$genere:o$empatia_etichetta ← effetto di interazione genere*empatia
                           diff       lwr.ci      upr.ci      pval
M:alta-F:alta       -9.9777778    -60.54825    40.59270    0.9933
F:bassa-F:alta       3.7000000    -45.95922    53.35922    0.9999
M:bassa-F:alta       1.4250000    -45.02695    47.87695    1.0000
F:norma -F:alta      5.4818182    -39.43664    50.40027    0.9993
M:norma-F:alta       0.1333333    -48.12675    48.39342    1.0000
F:bassa-M:alta      13.6777778    -27.97988    55.33543    0.9367
M:bassa-M:alta      11.4027778    -26.37427    49.17983    0.9554
F:norma-M:alta      15.4595960    -20.41504    51.33423    0.8217
M:norma-M:alta      10.1111111    -29.86836    50.09058    0.9793
M:bassa-F:bassa     -2.2750000    -38.82320    34.27320    1.0000
F:norma-F:bassa      1.7818182    -32.79642    36.36006    1.0000
M:norma-F:bassa     -3.5666667    -42.38706    35.25373    0.9998
F:norma-M:bassa      4.0568182    -25.73244    33.84608    0.9989
M:norma-M:bassa     -1.2916667    -35.91490    33.33157    1.0000
M:norma-F:norma     -5.3484848    -37.88543    27.18846    0.9972
```

Il test di **Student-Newman-Keuls** (**SNK**) (Newman, 1939; Keuls, 1952) usa una procedura affine a quella di Tukey, ma è **più potente** (protegge meno dall'errore di I tipo). Il test può essere richiesto nell'argomento `method= "newmankeuls"` della funzione `postHocTest(modello aov)` di `DescTools`:

```
postHocTest(dipe, method = "newmankeuls)
Post hoc multiple comparisons of means : Newman-Keuls
  95% family-wise confidence level

$ cp$gruppo_quattro_livelli
                              diff       lwr.ci      upr.ci       pval
controllo_M-controllo_F -1.714286      -15.917739    12.49817    0.81048
offender-controllo_F    17.675595        2.402755    32.94844    0.02935
vittime-controllo_F     29.363095       11.026215    47.69998    0.00079
offender-controllo_M    19.389981        1.053001    37.72676    0.03579
vittime-controllo_M     31.077381       10.923518    51.23124    0.00072
vittime-offender        11.687500       -4.584600    27.95960    0.15645
```

Notate che con questa procedura abbiamo il **massimo numero di confronti a coppie significativi**.

Il **test Q di Dunnet** (1955, 1964 e successivi) è particolare: si usa per il confronto di due o più livelli (**trattamenti**) con un livello di **controllo**. Il numero di confronti effettuati diminuisce rispetto a tutti quelli possibili, dato che è uguale a $k - 1$ (per esempio, con una X a 5 livelli, di cui uno di controllo e quattro trattamenti, i confronti sono X_aX_b, X_aX_c, X_aX_d, X_aX_e): si aumenta la potenza, riducendo però la versatilità dei confronti. Il set dei confronti del test Q è evidentemente un set di contrasti a priori semplici, e quindi potrebbe essere valutato come tale, ma, poiché i contrasti non sarebbero ortogonali, il **controllo sull'errore di I primo tipo garantito del test di Dunnet lo rende preferibile**.

Usa la stessa formula del test di Tukey: c è il gruppo di controllo, i il gruppo di trattamento implicato nel confronto, k è il numero di medie a confronto e ν i df di MS_R.

$$Q_{\alpha,k-1,\nu} = \frac{\bar{x}_c - \bar{x}_i}{\sqrt{MS_R \times \left(\frac{1}{n_c} + \frac{1}{n_i}\right)}}$$

A differenza degli altri test post hoc, che a parità di n ottimizzano la loro potenza con gruppi bilanciati, in questo test la **potenza è maggiore quando il gruppo di controllo ha una n_c maggiore** rispetto ai trattamenti, secondo la relazione: $N_c = N_i \times \sqrt{k-1}$.

Il test Q di Dunnet è disponibile in `DescTools` con `DunnetTest(formula)`, applicabile anche a un subset (`subset=`) e in presenza di dati mancanti (`na.action=`). Ad esempio:

```
DunnetTest(cp$dipendente~cp$gruppo_quattro_livelli)
      Dunnet's test for comparing several treatments with a control :
        95% family-wise confidence level
$controllo_F
                              diff       lwr.ci      upr.ci       pval
controllo_M-controllo_F    -1.714286   -18.8730957   15.44452    0.99088
offender-controllo_F       17.675595    -0.7751128   36.12630    0.06349
vittime-controllo_F        29.363095    10.9123872   47.81380    0.00085
```

Come sarebbe andato il **confronto condotto tra i livelli con i contrasti semplici** in un `lm`?

```
summary(lm(cp$dipendente~o$gruppo_quattro_livelli))
[omissis]
                                      Estimate  Std.Error   t value    Pr(>|t|)
[Intercept]                            35.762      5.036     7.102    8.15e-10
cp$gruppo_quattro_livellicontrolloM    -1.714      7.122    -0.241    0.810477
cp$gruppo_quattro_livellioffender      17.676      7.658     2.308    0.023947
cp$gruppo_quattro_livellivittime       29.363      7.658     3.834    0.000272
```

A parità di differenze tra i livelli (guardate i b_1 dei contrasti semplici e le differenze tra le medie del test di Dunnet), nei contrasti semplici, che, non essendo ortogonali, non sono corretti per il family-wise error rate, il confronto **tra offenders e controlli F è pienamente significativo**, mentre è a soglia nel test di Dunnet.

b. Test basati sul False Discovery Rate

Holm, Benjamini e Hochberg hanno proposto procedure differenti, con un **approccio per passi** (**step**) o **sequenziale**, con qualche differenza tra gli autori.

Holm (1979) è un precursore dell'approccio per passi: nel suo metodo s'inizia calcolando per tutti i confronti a coppie i $p - value$ non corretti, che poi si **ordinano dal più piccolo al più grande**. A ogni $p - value$ è **assegnato un indice** j che indica in quale punto della lista si trova il $p - value$: si assegna **1 al** $p - value$ **più grande**, 2 al secondo più grande e così via, finché il più piccolo $p - value$ riceverà un indice corrispondente al numero di confronti.

Il **valore** α_k **di ogni confronto sarà quindi dato dalla soglia** α_T **divisa per l'indice assegnato** (l'α_k con cui sarà confrontato il $p - value$ più piccolo coinciderà con quello proposto da Bonferroni), e ogni confronto è così corretto per i restanti confronti: quando si incontra la prima differenza non significativa, ci si può fermare, dato che tutte le successive differenze non saranno significative.

$$\alpha_k = \frac{\alpha_T}{j}$$

Una modalità più recente di adattare un approccio sequenziale ai confronti multipli è ignorare il family-wise error rate per concentrarsi sul ***false discovery rate*** (***FDR***) (Benjamini e Hochberg, 1995): la "false discovery" è l'errore di I tipo, e la FDR è la stima di quanti errori di I tipo potrebbero essere stati fatti nelle analisi.

$$FDR = \frac{numero\ di\ H_o falsamente\ respinte}{numero\ totale\ di\ H_o\ respinte}$$

Questo approccio è più potente rispetto al family-wise error rate, e compensa un maggior rischio di errore di I tipo con una riduzione dell'errore di II tipo.

Come nel metodo di Holm, si inizia calcolando i $p - value$ di tutti i confronti e ordinandoli, ma l'ordinamento parte dal $p - \boldsymbol{value}$ **più piccolo**, cui si assegna un **indice** $j = 1$**,** per poi procedere all'insù (procedura *step up*): per ogni confronto si calcola la soglia critica. Di nuovo, per il $p - value$ più piccolo si ha la correzione di Bonferroni, per gli altri il criterio è più liberale.

$$\alpha_k = \frac{j}{k}\alpha_T$$

Vediamo un confronto tra le soglie corrette secondo Bonferroni, Holm e Benjamini-Hochberg (lo script per produrre la tabella `confronti` è in fondo alla dispensa, ma potreste saperla fare anche da soli), relativo al solito modello **`dipe`**:

```
confronti
                    coppia p_non_corretto  alfa_bonf   j_holm  alfa_holm    j_BH   alfa_BH
controlli M vs vittime             0.0001      0.0083        6     0.0083       1    0.0083
controlli F vs vittime             0.0003      0.0083        5     0.0100       2    0.0167
controlli M vs offenders           0.0136      0.0083        4     0.0125       3    0.0250
controlli F vs offenders           0.0239      0.0083        3     0.0167       4    0.0333
offenders vs vittime               0.1564      0.0083        2     0.0250       5    0.0417
controlli M vs controlli F         0.8105      0.0083        1     0.0500       6    0.0500
```

Considerando i $p - value$ non corretti, i **primi quattro** confronti sono significativi, in quanto i loro $p - value$ sono $p < \alpha_T = .05$. Usando la correzione di Bonferroni, solo i **due** $\boldsymbol{p - value}$ dei confronti Controlli M e Controlli F *versus* vittime (le differenze più grandi) sono significativi: $.0001$ e $.0003 < \alpha_k = .0083$. Secondo l'approccio di Holm, restano significativi i primi **due confronti** ($.0001 < \alpha_k = .0083$ e $.0003 < \alpha_k = .0100$), e il terzo (Controlli M versus offenders) è appena superiore alla soglia: $.0136 > \alpha_k = .0125$. Secondo l'approccio di Benjamini-Hochberg, sono i primi **quattro confronti** a essere significativi: $.0001 < \alpha_k = .0083$, $.0003 < \alpha_k = .0167$, $.0136 < \alpha_k = .0250$, $.0239 < \alpha_k = .0333$.

Fortunatamente, la funzione **`pairwise.t.test(Y, X, p.adjust.method="…")`** facilita la decisione su H_0 perché non mostra la soglia alfa corretta per ogni confronto, ma **direttamente i $p-value$ corretti secondo diversi approcci**; il **$p-value$ non corretto** (`p.adjusted.method=` **"none"**) viene:

- **moltiplicato per il numero di confronti**, impostando `p.adjust.method= "bonferroni"`:

```
pairwise.t.test(cp$dipendente,cp$gruppo_quattro_livelli,p.adjust.method="bonferroni")
Pairwise comparisons using t tests with pooled SD

data: cp$dipendente and cp$gruppo_quattro_livelli

                    controllo_F   controllo_M   offender
controllo_M             1.00000             -          -
offender                0.14368       0.08151          -
vittime                 0.00163       0.00076    0.93869
P value adjustment method: bonferroni
```

Cioè:

```
round(confronti$pvalue_non_corretto*6, 3)
[1] 0.001 0.002 0.082 0.143 0.938 4.863
```

Come visto nella tabella precedente, sono significativi solo i due confronti tra controlli F e vittime e tra controlli M e vittime. Notate che quando il calcolo darebbe un $p-value > 1$ (privo di senso), R lo tronca a $p = 1.0$ nell'output.

- **moltiplicato per l'indice** j di Holm, impostando `p.adjust.method= "holm"` (è il metodo di **default**):

```
pairwise.t.test(cp$dipendente,cp$gruppo_quattro_livelli, p.adjust.method = "holm").

        Pairwise comparisons using t tests with pooled SD
data:  cp$dipendente and cp$gruppo_quattro_livelli

                  controllo_F   controllo_M   offender
controllo_M           0.81048             -          -
offender              0.07184       0.05434          -
vittime               0.00136       0.00076    0.31290
```

Cioè:

```
round(confronti$pvalue_non_corretto*confronti$j_holm,3)
[1] 0.001 0.001 0.054 0.072 0.313 0.810
```

Come visto nella tabella precedente, sono significativi solo i due confronti tra controlli F e vittime e tra controlli M e vittime, ma significatività della differenza tra Controlli M e offenders è appena sopra la soglia.

- **moltiplicato per l'indice** j di Benjamini – Hochberg impostando `p.adjust.method= "BH"` o `"fdr"`.

```
pairwise.t.test(cp$dipendente,cp$gruppo_quattro_livelli, p.adjust.method = "BH")
        Pairwise comparisons using t tests with pooled SD
data:  cp$dipendente and cp$gruppo_quattro_livelli

                   controllo_F   controllo_M   offender
controllo_M            0.81048             -          -
offender               0.03592       0.02717          -
vittime                0.00082       0.00076    0.18774
```

c. Test post hoc in caso di violazione dei requisiti del modello lineare

Alcuni test post hoc sono particolarmente raccomandabili se i **requisiti del modello lineare non sono rispettati**.

In caso di **violazione dell'omoschedasticità** si possono citare, ad esempio, i test $T3$ e C di Dunnet o il test di **Games – Howell,** che è più potente, purché N sia abbastanza grande, mentre i due test di Dunnet sono più conservativi rispetto all'errore di I tipo (Toothaker, 1993). Il test di Games – Howell può essere gestito dalla funzione **`games_howell_test(data= dataframe, formula= Y~X)`** del package `rstatix`. Nell'output sono riportate la differenza tra le medie, il suo CI e il $p-value$ corretto (cui si aggiunge un esplicito asterisco per indicarne la significatività) .

```
games_howell_test(data=cp, formula=dipendente~gruppo_quattro_livelli)
# A tibble[99]: 6 x 8
  .y.       group1    group2   estimate conf.low conf.high p.adj p.adj.signif
* <chr>     <chr>     <chr>       <dbl>    <dbl>     <dbl> <dbl> <chr>
1 dipende~ controll~ control~    -1.71    -19.7      16.3 0.994    ns
2 dipende~ controll~ offender    17.7      -1.54     36.9 0.081    ns
3 dipende~ controll~ vittime     29.4       4.92     53.8 0.014     *
4 dipende~ controll~ offender    19.4       2.02     36.8 0.024     *
5 dipende~ controll~ vittime     31.1       7.91     54.2 0.006    **
6 dipende~ offender  vittime     11.7     -12.3      35.7 0.549    ns
```

La procedura post hoc più comune in caso di **violazione della normalità**, con omoschedasticità rispettata, è quella di **Dunn** (Zar, 1999), che in R è gestito dalla funzione `DunnTest(formula)` di `DescTools`: il test si basa sui ranghi e usa la correzione di Holm:

```
DunnTest(cp$dipendente~cp$gruppo_quattro_livelli)
Dunn's test for multiple comparisons using rank sums : holm

                          mean.rank.diff      pval
controllo_M-controllo_F       -0.6428571    0.8227
offender-controllo_F          14.7202381    0.1237
vittime-controllo_F           23.5369881    0.0047
offender-controllo_M          15.3630952    0.1237
vittime-controllo_M           24.2068452    0.0040
vittime-offender              28.8437500    0.4873
```

Se le violazioni sono a carico di tutti i requisiti, e quindi l'opzione di analisi *overall* è più correttamente un approccio non parametrico, fate riferimento ai confronti post hoc non parametrici / robusti presentati nel §12.2.

Ci sono molte altre procedure post hoc oltre a quelle presentate in questo capitolo, alcune delle quali oramai piuttosto superate (ad esempio, la **Least Significant Difference** – $\boldsymbol{LSD}$ di Fisher, troppo liberale). Per scegliere quale usare, comunque, i principi sono pochi e piuttosto chiari: bisogna stabilire quanto la procedura controlla l'errore di I tipo, quanto controlla l'errore di II tipo, e se è affidabile nel caso in cui i requisiti di ANOVA siano violati. Bonferroni e Tukey controllano molto bene l'errore di I tipo, ma producono un tasso eccessivo di errori di II tipo; tra i due, Bonferroni è più potente per pochi confronti, Tukey lo è per molti, ed è più potente delle procedure di Dunn e Scheffé. Holm è più potente di Bonferroni, e Benjamini- Hochberg più di Holm.

Concludiamo con i **coefficienti di intensità dell'effetto** per i **singoli confronti a coppie**, contrasti o post hoc che siano, per cui non abbiamo a disposizione dagli output un risultato diretto. Potremo:

a) ricavare un coefficiente r dal quantile $\boldsymbol{t}$ ottenuto in ogni confronto ($df = N - numero\ di\ predittori$, ovvero variabili di contrasto);

$$r = \sqrt{\frac{t^2}{t^2 + df}}$$

b) ricorrere a un package specializzato, per esempio **`compute.es`**, che contiene la funzione `mes(media1=, media2=, sd1=, sd2=, n1=, n2=)`. La funzione restituisce l'entità della differenza, espressa come d di Cohen[100], g di Hedges, z e r di Pearson (più altri...).

[99] Abbiamo visto i `tibble` in Tecniche di analisi di dati I, §2.2.2

[100] Il denominatore del coefficiente d consiste nella radice quadrata della MS_R divisa per N.

12.2 Test non parametrici e ANOVA robusta

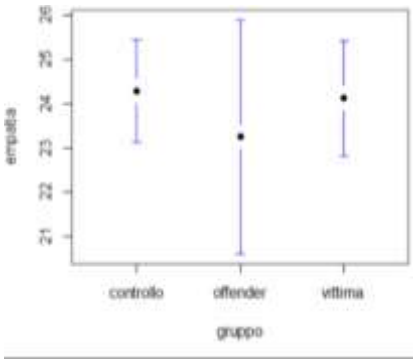
Naturalmente, anche con una X a più di due livelli devono essere rispettati i soliti prerequisiti di normalità, omoschedasticità e indipendenza degli errori, così come va evidenziata e trattata la presenza di casi influenti che danneggiano la generalizzabilità del modello.

```
summary(rstandard(schizo_gruppo))
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.31700 -0.54170 -0.04526  0.00000  0.77050  1.89800
summary(cooks.distance(schizo_gruppo))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
2.358e-05 1.163e-03 5.697e-03 1.259e-02 1.587e-02 1.193e-01
shapiro.test(residuals(schizo_gruppo))
       Shapiro-Wilk normality test
data:  residuals(schizo_gruppo)
W = 0.97768, p-value = 0.2144
bp.test(schizo_gruppo, studentize=FALSE)
       Breusch-Pagan test
data:  schizo_gruppo
BP = 3.4286, p-value = 0.1801
dw.test(schizo_gruppo, studentize=FALSE)
       Durbin-Watson test
data:  schizo_gruppo
DW = 2.0665, p-value = 0.5218
Alternative hypothesis: true autocorrelation is greater than 0
```

Nel caso in cui la **violazione sia limitata all'omoschedasticità,** e se occorre un'informazione limitata alla significatività dell'effetto di X su Y, si può usare la funzione di base **`oneway.test (Y~X, var= TRUE/FALSE)`**, che riporta solo F, df e $p-value$, ma consente di impostare con `var= FALSE` la **correzione di Welch** che già abbiamo incontrato nel t-test per campioni indipendenti .

Per esempio, vediamo se l'empatia differisce nei gruppi del dataframe `cp`:

```
tapply(cp$empatia_risposte_esatte, cp$gruppo, mean)
controllo  offender   vittima
 24.28571  23.25000  24.12500
shapiro.test(residuals(lm(cp$empatia_risposte_esatte~cp$gruppo)))
         Shapiro-Wilk normality test
data:  residuals(lm(cp$empatia_risposte_esatte ~ cp$gruppo))
W = 0.98197, p-value = 0.3724
dwt(lm(cp$empatia_risposte_esatte~cp$gruppo))
 lag Autocorrelation D-W Statistic p-value
   1      0.05122572      1.879619   0.474
 Alternative hypothesis: rho != 0

leveneTest(cp$empatia_risposte_esatte, cp$gruppo)
Levene's Test for Homogeneity of Variance
     Df  F value  Pr(>F)
group 2  3.5199  0.03487
     71
oneway.test(cp$empatia_risposte_esatte~cp$gruppo, var=FALSE)
      One-way analysis of means (not assuming equal variances)
Data: cp$empatia_risposte_esatte and cp$gruppo
F = .028107, num df = 2.000, denom df = 32.129, p-value = 0.7568
```

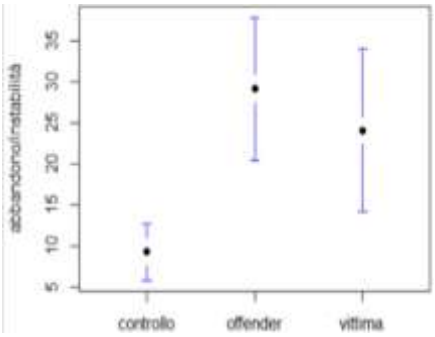


Sembra di no.

Altrimenti, possiamo servirci delle **analisi robuste di Wilcox**, contenute nel package **`WRS2`** in parte già esaminato nel capitolo precedente. Per esempio, abbiamo l'ANOVA robusta con un solo predittore: **`med1way(Y~X)`** confronta le

mediane dei gruppi ricavando una statistica che si distribuisce come un quantile F. Vediamo se uno schema del dominio Distacco/ rifiuto, ovvero **Abbandono/instabilità**, è significativamente diverso nei gruppi:

```
round(tapply(cp$ysq_abbandono_instabilita, cp$gruppo, mean), 1)
controllo  offender   vittima
      9.3      29.2      24.1

leveneTest(cp$ysq_abbandono_instabilita, cp$gruppo)
Levene's Test for Homogeneity of Variance
     Df  F value  Pr(>F)
group 2  4.019  0.02221
      71
med1way(cp$ysq_abbandono_instabilita~cp$gruppo)
Call:
med1way(formula = cp$ysq_abbandono_instabilita~cp$gruppo)

Test statistic: 11.5331
Critical value: 2.2359
p-value: 0
```



In alternativa, si possono confrontare le **medie trimmed** usando **`t1way(Y~X, data=, tr=)`**, che fornisce anche una misura **di effect size** (≅ .1: $debole$; ≅ .30: $medio$; ≅ .50 $forte$) e a cui è possibile **aggiungere un confronto post hoc** con la funzione **`lincon(Y~X, data=, tr=)`**.

```
t1way(ysq_abbandono_instabilita~gruppo, data= cp, tr = .2)
Call:
t1way(formula = ysq_abbandono_instabilita~$gruppo, data= cp, tr = 0.2)

Test statistic: 11.0284   ← ξ² = σ²(Ŷ) / σ²(Y)
Degrees of Freedom 1: 2
Degrees of Freedom 2: 13.97
p-value: 0.00134

Explanatory measure of effect size: 0.54
```

Attenzione: nel post hoc, il CI **è corretto per tener conto del family-wise error**, ma il $p-value$ **NO**: quindi, dobbiamo interpretare il CI:

```
lincon(ysq_abbandono_instabilita~gruppo, data= cp, tr = .2)
Call:
lincon(formula = ysq_abbandono_instabilita~$gruppo, data= offenders, tr = 0.2)
                            psihat[101]      ci.lower   ci.upper       p.value
controllo vs. offender    -23.73846        -39.71054   -7.76638       0.00171
controllo vs. vittima     -15.83846        -31.97391    0.29699       0.01958
offender vs. vittima        7.90000        -12.48229   28.28229       0.32333
```

Nel caso in cui sia la **normalità** a essere violata e una trasformazione non lineare di Y non rimedi al problema, o se Y è **ordinale**, si può utilizzare **l'Anova non parametrica a una via** (un predittore) di **Kruskal-Wallis** per k **gruppi indipendenti**: analogamente al test di Wilcoxon - Mann-Whitney per due gruppi indipendenti, di cui si può considerare un'estensione, lavora sui ranghi, prima assegnati a tutte le osservazioni indipendentemente dal gruppo, e poi sommati separatamente per gruppi. in R usiamo **`kruskal.test(Y~X)`**, in cui inseriamo `na.action= exclude` in presenza di NA.

```
shapiro.test(residuals(lm(cp$ysq_abbandono_instabilita~cp$gruppo)))
       Shapiro-Wilk normality test
data:  residuals(lm(cp$ysq_abbandono_instabilita ~ cp$gruppo))
W = 0.96134, p-value = 0.02332
kruskal.test(cp$ysq_abbandono_instabilita~cp$gruppo)
       Kruskal-Wallis rank sum test
data:  cp$ysq_abbandono_instabilita by cp$gruppo
```

[101] La statistica del test (psihat) è: $\widehat{\Psi} = \sum_{j=1}^{J} c_j \bar{X}_{tj}$

```
Kruskal-Wallis chi-squared = 19.409, df = 2, p-value = 6.101e-05
```

Come **indicatore di effect size** da associare a un test di Kruskal – Wallis, possiamo usare l'epsilon quadrato applicato ai ranghi (ε^2_{rank}, **rank epsilon squared**), che varia da 0 a 1.

$$\varepsilon^2_{rank} = \frac{\chi^2_{KW}}{N^2/N+1}$$

Il coefficiente è facilmente derivato dalla statistica χ^2 del test:

```
19.409/(((74^2)-1)/75)
[1] 0.2658767
```

In R possiamo usare anche `rank_epsilon_squared(formula)` di `effectsize`:

```
rank_epsilon_squared(cp$ysq_abbandono_instabilita~cp$gruppo)
Epsilon2 (rank) |       95% CI
--------------------------------
0.27            | [0.11, 0.47]
```

Una sorpresa: il test è prodotto nell'output di `Desc` (`DescTools`), quando il suo oggetto è di classe `formula` ($Y \sim X$, indipendentemente dal numero di livelli di X), insieme alle opportune descrittive, boxplot e grafico delle medie:

```
Desc(cp$ysq_abbandono_instabilita~cp$gruppo)
-------------------------------------------------------------------------
cp$ysq_abbandono_instabilita ~ cp$gruppo
Summary:
n pairs: 74, valid: 74 (100.0%), missings: 0 (0.0%), groups: 3
        controllo   offender    vittima
mean        9.262     29.188     24.125
median      8.000     36.500     28.000
sd         11.175     16.412     18.665
IQR        14.000     28.500     28.000
n              42         16         16
np         56.757%    21.622%    21.622%
NAs              0          0          0
0s              18          0          1

Kruskal-Wallis rank sum test:
  Kruskal-Wallis chi-squared = 19.409, df = 2, p-value = 6.101e-05
```

Anche con X a due livelli:

```
Desc(cp$ysq_abbandono_instabilita~cp$genere)
-----------------------------------------------------------------
cp$ysq_abbandono_instabilita ~ cp$genere
Summary:
n pairs: 74, valid: 74 (100.0%), missings: 0 (0.0%), groups: 2
               F        M
mean      16.270   17.297
median    12.000   11.000
sd        16.525   16.862
IQR       30.000   30.000
n             37       37
np        50.000%  50.000%
NAs             0        0
0s             10        9
```

```
Kruskal-Wallis rank sum test:
  Kruskal-Wallis chi-squared = 0.19999, df = 1, p-value = 0.6547
```

Se l'effetto del predittore risulta significativo secondo il test di Kruskal-Wallis, come nell'esempio, per i confronti post hoc si può usare il **test post hoc di Games - Howell** (vedi §6.1.1), oppure fare una **serie di test di** Wilcoxon / Mann-Whitney **per gruppi indipendenti con l'opportuna correzione per confronti multipli**, usando **`pairwise.wilcox.test(Y, X, prob.adjust.method)`**; le correzioni possibili sono le stesse utilizzate in `pairwise.t.test`.

```
pairwise.wilcox.test(cp$ysq_abbandono_instabilita, cp$gruppo, p.adjust.method = "b")
       Pairwise comparisons using Wilcoxon rank sum test
data:  cp$ysq_abbandono_instabilita and cp$gruppo

         controllo  offender
offender 0.00021    -
vittima  0.01195    0.68186

P value adjustment method: bonferroni
Warning messages:
1: In wilcox.test.default(xi, xj, paired = paired, ...) : impossibile calcolare p-value esatto in
   presenza di ties
2: In wilcox.test.default(xi, xj, paired = paired, ...) : impossibile calcolare p-value esatto in
   presenza di ties
3: In wilcox.test.default(xi, xj, paired = paired, ...) : impossibile calcolare p-value esatto in
   presenza di ties
```

Come facilmente predicibile dal grafico, i controlli attivano significativamente meno Abbandono/instabilità di vittime e offenders, tra loro, invece, non significativamente differenti. Notate il *warning*: per ogni confronto, non è possibile ottenere il $p - value$ esatto, dato che esistono ties nella distribuzione.

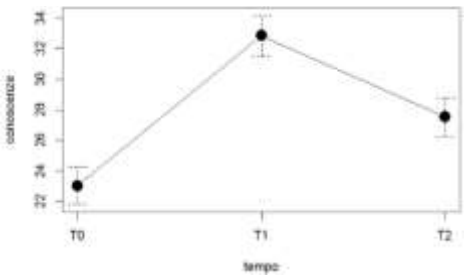
12.3 Test per disegni within subjects: ANOVA per un solo predittore a più di due livelli

12.3.1 ANOVA parametrica a misure ripetute

Fermo restando tutto quanto detto su contrasti e post hoc nel paragrafo precedente, valido anche nel caso dei disegni entro soggetti, rispetto all'ANOVA a misure ripetute per X a due livelli (§4.2.1) aggiungeremo un prerequisito assai importante: la **sfericità**.

Per una X a più di due livelli entro soggetti, la partizione della variabilità complessiva della misura (SS_T) avviene come già visto nel caso più semplice: l'effetto di X sarà valutato **entro i soggetti, distinguendolo dalla variabilità d'errore entro i soggetti**, e tenendo a parte la variabilità **tra** i soggetti. Ricordiamone i passaggi facendo un passo avanti nell'analisi dell'efficacia sul corso per la sicurezza nel lavoro, che avevamo lasciato (§4.2.1) ferma al confronto tra T_0 e T_1: aggiungiamo ora l'informazione sull'efficacia nell'incremento delle conoscenze (Y) a lungo termine (oltre tre mesi dalla fine del corso: T_2). Continuiamo a ignorare l'esistenza dei soggetti di controllo, per cui estraiamo un subset:

```
s<-subset(sicurezza, sicurezza$gruppo!="controllo", select =
   c(codice, conoscenze_t0, conoscenze_t1, conoscenze_t2))
names(s)<-c("sogg", "T0", "T1", "T2")medie<-c(mean(s$T0),
mean(s$T1), mean(s$T2))
sd<-c(sd(s$T0), sd(s$T1), sd(s$T2))
round(medie,2);round(sd,1)
[1] 23.03 32.84 27.54
[1] 5.8 6.4 6.0
```



Dopo un primo, significativo incremento a T_1, i partecipanti sembrano manifestare un calo della prestazione piuttosto brusco (i punteggi si abbassano significativamente) a T_2; per fortuna, le conoscenze restano significativamente superiori a quelle della baseline T_0, altrimenti avremmo proprio buttato via del tempo.

Calcoliamo *grand mean* e *grand variance*:

```
totale<-rbind(s$T0, s$T1, s$T2)
(grand_mean<-mean(totale))
[1] 27.8022
(grand_variance<-sd(totale)^2)
[1] 52.8872
```

Ricaviamo la **devianza totale** SS_T dalla varianza totale, con $df = N_{osservazioni} - 1$: $(91 * 3) - 1$

```
SS_T<-grand_variance*272
SS_T
[1] 14385.32
```

La **devianza entro i soggetti** SS_W rappresenta la **variabilità nella performance all'interno** di **ciascun soggetto** nel passare da una rilevazione all'altra: è perciò rappresentata dalla **somma** degli **scarti tra la prestazione del soggetto in ogni condizione e la media complessiva del soggetto stesso, al quadrato**.

```
s$totale<-s$T0+s$T1+s$T2
s$media<-s$totale/3
s$scarti<-(s$T0-s$media)^2+(s$T1-s$media)^2+(s$T2-s$media)^2

head(s,5)
     sogg T0 T1 T2     media totale    scarti
1  BC0Y29 34 42 30 35.333333    106  74.66667
2  BG3M14 28 39 33 33.333333    100  60.66667
3  BL9D18 32 41 32 35.000000    105  54.00000
4  BR3E44 21 36 29 28.666667     86 112.66667
5  BR3P27  7 11 11  9.666667     29  10.66667
6  BR3S10 15 28 22 21.666667     65  84.66667
7  BZ8W24 19 30 24 24.333333     73  60.66667
8  CC3I22 20 33 25 26.000000     78  86.00000
9  CR6G28 20 37 27 28.000000     84 146.00000
10 CS8N06 15 28 20 21.000000     63  86.00000
```

Da questo piccolo estratto del campione, sembra che i soggetti siano molto diversi tra loro nel modo in cui le loro conoscenze variano nel tempo: i loro scarti vanno da molto piccoli (il soggetto 5 sa abbastanza poco, ma è molto costante nel tempo) a molto grandi (le conoscenze dei soggetti 4 e 9 sono molto incoerenti tra le rilevazioni). Calcoliamo la somma degli scarti e otteniamo la SS_W:

```
(SS_W<-sum(s$scarti))
[1] 6213.333
```

Calcoliamo la **devianza del modello** SS_M ($df = k - 1$): facciamo lo **scarto** al quadrato **della media di ogni condizione** (tempo) **dalla grand mean**, lo moltiplichiamo **per il numero di soggetti** di ogni condizione e sommiamo:

```
(SS_M<-(91*((mean(s$T0)-grand_mean)^2)+91*((mean(s$T1)-grand_mean)^2) + 91*((mean(s$T2)-
  grand_mean)^2)))
[1] 4381.275
```

Da SS_W e SS_M ricaviamo per differenza la quota di errore SS_R, con $df = (N - 1) \times (k - 1)$:

```
(SS_R<-SS_W-SS_M)
[1] 1832.059
```

e, anche se non ci servirà, la devianza attribuita alle differenze tra i soggetti SS_B, che delinea l'eterogeneità del campione reclutato, come differenza tra devianza totale e devianza entro i soggetti:

```
(SS_B<-SS_T-SS_W)
[1] 8171.985
```

Ora ricaviamo le varianze del modello MS_M e di errore MS_R, per avere il relativo rapporto F:

```
(MS_M<-SS_M/(3-1))
[1] 2190.637
(MS_R<-SS_R/((91-1)*(3-1)))
[1] 10.1781
(F<-MS_M/MS_R)
[1] 215.2304
```

Calcoliamo la probabilità di ottenere questo rapporto F o uno più grande sotto condizione di ipotesi nulla:

```
pf(q = F, df1 = 1, df2 = 180, lower.tail = FALSE)
```

```
[1] 1.451089e-32
```

Evento decisamente eccezionale, direbbe Fisher; assecondiamolo, rifiutando H_0: esiste una differenza non casuale, in popolazione, nei punteggi delle conoscenze da T_0 a T_2.

Attenzione, però: non abbiamo verificato l'assunto di **sfericità** (o **circolarità**). L'assunto di sfericità è rispettato quando **le varianze delle differenze tra tutte le coppie delle misure ripetute sono uguali**; si applica, per evidenti motivi, quando le misure ripetute sono almeno tre. La sfericità è stimata dal coefficiente **epsilon** ($\boldsymbol{\varepsilon}$): una sfericità perfetta è descritta da $\boldsymbol{\varepsilon = 1}$. Tra le varie stime di sfericità, le più usate sono la stima $\tilde{\boldsymbol{\varepsilon}}$ di **Greenhouse-Geisser** o la stima $\acute{\boldsymbol{\varepsilon}}$ **di Huyn – Feldt**: la prima è generalmente meno generosa della seconda, che tende a sovrastimare la sfericità.

La sfericità è un caso particolare di **simmetria composta**, che si ha quando, oltre ad avere uguali varianze delle differenze tra le coppie, si hanno anche **uguali covarianze** tra le coppie di misure.

$$\sum \begin{vmatrix} s_{11}^2 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & s_{22}^2 & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & s_{33}^2 & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & s_{nn}^2 \end{vmatrix}$$

Gli elementi sulla diagonale della matrice di varianza – covarianza sono le varianze di ogni livello, quelli fuori dalla diagonale, simmetricamente, sono le covarianze tra ogni coppia di livelli, che nella sfericità possono essere indifferentemente uguali o diverse (ricordiamo che le covarianze rappresentano le relazioni non standardizzate tra distribuzioni).

*Nella condizione di **simmetria composta**, invece, sia le varianze sia le covarianze, cioè le relazioni tra i livelli, sono uguali:*

$$\boldsymbol{s_{11}^2 \cong s_{22}^2 \cong s_{33}^2 \cong \cdots s_{nn}^2; a_{12}^2 \cong a_{13}^2 \cong a_{21}^2 \cong a_{23}^2 \cong a_{31}^2 \cong a_{32}^2 \cong \cdots a_{nn}^2}$$

Nella sfericità, condizione meno restrittiva della simmetria composta, solo le varianze delle differenze tra le coppie delle condizioni devono essere non significativamente differenti. Vediamo le tre varianze delle differenze T_0-T_1, T_0-T_2, T_1-T_2:

```
var(s$T0-s$T1);var(s$T0-s$T2); var(s$T1-s$T2)
[1] 21.13822
[1] 20.0083
[1] 19.9221
```

Il dato sembra molto promettente: le tre varianze delle differenze fra i tempi sono **molto simili tra loro**. L'H_0 di uguaglianza tra le varianze delle differenze può essere valutata con il **test di Mauchly**, la cui statistica $\boldsymbol{W}$ si distribuisce come un quantile χ^2 per $df = N_{misure\ ripetute}$: analogamente al test di Levene, **se $\boldsymbol{p > .05}$** possiamo assumere che la **sfericità è rispettata** (le differenze tra le varianze non sono significative), mentre se $p < .05$ la sfericità è significativamente violata. Naturalmente, anche questo test risente dei soliti problemi di potenza: pochi soggetti portano a confermare la sfericità anche in presenza di differenze tra varianze non trascurabili, mentre molti soggetti portano a rifiutarla nonostante le differenze siano di trascurabile entità.

La **violazione** della sfericità fa sì che il rapporto F ottenuto non sia correttamente interpretabile, dato che non corrisponde effettivamente ai quantili della distribuzione F per i df previsti, **aumentando il rischio d'incorrere in un errore di I tipo**: in questo caso, perciò, si rimedia alla **violazione della sfericità** correggendo i gradi di libertà di $\boldsymbol{F}$, che vengono moltiplicati per le stime $\tilde{\boldsymbol{\varepsilon}}$ di **Greenhouse-Geisser** o $\acute{\boldsymbol{\varepsilon}}$ **di Huyn – Feldt**. Naturalmente, se la sfericità è perfetta, e quindi $\varepsilon = 1$, i $\boldsymbol{df \times 1}$ **restano immutati**. Tanto più è grave la violazione della sfericità, e quindi tanto più $\varepsilon < 1$, più i df moltiplicati per questa stima ne verranno rimpiccioliti: come sappiamo, uno stesso quantile in distribuzioni con df minori si vedrà assegnato un $p - value$ più alto, per cui sarà più difficile respingere H_0, rimediando al bias indotto dalla violazione. Secondo Girden (1992), prudenzialmente, se $\boldsymbol{\acute{\varepsilon} \geq .75}$ (violazione non grave), allora è meglio correggere i df usando la stima di Huynh e Feldt; invece, se $\boldsymbol{\acute{\varepsilon} < .75}$ (violazione via via sempre più grave), è meglio usare la prudente stima di Greenhouse-Geisser. Stevens (1992) propone di usare una salomonica media delle due stime, e R propone entrambe le correzioni, lasciandoci la responsabilità di scegliere quella più adatta alla situazione.

Attenzione: lo stesso bias che ci obbliga a correggere i df del rapporto F nel modello overall ricade sui **contrasti** a **priori** e sui test **post hoc** applicati al predittore: per i post hoc, è meglio usare la correzione di Bonferroni quando la violazione è grave, e il test post hoc di Tukey quando non lo è.

Attenzione: si può ignorare il problema della sfericità usando un **modello che consente di fare regressioni quando le osservazioni sono dipendenti**, ovvero **correlate**, come accade nelle misure ripetute: i **mixed models** (o **multilevel models**, Capitolo 9).

Per ora, useremo il metodo "facile" con **ezANOVA** del package **ez**, già vista nel §4.2.1: quando X a misure ripetute ha più di due livelli, in output si manifestano **anche il test di Mauchly**, le stime di **Greenhouse – Geisser** e **Huyn-Feldt** e i **relativi $p-value$ di F** per i df corretti, per cui la funzione ci toglie un bel po' di preoccupazioni.

Vediamola applicata al problema delle conoscenze per cui abbiamo calcolato F "a mano", ma di cui ignoriamo la sfericità. Il primo passo è, non sorprendentemente, passare al long format, usando `melt`:

```
melt_conosc<-melt(data=s, id.vars="sogg", measure.vars=c("T0", "T1", "T2"))
names(melt_conosc)<-c("sogg", "tempo", "conoscenze")
```

```
 melt_conosc[1:5,]            melt_conosc[92:96,]           melt_conosc[183:187,]
     sogg tempo conoscenze         sogg tempo conoscenze         sogg tempo conoscenze
1 BC0Y29    T0         34   92 BC0Y29    T1         42   183 BC0Y29    T2         30
2 BG3M14    T0         28   93 BG3M14    T1         39   184 BG3M14    T2         33
3 BL9D18    T0         32   94 BL9D18    T1         41   185 BL9D18    T2         32
4 BR3E44    T0         21   95 BR3E44    T1         36   186 BR3E44    T2         29
5 BR3P27    T0          7   96 BR3P27    T1         11   187 BR3P27    T2         11
```

I contrasti semplici preimpostati sono sensati per le nostre ipotesi, dato che il la baseline T_0 è confrontata con i tempi successivi; poi vedremo i post hoc. Passiamo a **ezANOVA(data, wid= soggetti, dv= Y, within= X a misure ripetute, detailed= FALSE)**: indichiamo `detailed= FALSE` solo perché non ci interessa l'intercetta.

```
ezANOVA(data = melt_conosc, dv = conoscenze, wid = sogg, within = tempo, detailed= FALSE)
```

L'output è diviso in tre sezioni: nell'ordine, la significatività del modello non corretta, il test di Mauchly, le stime di Greenhouse- Geisser e Huyn – Feldt, con il $p-value$ del predittore corretto. Dato che la valutazione della sfericità precede logicamente quella della significatività dell'effetto, iniziamo da questa:

```
$'Mauchly's  Test for Sphericity'
   Effect          W      p p<.05
2  tempo    0.9985182   0.9361415
                  ↑
```

Evitate un errore comune all'esame: la statistica W ***non è una stima epsilon!***

Confermiamo l'impressione sulle tre varianze delle differenze: la **sfericità è confermata**, le varianze **non sono significativamente differenti**. Vediamo l'effetto del predittore tempo):

```
$ANOVA
  Effect DFn DFd        F       p p<.05           ges
2  tempo   2 180 215.2304 1.842145e-48     * 0.3045657
```

Rispetto all'output di `aov`, non vengono mostrate MS_M e MS_R, ma solo i df del modello e di errore (`DFn` e `DFd`): troviamo i valori già visti, compreso il $p-value$ del rapporto $F_{[2;180]} = 215.23$, largamente inferiore alla soglia di rifiuto di H_0, per cui affermiamo che esiste una differenza significativa nel livello di conoscenze nei tre tempi rilevati. Abbiamo già visto il coefficiente `ges` (*generalized eta squared*): il tempo spiega il 30.4% della variabilità delle conoscenze.

L'ultima sezione riporta il $p-value$ del rapporto F con i df corretti per la violazione, che dovremmo leggere nel caso in cui il test di Mauchly sia significativo:

```
$'Sphericity Corrections'
```

```
  Effect     GGe p[GG]    p[GG]<.05       HFe p[HF]     p[HF]<.05
2  tempo     0.9985204  2.152325-48     *  1.021163   1.842145e-48*
```

La stima della sfericità di Greenhouse-Geisser (Greenhouse-Geisser estimate: GGe) è praticamente $\varepsilon = 1$, e quella più liberale di Huyn-Feldt (HFe) è decisamente $\varepsilon = 1$. I *p*-value del predittore Tempo corretto secondo Greenhouse – Geisser e secondo Huyn-Feldt sono praticamente uguali a quello non corretto – ovviamente, dato che i df del rapporto F sono moltiplicati per 1.02 e .999, quindi restano praticamente identici.

Possiamo verificare come è stato ottenuto il $p - value$ corretto secondo GG_ε: i df corretti sono

```
(df_M_GGe <- 2 * .9985204)
[1] 1.997041
(df_R_GGe <- 180 * .9985204)
[1] 179.7337
```

e il $p - value$ associato a un quantile $F \geq 215.2304$, in una distribuzione F con i df corretti è:

```
pf(215.2304, df1 = df_M_Gge, df2 = df_R_GGe, lower.tail = FALSE)
[1] 2.152319e-48
```

Poiché l'effetto è significativo, dobbiamo fare i confronti a coppie: facciamo i test post hoc con **pairwise.t.test**, ricordandoci di specificare che sono **dati dipendenti** (**paired= TRUE**):

```
pairwise.t.test(melt_conosc$conoscenze,melt_conosc$tempo,paired = TRUE, p.adjust.method = "b")
         T0      T1
T1  < 2e-16       -
T2  5.6e-15  <2e-16
```

il miglioramento da T_0 a T_1 è significativo, ma, purtroppo, anche il peggioramento da T_1 a T_2 lo è. Per fortuna, la differenza tra T_0 e T_2 è anch'essa significativa, quindi le conoscenze al follow up, pur calate, sono ancora significativamente superiori a quelle possedute dai soggetti a T_0 (però, chi sa cosa sarebbe successo se avessimo fatto un T_3 dopo altri tre mesi…).

Verificate se lo stesso tipo di variazione si manifesta anche per le altre misure: la gravità dei comportamenti a rischio, gli atteggiamenti positivi verso l'auto-tutela, le conseguenze sulla salute.

Quale suggerimento potreste dare allo psicologo del lavoro che progetta corsi sulla sicurezza, alla luce di questi risultati?

Potremmo eseguire ANOVA a misure ripetute anche adattando la funzione **aov** alla natura entro soggetti dei dati: si dovrà inserire nel modello il termine di **errore within subjects** aggiungendo **Error($soggetto/fattore within)**. In R, lo slash "/" indica che il termine che precede "/" è **nidificato** nel termine che lo segue: infatti, nel disegno a misure ripetute il soggetto è nidificato (***nested***) nel tempo (ne riparleremo nel capitolo 15). R calcola il termine di errore principale $soggetto (che non ci interessa: è la SS_R **del parametro intercetta**) e l'interazione $soggetto* $fattore within, che produce il termine di errore within corretto, usato per ricavare il rapporto F. Nel nostro esempio:

```
summary(aov(melt_s$rilevazioni~melt_s$tempo+Error(melt_s$sogg/melt_s$tempo)))
Error: melt_s$sogg
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 90   8172    90.8

Error: melt_s$sogg:melt_s$tempo
              Df Sum Sq Mean Sq F value Pr(>F)
melt_s$tempo   2   4381  2190.6   215.2 <2e-16
Residuals    180   1832    10.2
```

... e infatti in ezANOVA abbiamo trovato:

```
$ANOVA
       Effect   DFn DFd         SSn       SSd         F
2       tempo   2  180    4381.275  1832.059  215.23041
```

Tuttavia, questa procedura presenta alcuni svantaggi: non riporta il test di Mauchly né la stima della sfericità e la correzione per la sua violazione, che andrebbero quindi eseguiti a parte, né riporta un coefficiente di effect size. Useremmo quindi questa funzione nel solo caso particolare **dell'analisi della covarianza a misure ripetute** (ma, vedi §13.3…).

12.3.2 Test non parametrici e ANOVA robusta misure ripetute

Se la violazione della sfericità può essere corretta entro l'ANOVA a misure ripetute, cosa fare se è la **normalità** a essere **violata**, o se Y è **ordinale**?

Possiamo usare **l'Anova non parametrica di Friedman**, che è un'estensione del test di Wilcoxon per dati appaiati al caso di una X a misure ripetute a più di due livelli ed usa la **distribuzione di probabilità** χ^2. La funzione di base `friedman.test(matrice)` restituisce un semplice output e lavora su matrici, costituite dalle sole misure ripetute in formato *wide*. Applichiamola alle conoscenze:

```
friedman_conosc<-as.matrix(s[,11:13])
friedman.test(friedman_conosc)
```

```
        Friedman rank sum test
data:  friedman_conosc
Friedman chi-squared = 146.16, df = 2, p-value < 2.2e-16
```

Per applicare un **metodo robusto**, ovvero un'Anova non sensibile alle violazioni dei requisiti, ricorriamo al solito package `WRS2` e usiamo `rmanova(Y, groups=X, blocks= soggetto, trimmed)`, che considera i valori *trimmed* (richiede che il formato del dataframe sia *long*); gli argomenti possono sembrare un po' stravaganti, dato che in `groups=` si indica la X a misure ripetute e in `blocks=` la variabile che identifica il soggetto.

```
rmanova(y = melt_conosc$conoscenze, groups = melt_conosc$tempo, blocks = melt_conosc$sogg, tr =
.20)
```

```
Call:
rmanova(y = melt_conosc$conoscenze, groups = melt_conosc$tempo,
blocks =melt_conosc$sogg, tr = .20)

Test statistic: 182.8665
Degrees of Freedom 1: 1.94
Degrees of Freedom 2: 104.54
p-value: 0
```

L'output è sintetico: indica il $p-value$ ($p < .001$) del quantile ($F = 182.87$) in una distribuzione F con i due $df = 1.94;\ 104.54$: nuovamente, la differenza tra i tempi è significativa.

Per i post hoc robusti a misure ripetute serve `rmcp`, con i medesimi argomenti di `rmanova`:

```
rmmcp (y = melt_conosc$conoscenze, groups = melt_conosc$tempo, blocks = melt_conosc$sogg, tr =
   .20)
```

```
Call:
rmmcp (y = melt_conosc$conoscenze, groups = melt_conosc$tempo, blocks =melt_conosc$sogg, tr = .2
0)
             psihat   ci.lower   ci.upper p.value  p.crit  sig
T0 vs. T1  -9.36364  -10.38474   -8.34253       0  0.0250  TRUE
T0 vs. T2  -4.12727   -5.13297   -3.12157       0  0.0500  TRUE
T1 vs. T2   4.29091    4.26814    6.61368       0  0.0169  TRUE
```

In tutti i confronti il CI della differenza tra le medie *trimmed* non comprende $H_0 = 0$, quindi nei tempi le conoscenze sono significativamente diverse. Ridondantemente, se il $p-value$ è maggiore di $p_{critico}$ (`p.crit`), la differenza non è significativa.

Capitolo 13.
Regressione con più predittori categoriali: ANOVA per disegni fattoriali

In questo capitolo useremo i dataframe ***ep, simon*** *e* ***sicurezza*** *pubblicati su Elly: apriteli prima di proseguire*

"The results of interaction effects are probably the universally most misinterpreted empirical results in psychology".
Rosnow and Rosenthal (1989, pag. 1282)

In questo capitolo vedremo come adattare un modello lineare (ma chiameremo l'analisi ANOVA, come da tradizione) al caso in cui una sola Y viene predetta da due o più predittori categoriali. L'ANOVA per disegni fattoriali, o per brevità **ANOVA fattoriale**[102], può essere between groups, se tutti i predittori del modello sono between groups, o within subjects, se tutti i predittori sono a misure ripetute, o mista, se almeno un predittore è somministrato between groups e almeno un altro predittore è somministrato within subjects. Per semplicità di esposizione, analizzeremo prima il caso più semplice, con due soli predittori: **ANOVA a due vie**; vedremo un modello con tre predittori categoriali nell'ANOVA mista (§13.3) e con più predittori categoriali e continui nell'analisi della covarianza (§13.4).

Il modello può contenere solo i coefficienti relativi a ciascun predittore, nel qual caso si verifica **l'effetto principale** di ciascuno, cioè l'effetto del predittore X_1 indipendentemente dall'effetto che il predittore X_2 ha su Y (e, ovviamente, viceversa): è il tipo di modello lineare additivo che abbiamo approfondito nella regressione lineare multipla.

Più frequente, e più utile, è però il modello lineare che oltre agli affetti principali di ogni X contiene anche il coefficiente angolare che esprime l'effetto della **interazione** tra X_1 e X_2, come abbiamo rapidamente anticipato nella regressione multipla con più X continue. L'**effetto d'interazione** è l'effetto **esercitato da X_1 su Y a seconda del livello della variabile X_2** e viceversa (Keppel, 1991: "An interaction is present when the effects of one independent variable on behavior change at the different levels of the second independent variable", pag. 196): ciò significa che appartenere all'uno o all'altro livello di X_2 fa variare l'effetto di X_1 su Y, così come appartenere all'uno o all'altro livello di X_1 fa variare l'effetto di X_2 su Y.

Per esempio, frequentare (X_{1a}) le lezioni di Tecniche di analisi di dati II determina in media un voto più alto all'esame (Y) rispetto a non frequentare (X_{1b}), ma solo per chi non fa esercizi a casa (X_{2b}), mentre chi fa esercizi a casa (X_{2a}) ottiene in media lo stesso voto sia che frequenti (X_{1a}), sia che non frequenti (X_{1b}).

In pratica, quando le differenze in Y tra le condizioni sperimentali non sono spiegate solo dagli effetti principali e dell'errore, queste differenze non attribuite sono attribuite all'interazione tra le X: per questo motivo le **interazioni sono anche definite effetti residuali** [da non confondere con i residui, cioè gli errori!], ovvero gli effetti che restano dopo che sono stati rimossi dal modello gli effetti di ordine inferiore, cioè gli effetti principali (Rosenthal e Rosnow, 1984; Rosnow e Rosenthal, 1991).

Approfondiamo rapidamente il modello matematico sottostante il disegno fattoriale con interazione, prima di procedere alla pratica.

[102] Attenzione a non confondere questo tipo di analisi della varianza con l'analisi fattoriale, che è una famiglia di tecniche statistiche di riduzione della matrice delle correlazioni tra variabili – e che quindi non c'entra proprio nulla.

Secondo il modello:

$$\boldsymbol{y_{ijk} = \left(b_0 + b_1X_i + b_2X_j + b_3X_iX_j\right) + e_{ijk}}$$

ogni osservazione y_{ij} può essere ripartita tra cinque componenti (Marascuilo e Levin, 1970):

- $\boldsymbol{b_0}$: intercetta ovvero **grand mean** $\boldsymbol{\mu}$,
- $\boldsymbol{b_1X_i}$: **effetto principale di** $\boldsymbol{X_i}$, dato da $\boldsymbol{grand\ mean - \bar{x}_i}$, cioè dalla differenza tra la media complessiva del campione e la media di tutti i casi appartenenti allo stesso livello i di $\boldsymbol{X_i}$: $\boldsymbol{\mu_{i-}\mu}$;
- $\boldsymbol{b_2X_j}$: **effetto principale di** $\boldsymbol{X_j}$, dato da: $\boldsymbol{grand\ mean - \bar{x}_j}$, cioè dalla differenza tra la media complessiva del campione e la media di tutti i casi appartenenti allo stesso livello j di $\boldsymbol{X_j}$: $\boldsymbol{\mu_{j-}\mu}$;
- $\boldsymbol{b_3X_iX_j}$: **effetto di interazione**: la **media di tutti i casi della condizione** $\boldsymbol{X_iX_j}$, **meno la media** di tutti i casi appartenenti al livello $\boldsymbol{X_i}$, **meno la media** di tutti i casi appartenenti al livello $\boldsymbol{X_j}$, **più la** $\boldsymbol{grand\ mean}$: $\boldsymbol{\mu_{ij} - \mu_i - \mu_j + \mu}$;
- $\boldsymbol{e_{ijk}}$ = **errore**.

In **un disegno bilanciato ciascuna di queste componenti è perfettamente indipendente** (non correlata) dalle altre (Hester,1996): perciò, conoscere gli effetti principali non dà alcuna informazione sull'interazione.

La verifica della significatività dell'interazione testa l'ipotesi nulla che gli **effetti d'interazione per ogni condizione sperimentale siano uguali a zero**: vediamolo con un esempio (Graham, 2000), relativo a due variabili indipendenti (A e B) a due livelli (a_1, a_2; b_1, b_2). Conosciamo le **medie di ogni cella** (le **quattro condizioni** sperimentali) e le **medie marginali, cioè le medie delle condizioni di ogni livello; calcoliamo anche la grand mean:**

		B		
		b_1	b_2	**Media marginale μ_a**
A	a_1	**7**	**5**	$(7+5)/2 = 6$
	a_2	**5**	**3**	$(5+3)/2 = 4$
	Media marginale μ_b	$(7+5)/2 = 6$	$(5+3)/2 = 4$	
		$\boldsymbol{grand\ mean}$**:** $(6+4+6+4)/4 = 5$		

Aggiungiamo gli effetti principali, cioè le **differenze tra le medie marginali e la grand mean**, per ogni livello di A e B:

		B			
		b_1	b_2	**Media marginale μ_a**	**Effetto di A α**
A	a_1	**7**	**5**	$(7+5)/2 = 6$	$6 - 5 = +1$
	a_2	**5**	**3**	$(5+3)/2 = 4$	$4 - 5 = -1$
	Media marginale μ_b	$(7+5)/2 = 6$	$(5+3)/2 = 4$		
	Effetto di B β	$6 - 5 = +1$	$4 - 5 = -1$		
		$\boldsymbol{grand\ mean}$: $(6+4+6+4)/4 = 5$			

L'ipotesi nulla per A è che la differenza tra la media marginale a_1 e la $grand\ mean$ è uguale alla differenza fra la media marginale di a_2 e la $grand\ mean$: $\boldsymbol{H_0: \mu_{a1} - \mu = \mu_{a2} - \mu}$. L'ipotesi nulla per B è che la differenza tra la media marginale di b_1 e la $grand\ mean$ sia uguale alla differenza fra la media marginale di b_2 e la $grand\ mean$: $\boldsymbol{H_0: \mu_{b1} - \mu = \mu_{b2} - \mu}$.

Osserviamo che **per questi dati** la **media di ogni condizione può essere ricavata esattamente** conoscendo la $grand\ mean$ e gli effetti principali: verifichiamolo aggiungendo alla $grand\ mean$ l'effetto di A e di B per ciascun livello.

$media_{a1b1} = \boldsymbol{\mu + \alpha_1 + \beta_1} = 5 + (+1) + (1) = 7$ $\quad$ $media_{a1b2} = \boldsymbol{\mu + \alpha_1 + \beta_2} = 5 + (+1) + (-1) = 5$

$media_{a2b1} = \boldsymbol{\mu + \alpha_2 + \beta_1} = 5 + (-1) + (1) = 5$ $\quad$ $media_{a2b2} = \boldsymbol{\mu + \alpha_2 + \beta_2} = 5 + (-1) + (-1) = 3$

Perciò, dato che conoscendo $grand\ mean$ (b_0), effetto principale di A (b_1X_1) ed effetto principale di B (b_1X_2) siamo perfettamente in grado di predire la media di ciascuna condizione, **non servono altri effetti (effetti residuali)**: in questo modello gli effetti di **interazione sono** $= \mathbf{0}$, quindi l'interazione non è significativa. Infatti, se calcoliamo gli effetti di interazione come $\boldsymbol{\mu_{ij} - \mu_i - \mu_j + \mu}$, abbiamo:

```
7-6-6+5
[1] 0
5-4-6+5
[1] 0
```

```
5-6-4+5
[1] 0
3-4-4+5
[1] 0
```

Vediamo invece questo diverso esempio:

		B			
		b_1	b_2	**Media marginale μ_a**	**Effetto di A α**
A	a_1	**2**	**4**	$(2+4)/2 = \mathbf{3}$	$3 - 4.5 = \mathbf{-1.5}$
	a_2	**8**	**4**	$(8+4)/2 = \mathbf{6}$	$6 - 4.5 = \mathbf{+1.5}$
	Media marginale μ_b	$(2+8)/2 = \mathbf{5}$	$(4+4)/2 = \mathbf{4}$		
	Effetto di B β	$5 - 4.5 = \mathbf{+.5}$	$4 - 4.5 = \mathbf{-.5}$		

$\boldsymbol{grand\ mean}$: $(\mathbf{3+6+5+4})/\mathbf{4} = \mathbf{4.5}$

Verifichiamolo se anche in questo caso, conoscendo grand mean ed effetti principali, possiamo ricavare correttamente le medie delle condizioni.

$media_{a1b1} = \boldsymbol{\mu + \alpha_1 + \beta_1} = 4.5 + (-1.5) + (+.5) = 3.5$ $\quad$ $media_{a1b2} = \boldsymbol{\mu + \alpha_1 + \beta_2} = 4.5 + (-1.5) + (-.5) = 2.5$

$media_{a2b1} = \boldsymbol{\mu + \alpha_2 + \beta_1} = 4.5 + (+1.5) + (+.5) = 6.5$ $\quad$ $media_{a2b2} = \boldsymbol{\mu + \alpha_2 + \beta_2} = 4.5 + (+1.5) + (-.5) = 5.5$

In nessuna cella troviamo il vero valore medio: quello che manca per far tornare il modello, il "rimasuglio" di effetto che serve per soddisfare l'equazione, è **appunto l'effetto residuale** attribuito all'interazione:

effetto di interazione $a_1b_1 = 2 - \mathbf{3.5} = \mathbf{-1.5}$ $\quad$ effetto di interazione $a_1b_2 = 4 - \mathbf{2.5} = \mathbf{+1.5}$

effetto di interazione $a_2b_1 = 8 - \mathbf{6.5} = \mathbf{+1.5}$ $\quad$ effetto di interazione $a_2b_2 = 4 - \mathbf{5.5} = \mathbf{-1.5}$

Ovvero, usando $\boldsymbol{\mu_{ij} - \mu_i - \mu_j + \mu}$ come sopra descritto:

```
2-3-5+4.5
[1] -1.5
8-6-5+4.5
[1] 1.5
```

```
4-3-4+4.5
[1] 1.5
4-6-4+4.5
[1] -1.5
```

Perciò, quando l'interazione è significativa le medie delle condizioni sono gli effetti combinati dell'interazione, di $\boldsymbol{X_1}$, di $\boldsymbol{X_2}$ e della $grand\ mean$ (Rosnow e Rosenthal, 1989).

13.1 ANOVA fattoriale between groups

Inizieremo, come tradizione, dall'applicazione dell'ANOVA a disegni between groups con almeno due predittori.
Per concretizzare, usiamo il dataframe **ep**, che contiene i dati di un gruppo di **pazienti con eiaculazione precoce** (EP) e di un gruppo di controllo: conosciamo le loro risposte al test **IIEF** (visto nel precedente capitolo su altri soggetti, qui presente anche con le sue sottoscale), il loro punteggio alla scala temperamentale **Harm Avoidance**, ormai ben nota, e alcune **variabili socio anagrafiche**. È anche riportato il loro profilo allelico relativo al **gene 5-HTTLPR**: omozigote L/L (Long/Long), omozigote S/S (Short/Short), eterozigote L/S (Long/Short). Da letteratura, i portatori della variante S

mostrano iperattivazione dell'amigdala e, coerentemente, iper-reattività emotiva, con aumentato rischio di sviluppare disturbi dell'umore. Sembra, inoltre, che la variante S predisponga a maggior HA. Concentriamoci, per ora, sulla sottoscala del desiderio ($\boldsymbol{Y}$).

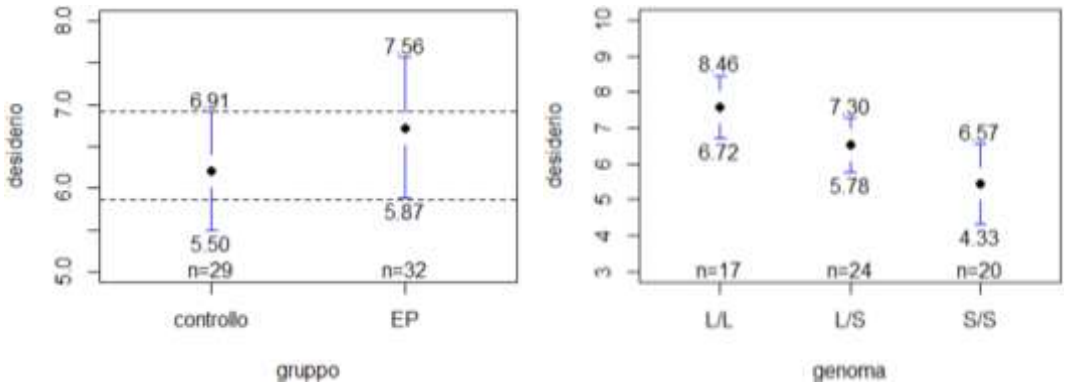
Il modello prevede **tre ipotesi nulle**, di cui due sono relative agli effetti principali:

- $H_{0\ Gruppo}$: **indipendentemente dal profilo genetico, il gruppo non esercita alcun effetto sul desiderio dei soggetti** → indipendentemente dal profilo genetico, controlli e pazienti appartengono alla medesima popolazione per la $Y_{desiderio}$;
- $H_{0\ Gene}$: **indipendentemente dal gruppo di appartenenza, il gene 5-HTTLPR non esercita alcun effetto sul desiderio dei soggetti** → indipendentemente dal gruppo, i soggetti L/L, L/S e S/S appartengono alla medesima popolazione per la $Y_{desiderio}$.

La terza H_0 si riferisce all'interazione:

- $H_{0\ Interazione}$: la **differenza** nel desiderio tra Controlli ed EP con genoma L/L è uguale alla differenza tra Controlli ed EP con genoma L/S e alla differenza tra Controlli ed EP con genoma S/S, **OPPURE:** le differenze nel desiderio tra L/L, L/S e S/S rilevate nel gruppo di controllo sono uguali alle differenze nel desiderio tra L/L, L/S e S/S rilevate nel gruppo di EP.

Cominciamo a descrivere i livelli dei fattori: il desiderio è diverso nei due **gruppi** ($\boldsymbol{X_1}$)? Ed è diverso a seconda del **profilo allelico** dei soggetti ($\boldsymbol{X_2}$)?



Il disegno è ben bilanciato per gruppo, un po' meno bene, ma non disastrosamente, per genoma.

```
gruppo<-tapply(ep$desiderio_sex, ep$gruppo, mean)
gruppo[1];gruppo[2]                    → Tenete d'occhio queste medie
controllo
 6.206897
      EP
6.71875

genoma<-tapply(ep$desiderio_sex, ep$genoma, mean)
genoma[1];genoma[2];genoma[3]          → Tenete d'occhio queste medie
      L/L
7.588235
      L/S
6.541667
 S/S
5.45
```

Le differenze medie per gruppo, indipendentemente dal genoma (effetto principale di X_1) sembrano molto ridotte, quelle per genoma, indipendentemente dal gruppo di appartenenza (effetto principale di X_2) sono più apprezzabili.

```
round(tapply(ep$desiderio_sex, ep$gruppo, sd),1)
controllo        EP
        1.9       2.3
round(tapply(ep$desiderio_sex, ep$genoma, sd),1)
L/L      L/S      S/S
1.7      1.8      2.4
```

Passiamo ora **all'interazione**, per cui dobbiamo **incrociare** l'appartenenza di ogni caso per gruppo e genoma: avremo **6 medie**, corrispondenti alle 6 condizioni del disegno di ricerca (3×2):

		Genoma X_2		
		L/L_a	**L/S_b**	**S/S_c**
GRUPPO X_1	**Controlli$_a$**	$x_{1a}x_{2a}$	$x_{1a}x_{2b}$	$x_{1a}x_{2c}$
	EP$_b$	$x_{1b}x_{2a}$	$x_{1b}x_{2b}$	$x_{1b}x_{2c}$

Vediamo le numerosità dei 3×2 gruppi:

```
table(ep$gruppo,ep$genoma)
            L/L   L/S   S/S
  controllo   7    10    12
  EP         10    14     8
```

		Genoma X_2		
		L/L_a	**L/S_b**	**S/S_c**
GRUPPO X_1	**Controlli$_a$**	6.29	6.40	6.00
	EP$_b$	8.50	6.64	4.62

Possiamo ricavare le medie dei sei gruppi con `tapply`, inserendo nella funzione la **lista** dei predittori: **`list(x1, x2)`**, invece di un solo predittore:

```
interazione<-tapply(ep$desiderio_sex, list(ep$gruppo, ep$genoma), mean)
interazione
```

	interazione[1] L/L	*interazione[3]* L/S	*interazione[5]* S/S
controllo	6.285714	6.400000	6.000
EP	8.500000	6.642857	4.625
	interazione[2]	*interazione[4]*	*interazione[6]*

L'effetto principale del predittore $X_{1\ Gruppo}$, indipendentemente dal genoma, è dato dalla differenza tra la media delle medie in X_2 dei controlli - X_{1a} e la media delle medie in X_2 dei pazienti - X_{1b}: $H_{0_{X1}} = \mu_{X1_a} = \mu_{X1_b}$

```
media_x1a<-(interazione[1]+interazione[3]+interazione[5])/3
media_x1b<-(interazione[2]+interazione[4]+interazione[6])/3
media_x1a; media_x1b
[1] 6.228571
[1] 6.589286
```

		Genoma X_2		
		L/L_a	L/S_b	S/S_c
GRUPPO X_1	Controlli$_a$	6.29 +	6.40 +	6.00 / 3
	EP$_b$	8.50 +	6.64 +	4.62 / 3

Il desiderio di controlli e pazienti, indipendentemente dal loro genoma, sembra sovrapporsi. Avete notato che le medie dei due gruppi calcolate così e calcolate con `tapply` non coincidono? Daremo la soluzione del mistero tra un po'.

L'effetto principale del predittore $X_{2\ Genoma}$, indipendentemente dal gruppo, è dato dalla differenza tra la media delle medie in X_1 dei soggetti LL - X_{2a}, la media delle medie in X_1 dei soggetti L/S - X_{2b} e la media delle media in X_1 dei soggetti S/S - X_{2c}: $H_{0_{X2}} = \mu_{X2_a} = \mu_{X2_b} = \mu_{X2_{bc}}$

```
media_x2a<-(interazione[1]+interazione[2])/2
media_x2b<-(interazione[3]+interazione[4])/2
media_x2c<-(interazione[5]+interazione[6])/2
media_x2a;media_x2b;media_x2c
[1] 7.392857
[1] 6.521429
[1] 5.3125
```

		Genoma X_2		
		L/L_a	L/S_b	S/S_c
GRUPPO X_1	Controlli$_a$	6.29 +	6.40 +	6.00 +
	EP$_b$	8.50	6.64	4.62
		/ 2	/ 2	/ 2

Il desiderio dei soggetti con genoma L/L, indipendentemente dall'essere controllo o paziente, sembra maggiore di quello dei soggetti con L/S (a stento) e S/S (più chiaramente); queste due tipologie non sono chiaramente differenziate,

probabilmente soprattutto a causa della grande variabilità in questo gruppo. Avete notato che le medie dei diversi genomi calcolate così e calcolate con `tapply` non coincidono?

Il motivo della mancata coincidenza **in un disegno sbilanciato** è dato dal fatto che sono calcolate diversamente **(`tapply` calcola le medie pesate, qui abbiamo calcolato medie non pesate da medie marginali).** Riprenderemo le medie pesate e non pesate nella partizione della devianza di Y.

L'effetto d'interazione $X_{1\ Gruppo} * X_{2\ Genoma}$ è dato dalla **differenza** nelle medie di controlli $X_{1\,a}$ e pazienti X_{1b} con genoma L/L-$X_{2\,a}$ **rispetto alla differenza** nelle medie di controlli $X_{1\,a}$ e pazienti $X_{1\,b}$ con genoma L/S-$\boldsymbol{X_{2b}}$ e **rispetto alla differenza** nelle medie di controlli $X_{1\,a}$ e pazienti $X_{1\,b}$ con genoma S/**S-**$\boldsymbol{X_{2c}}$.

```
diff_x1ab_x2a<-interazione[1]-interazione[2]
diff_x1ab_x2b<-interazione[3]-interazione[4]
diff_x1ab_x2c<-interazione[5]-interazione[6]
diff_x1ab_x2a;diff_x1ab_x2b;diff_x1ab_x2c
[1] -2.214286
[1] -0.2428571
[1] 1.375
```

		Genoma X_2		
		L/L_a	L/S_b	S/S_c
GRUPO X_1	Controlli$_a$	6.29	6.40	6.00
	EP$_b$	8.50	6.64	4.62

L'ipotesi nulla d'interazione è che le **tre differenze fra le medie non siano significativamente differenti**: avere un genoma L/L o L/S o S/S non incide sulla differenza tra pazienti e controllo, cioè la differenza tra i livelli nella variabile X_1 si mantiene costante indipendentemente dal livello di appartenenza del soggetto nella variabile X_2:

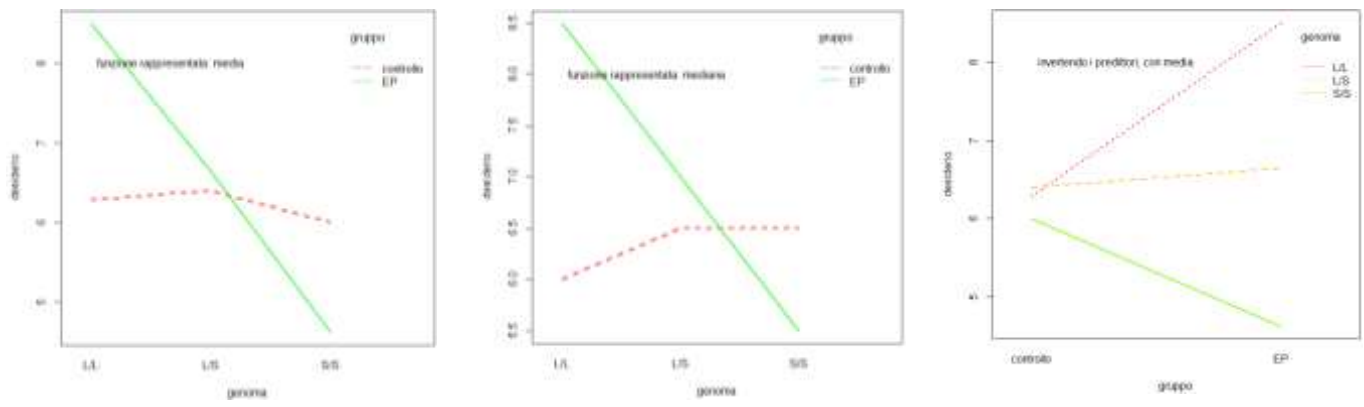
$$H_{0X2}: \mu_{X_{1_a}X_{2_a}} - \mu_{X_{1_b}X_{2_a}} = \mu_{X_{1_a}X_{2_b}} - \mu_{X_{1_b}X_{2_b}} = \mu_{X_{1_a}X_{2_c}} - \mu_{X_{1_b}X_{2_c}}$$

Naturalmente, l'interazione si può esprimere anche invertendo X_1 e X_2, mantenendo lo stesso significato: appartenere al gruppo di controllo o al gruppo dei pazienti non incide sulla differenza tra L/L ed L/S, tra L/L e S/S, tra L/S e S/S, cioè la differenza tra i livelli della variabile X_2 si mantiene costante indipendentemente del livello di appartenenza del soggetto nella variabile X_1:

$$H_{0X2}: \mu_{X_{1_a}X_{2_a}} - \mu_{X_{1_a}X_{2_b}} = \mu_{X_{1_a}X_{2_a}} - \mu_{X_{1_a}X_{2_c}} = \mu_{X_{1_a}X_{2_b}} - \mu_{X_{1_a}X_{2_c}} = \mu_{X_{1_b}X_{2_a}} - \mu_{X_{1_b}X_{2_b}} = \mu_{X_{1_b}X_{2_a}} - \mu_{X_{1_b}X_{2_c}} = \mu_{X_{1_b}X_{2_b}} - \mu_{X_{1_b}X_{2_c}}$$

13.1.1 La rappresentazione grafica dell'effetto di interazione

L'effetto di interazione può essere rappresentato graficamente in vari modi. I grafici d'interazione "tradizionali" sono grafici lineari in cui le categorie di un predittore sono in ascissa e le categorie dell'altro costituiscono le tracce entro il grafico. La funzione – base per creare un grafico di questo tipo in R è `interaction.plot(x.factor= predittore X1, trace.factor= predittore X2, response= Y)`; è utile aggiungere anche una legenda per le tracce, con `legend= TRUE`. Di default sono rappresentate le medie dei gruppi: è possibile cambiarle in mediane con l'argomento `fun = "median"`. Vediamo come è rappresentata l'interazione $X_1 \times X_2$ dell'esempio:



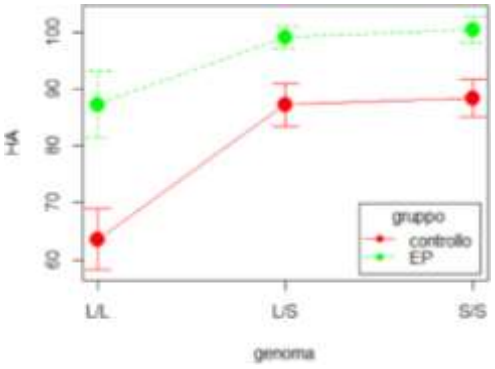
```
interaction.plot(x.factor =
ep$genoma, trace.factor =
ep$gruppo, response =
ep$desiderio_sex, legend = TRUE,
xlab = "genoma", ylab =
"desiderio", trace.label =
"gruppo", col=rainbow(3), lwd=2)
```

```
interaction.plot(x.factor =
ep$genoma, trace.factor =
ep$gruppo, response =
ep$desiderio_sex, legend = TRUE,
xlab = "genoma", ylab =
"desiderio", trace.label =
"gruppo", col=rainbow(3), lwd=2,
fun="median")
```

```
interaction.plot(x.factor =
ep$gruppo, trace.factor =
ep$genoma, response =
ep$desiderio_sex, legend = TRUE,
xlab = "gruppo", ylab =
"desiderio", trace.label =
"genoma", col=rainbow(8), lwd=2)
```

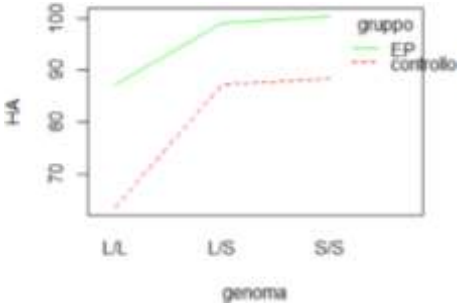
La differenza nel desiderio tra controlli e pazienti è ampia, a favore dei pazienti, se il genoma è L/L; si annulla completamente quando il genoma è L/S e torna ad aumentare, ma invertendo la direzione, quando il genoma è S/S: in questo caso, il desiderio dei pazienti è inferiore a quello dei controlli. Sembrerebbe, quindi, di poter leggere l'esistenza di una interazione tra genoma e gruppo: a seconda del genoma del soggetto, l'effetto del gruppo è differente. Un modo analogo per dire la stessa cosa è: il desiderio dei controlli resta praticamente costante nei tre diversi tipi di genoma, mentre quello dei pazienti cambia drasticamente al variare del genoma. Nel secondo grafico, la rappresentazione delle mediane conferma questa interpretazione, aumentando ulteriormente il dislivello tra i gruppi nella categoria L/L di X_2. Infine, nel terzo grafico vediamo l'interazione espressa invertendo i predittori nel grafico: di nuovo, vediamo che i controlli e i pazienti con genoma L/S mostrano ugual desiderio, mentre il desiderio dei pazienti con genoma L/L è molto più alto dei controlli con uguale genoma e il desiderio dei pazienti con genoma S/S è più basso di quello dei controlli S/S: sembra che l'effetto del genoma sul desiderio sia diverso a seconda del gruppo di appartenenza del soggetto.
Insomma, da qualsiasi parte si guardi il grafico, le linee che rappresentano le categorie di un predittore **non** sono **parallele** nel loro andare da una categoria all'altra del secondo predittore: **l'assenza di parallelismo** suggerisce la **presenza di interazione**, mentre linee parallele ne suggeriscono l'assenza.

Vediamo un esempio di assenza di interazione con la dimensione di Harm Avoidance, che riprenderemo successivamente. I pazienti hanno Harm Avoidance maggiore in tutti i tipi di genoma (effetto principale del Gruppo) e i soggetti con genoma L/L, sia i pazienti sia i controlli, hanno i punteggi di HA più bassi, mentre la differenza tra L/S e S/S è scarsa e di uguale entità nei due livelli del gruppo (effetto principale del Genoma). Passando da un genoma all'altro, la variazione dei pazienti e dei controlli è analoga (effetto di interazione – probabilmente non significativo).



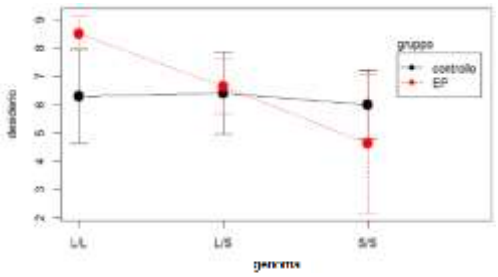
Questo tipo di grafico, anche se piuttosto intuitivo e molto utilizzato in articoli e testi, non è certo esente da difetti. Spesso la valutazione del parallelismo è fuorviante rispetto alla significatività dell'effetto d'interazione secondo l'ANOVA; deve sempre comprendere la rappresentazione dell'errore (errore standard, deviazione standard, CI...), pena diversi errori interpretativi; manca la rappresentazione dei punteggi individuali, che spesso dà informazioni sulla qualità del modello lineare (presenza di outlier, eteroschedasticità...).

Abbiamo visto `interaction.plot`, che dà un grafico bruttino, in cui non è possibile inserire facilmente informazioni sull'errore:

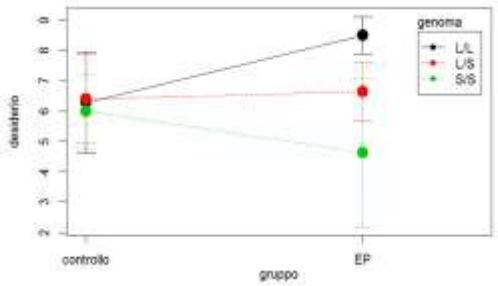


L'utilizzo di package grafici più sofisticati rispetto a quello di base può aiutare: un primo accorgimento essenziale è usare `plotMeans` di `RcmdrMisc` per aggiungere le barre di errore. Alla nota funzione aggiungiamo l'argomento `factor2=`

"X2" per rappresentare l'interazione; legend.lab= "legenda" e error.bars= "conf.int" completano l'informazione richiesta:




```
plotMeans(ep$desiderio_sex, factor1=ep$genoma,
factor2= ep$gruppo, error.bars = "conf.int",
connect= TRUE, ylab= "desiderio", legend.lab=
"gruppo", pch=19, xlab= "genoma")
```




```
plotMeans(ep$desiderio_sex, factor1=
ep$gruppo, factor2= ep$genoma, error.bars=
"conf.int", connect=TRUE, ylab= "desiderio",
legend.lab= "gruppo", pch=19, xlab="gruppo")
```

L'aggiunta delle barre di errore attorno alle medie dei gruppi dà informazioni in più, sia rispetto alla variabilità intragruppo sia rispetto all'appartenenza dei gruppi a diverse popolazioni (genoma L/L) o una medesima popolazione (L/S e – probabilmente – S/S).

Un'altra frequente rappresentazione grafica utilizza i barplot delle condizioni sperimentali. In barplot non è possibile inserire le barre d'errore, ma è sufficiente calcolare i valori che devono rappresentare (deviazione standard, SE, CI...) e aggiungerle con segments e arrows al plot. Vediamone tre esempi: per rappresentare sd, SE o CI dobbiamo comunque calcolare la deviazione standard di ogni gruppo, quindi creiamo l'oggetto **sd**:

```
sd<-tapply(ep$desiderio_sex,list(ep$gruppo, ep$genoma),sd)
sd
                L/L      L/S      S/S
controllo 1.7994708 2.011080 1.906925
EP        0.8498366 1.691933 2.924649
```

Per rappresentare lo SE della media, basta ricordare la sua formula: $SE = sd/\sqrt{N}$ e calcolare:

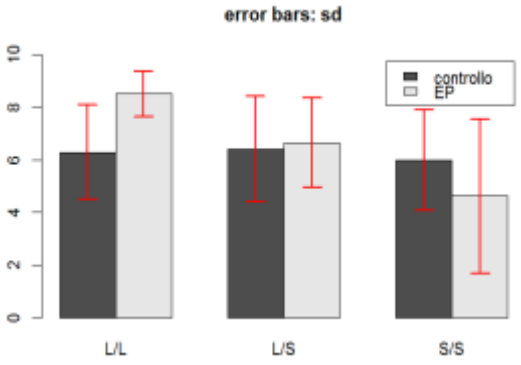
```
se<-tapply(ep$desiderio_sex, list(ep$gruppo, ep$genoma), sd) / sqrt(table(ep$gruppo, ep$genoma))
```

Oppure, più semplicemente, usare Mean.Se di DescTools:

```
se<-tapply(ep$desiderio_sex, list(ep$gruppo, ep$genoma), Mean.Se)
```

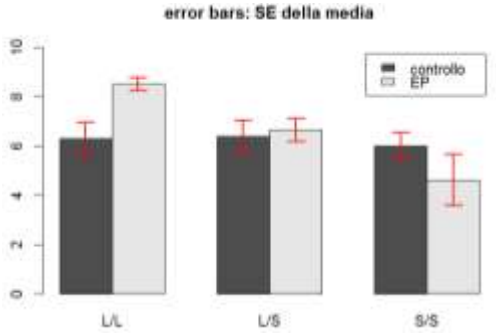
Abbiamo già creato interazioni, che contiene le medie delle sei condizioni. Facciamo il barplot: con beside= TRUE rappresentiamo le barre affiancate e con legend.text=TRUE aggiungiamo la legenda. Per rappresentare la sd nelle barre d'errore, usiamo prima segments(x0, y0, x1, x1) per tracciare la barra verticale con altezza dalla media$\pm sd$, poi arrows(x0, y0, x1, x1, code= 3, angle=90, lenght=) per tracciare i "baffi" orizzontali delle barre; code= 3 aggiunge il "baffo" ad entrambi i lati della barra verticale, lenght= ne indica la dimensione, espressa in pollici (= .1 o = .05 dovrebbero andare bene in tutti i casi).

```
grafico <- barplot(interazione, ylim=c(0,10),
beside=TRUE, legend.text=TRUE, main="error bars: sd")

segments(x0=grafico, y0=interazione-sd, x1=grafico,
y1=interazione+sd, col="red", lwd=2)
arrows(x0=grafico, y0=interazione-sd, x1=grafico,
y1=interazione+sd, lwd=2, angle=90,code= 3, col="red",
length =.1)
```



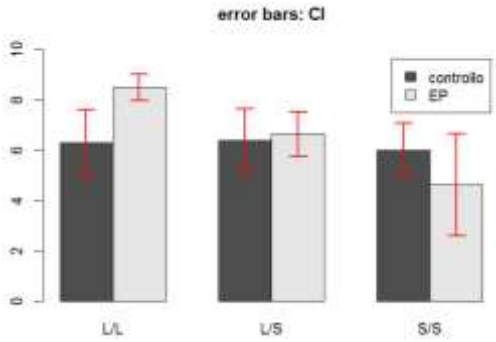
Sostituiamo sd con SE:

```
grafico <- barplot(interazione, ylim=c(0,10), beside=TRUE,
legend.text=TRUE, main="error bars: sd")

segments(x0=grafico, y0=interazione-se, x1=grafico,
y1=interazione+se, col="red", lwd=2)
arrows(x0=grafico, y0=interazione-se, x1=grafico,
y1=interazione+se,l wd=2, angle=90,code= 3, col="red",
length =.1)
```



Per rappresentare il CI attorno alla media, ricordiamoci che l'UL è dato dalla $media + SE$ moltiplicato per il quantile corrispondente ad $\alpha/2$, e il LL dalla $media - SE$ moltiplicato per quantile corrispondente a $\alpha/2$. Se si usa la distribuzione normale standardizzata, avremo allora:

```
grafico<-barplot(interazione, ylim=c(0,10),beside =
   TRUE,legend.text = TRUE, main="error bars: CI")

segments(x0=grafico, y0=interazione -se*1.96, x1=grafico,
   y1=interazione+se*1.96,  lwd=2, col="red")
arrows(x0=grafico, y0=interazione-se*1.96, x1=grafico,
   y1=interazione+se*1.96, lwd=2, angle=90,col="red", code=
   3, length = .1)
```

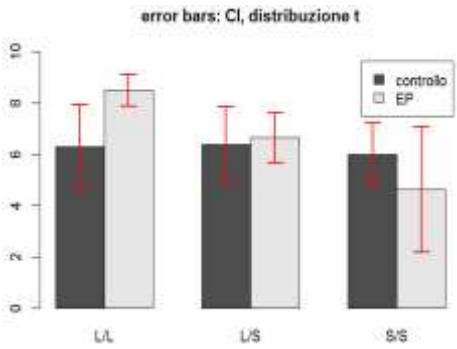


Naturalmente, se si usa la distribuzione di probabilità t si dovrà moltiplicare lo SE per il quantile t corrispondente ad $\alpha/2$ in una distribuzione con $\boldsymbol{df = N - 1}$. Calcoliamoli, rispolverando la funzione di ripartizione:

```
df<-table(ep$gruppo,ep$genoma)-1
quantile_t<-qt(p = .025, df = df,lower.tail=FALSE)
```

Inseriamo il quantile t nel calcolo del CI:

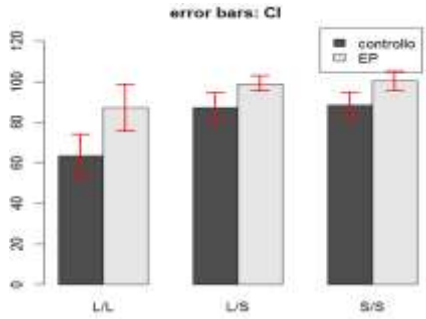
```
grafico<-barplot(interazione, ylim=c(0,10),beside =
   TRUE,legend.text = TRUE, main="error bars: CI")

segments(x0=grafico, y0=interazione -se*quantile_t,
   x1=grafico, y1=interazione+se* quantile_t,  lwd=2,
   col="red")
arrows(x0=grafico, y0=interazione-se*quantile_t, x1=grafico,
   y1=interazione+se*quantile_t, lwd=2, angle=90,col="red",
   code= 3, length = .1)
```



Se l'effetto principale di $X_{1Gruppo}$ fosse significativo, le barre dei Controlli dovrebbero essere più alte delle barre degli EP (o viceversa) in tutti i livelli in X_1X_2: invece, sembra che la media delle medie di Controlli e EP sia piuttosto simile. Se l'effetto principale di $X_{2Genoma}$ fosse significativo, le barre nel livello L/L, indipendentemente dal gruppo che rappresentano, dovrebbero essere ampiamente diverse dall'elevazione media in L/S e/o in S/S, e così le medie in L/S rispetto a S/S: in effetti, così è tra L/L e S/S, ma sembra più problematica la differenza tra L/L e L/S, come tra L/S e S/S. Infine, se fosse significativa l'interazione, la differenza tra le barre dei gruppi Controlli ed EP in ogni livello del genoma dovrebbe manifestarsi in maniera diversa: in effetti, in L/L la differenza è abbastanza rilevante e a favore dei controlli, in L/S i gruppi sono sostanzialmente identici, in S/S la differenza è minore (e con un ampio SE) e a favore dei pazienti.

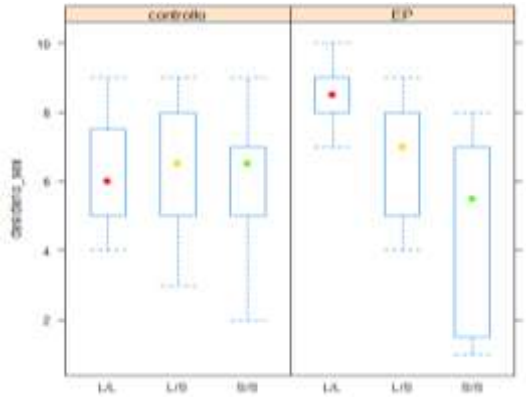
Rispetto alla dimensione di Harm Avoidance, che il grafico precedente suggeriva non risentire di un effetto di interazione, il barplot dice:

```
HA <- tapply(ep$HA, list(ep$gruppo, ep$genoma), mean)
se_HA<-tapply(ep$HA,list(ep$gruppo, ep$genoma), sd) /
  sqrt(table(ep$gruppo, ep$genoma))
grafico<-barplot(HA,ylim=c(0,130), beside = TRUE, legend.text =
  TRUE, main="error bars: CI")
segments(x0 = grafico, y0 = HA -se_HA*1.96, x1 = grafico, y1 =
  HA+se_HA*1.96, lwd=2, col="red")
arrows(grafico, HA-se_HA*1.96, grafico, HA+se_HA*1.96, lwd=2,
  angle=90, col="red", code= 3, length = .1)
```

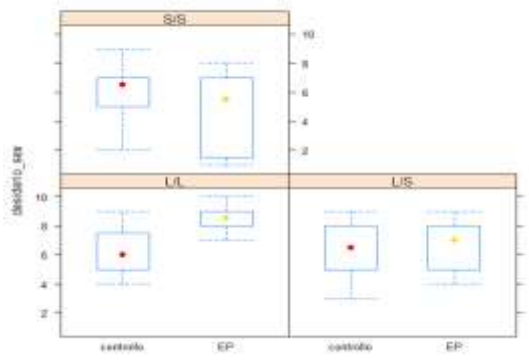


I pazienti EP hanno punteggi maggiori in tutti i livelli di X_2: l'effetto principale di $X_{1Gruppo}$ potrebbe essere significativo. Entrambi i gruppi nel livello L/L di X_2 hanno punteggi più bassi dei gruppi nei livelli L/S e S/S, ma solo la differenza degli EP sembra significativa, e L/S ed S/S, indipendentemente dal gruppo, sono uguali: l'effetto principale di $X_{2Genoma}$ è dubbio. La differenza tra i gruppi ha la stessa direzione in tutti i livelli di X_2 e l'intensità della differenza è uguale in L/S e S/S: la significatività dell'effetto di interazione è molto improbabile.

Infine, per rappresentare **mediane** invece di medie, si può usare il **boxplot condizionale** con **`bwplot(Y~X1|X2)`** di `lattice`, che restituisce **boxplot** condizionali in un layout di interazione:



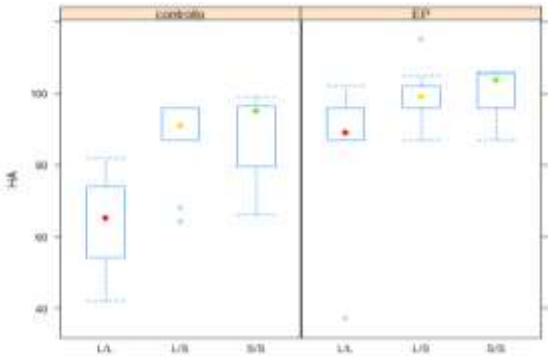

```
bwplot(desiderio_sex ~ genoma | gruppo, data=ep,
col=rainbow(7))
```




```
bwplot(desiderio_sex ~ gruppo | genoma, data=ep,
col=rainbow(7))
```

Il punto centrale di ogni box è, intuitivamente, la **mediana**; oltre all'elevazione delle mediane nei gruppi, il boxplot offre la preziosa informazione sulla dispersione intragruppo e gli outliers, assente nella versione base dell'interaction plot.

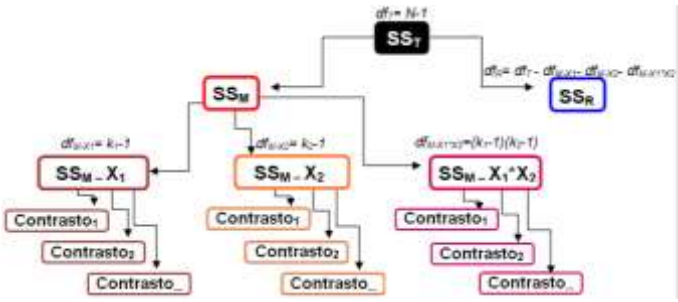
Rispetto alla dimensione di Harm Avoidance, che il grafico precedente suggeriva non risentire di un effetto di interazione, il boxplot condizionale dice:



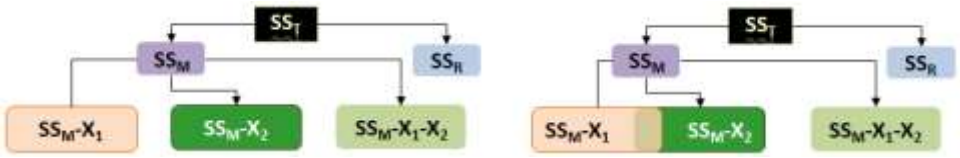
13.1.2 La partizione della devianza del modello

Una volta descritti gli effetti principali e d'interazione, potremmo passare alla partizione della devianza di Y tra predittori ed errore, cioè al modello lineare**:** $Y_{ij} = (b_0 + b_1X_1 + b_1X_2 + b_1X_1 * X_2) + e_{ij}$:

Le tre devianze dei predittori e la devianza d'errore, divise per i propri df, originano le varianze: $MS_{M_{X1}}$, $MS_{M_{X2}}$, $MS_{M_{X1\times X2}}$ e MS_R. Le tre varianze del modello, divise per la varianza d'errore, creano tre rapporti F_{X1}, F_{X2}, $F_{X1\times X2}$: ciascuno rappresenta il rapporto tra quantità di varianza spiegata dal predittore (o dall'interazione) e dalla varianza residua, e per ciascuno avremo un $p - value$. Naturalmente, **ogni SS_M** viene ulteriormente **scomposta nei propri contrasti**, come visto nel capitolo precedente.



Tutto sembrerebbe scorrere liscio, ma a questo punto insorge il vero problema dei disegni fattoriali, cioè **come ripartire la SS_M tra tutti i predittori**, effetti principali e interazione. In realtà, questo problema si pone **solo** quando i disegni **non sono bilanciati**, perché se i disegni sono **bilanciati** il problema della **sovrapposizione / confusione** tra gli effetti **non** c'è.



Come anticipato qualche capoverso più su, l'effetto principale di ogni fattore è indicato dalla differenza tra le **medie** delle condizioni, che è possibile calcolare in due modi:

1) la **media pesata (weighted, m_w) di un livello** prende in considerazione la **frequenza delle osservazioni in ogni cella** della tabella fattoriale: è la "solita" media, calcolata come d'abitudine. Facciamo il semplice esempio di una tabella 2×2, in cui abbiamo le **risposte corrette** a un quiz di statistica di 16 studenti, divisi a seconda **di A – Frequenza in aula** (a_1: sì; a_2: no) e **B– Fare esercizi a casa** (b_1:sì; b_2: no):

```
a1b1<-c(20,25,30,35); a2b1<-c(15,10,8,7); a1b2<-c(20,15,20,10); a2b2<-c(6,6,5,10)
```

		B esercizi		
		b_1	b_2	

A aula	a_1	20
		25
		30
		35
			20
			15
			20
			10
				$m_{wa1} = \frac{f_{a1b1}\bar{x}_{a1b1} + f_{a1b2}\bar{x}_{a1b2}}{N_{b1}}$
				`(media_a1<-(4*mean(a1b1)+(4*mean(a1b2)))/8)`
				`[1] 21.875`
				`(20+25+30+35+20+15+20+10)/8`
				`[1] 21.875`

	a_2	15
		10
		8
		7
			6
			6
			5
			10
				$m_{wa2} = \frac{f_{a2b1}\bar{x}_{a2b1} + f_{a2b2}\bar{x}_{a2b2}}{N_{b1}}$
				`(media_a2<-(4*mean(a2b1) + (4*mean(a2b2)))/8)`
				`[1] 8.375`
				`(15+10+8+7+6+6+5+10)/8`
				`[1] 8.375`

		$m_{wb1} = \frac{f_{a1b1}\bar{x}_{a1b1} + f_{a2b1}\bar{x}_{a2b1}}{N_{h1}}$
		`(media_b1<-((4*mean(a1b1)) + (4*mean(a2b1)))/8)`
		`[1] 18.75`
		`(20+25+30+35+15+10+8+7)/8`
		`[1] 18.75`
			$m_{wb2} = \frac{f_{a1b2}\bar{x}_{a1b2} + f_{a2b2}\bar{x}_{a2b2}}{N_{h1}}$
			`(media_b2<-((4*mean(a1b2)) + (4*mean(a2b2)))/8)`
			`[1] 11.5`
			`(20+15+20+10+6+6+5+10)/8`
			`[1] 11.5`	

Cercheremo l'effetto principale di A – essere presente in aula nella **differenza tra 21.875 e 8.375**, e l'effetto principale di B - fare esercizi nella **differenza tra 18.75 e 11.5**.

2) la **media non pesata (unweighted, m_{uw})** di un livello **ignora l'informazione** sulla frequenza delle osservazioni in ogni cella della tabella fattoriale: è data dalla **media delle medie delle condizioni di quel livello**, ovvero dalla **media delle medie marginali**. Per gli stessi soggetti dell'esempio precedente, avremmo:

		B esercizi		
		b_1	b_1	

A aula	a_1	20
		25
		30
		35
			20
			15
			20
			10
				$m_{uwa1} = \frac{\bar{x}_{a1b1} + \bar{x}_{a1b2}}{2}$
				`(mean(a1b1)+mean(a1b2))/2`
				`[1] 21.875`

		`mean(a1b1)` `[1] 27.5`	`mean(a1b2)` `[1] 16.25`	

	a_2	15
		10
		8
		7
			6
			6
			5
			10
				$m_{wa2} = \frac{\bar{x}_{a2b1} + \bar{x}_{a2b2}}{2}$
				`(mean(a2b1)+mean(a2b2))/2`
				`[1] 8.375`

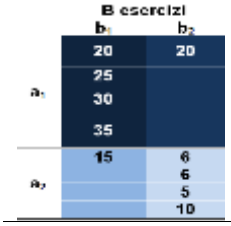
		`mean(a2b1)` `[1] 10`	`mean(a2b2)` `[1] 6.75`	

		$m_{wb1} = \frac{\bar{x}_{a1b1} + \bar{x}_{a2b1}}{2}$
		`(mean(a1b1)+mean(a2b1))/2`
		`[1] 18.75`
			$m_{wb2} = \frac{\bar{x}_{a1b2} + \bar{x}_{a2b2}}{2}$
			`(mean(a1b2)+mean(a2b2))/2`
			`[1] 11.5`	

Nuovamente, cercheremo l'effetto principale di A nella differenza tra 21.875 e 8.375, e l'effetto principale di B nella differenza tra 18.75 e 11.5. è È evidente che quando il disegno è bilanciato, cioè quando lo stesso numero di osservazioni si ripete per ogni cella, come nell'esempio, le **medie pesate e non pesate sono uguali:** ciascuna condizione "pesa" nello stesso modo nel determinare l'effetto del fattore A e del fattore B.

Quindi, quando il disegno è bilanciato usare per l'analisi le medie pesate o non pesate non crea differenza, dato che il loro risultato è uguale. Quando, però, le condizioni sperimentali hanno diversa numerosità, le medie pesate e non pesate tendono a divergere, tanto più fortemente quanto più le N dei gruppi sono differenti, **portando confusione nell'interpretazione dell'effetto dei predittori**. Usiamo un caso limite di disegno sbilanciato, per meglio evidenziare il

fenomeno: un solo studente frequentante che non fa esercizi ($N_{a_1b_2} = 1$) e un solo studente non frequentante che fa esercizi ($N_{a_2b_1} = 1$).



A	a_1	Pesata	`print(a1_pesata<-((4*mean(a1b1))+(1*(a1b2)))/5)`	`[1] 26`
		Non pesata	`print(a1_nonpesata<-(mean(a1b1)+mean(a1b2))/2)`	`[1] 23.75`
	a_2	Pesata	`print(a2_pesata<-((1*mean(a2b1))+(4*mean(a2b2)))/5)`	`[1] 8.4`
		Non pesata	`print(a2_nonpesata <-(mean(a2b1) + mean(a2b2))/ 2)`	`[1] 10.875`
B	b_1	Pesata	`print(b1_pesata<-((4*mean(a1b1))+(1*(a2b1)))/5)`	`[1] 25`
		Non pesata	`print(2_pesata<-((1*mean(a1b2))+(4*mean(a2b2)))/5)`	`[1] 21.25`
	b_2	Pesata	`print(b1_nonpesata<-(mean(a1b1)+mean(a2b1))/2)`	`[1] 9.4`
		Non pesata	`print(b2_nonpesata<-(mean(a1b2)+mean(a2b2))/2)`	`[1] 13.375`

Ora le medie pesate e non pesate divergono parecchio, e non solo: la **differenza che definisce l'effetto nel caso delle medie pesate è fuorviante**. Si direbbe che l'essere stati presenti in aula (a_1) abbia determinato un punteggio decisamente maggiore rispetto alla non presenza (A_2: $26\ versus\ 8.4$), ma il **dato sulla frequenza è confuso dal fatto che l'80% dei casi ha anche fatto esercizi**, e solo uno no. Analogamente, sembra che l'aver fatto esercizi (b_1) abbia determinato un punteggio decisamente maggiore rispetto al non farli (b_2**;** $25\ versus\ 9.4$), ma il **dato sull'esercizio è confuso dal fatto che l'80% dei casi non era frequentante**. L'effetto di A – Frequenza e l'effetto di B – Esercizio sono dunque **confusi, mescolati, intersecati → i loro effetti non possono essere (completamente) separati, perché variano insieme → i fattori non sono ortogonali** e **non sono indipendenti**. Abbiamo già visto questo problema, in modelli con predittori continui, nel Capitolo 5, parlando del requisito di assenza di **multicollinearità**.

Invece, la **differenza tra le medie non pesate non è non è gravata da questa confusione**: **nei disegni non bilanciati, le medie non pesate sono quindi una misura migliore dell'effetto principale**, dato che controllano l'effetto delle altre variabili e dunque abbattono la fusione dei loro effetti.

Applichiamo quanto detto al nostro dataframe `ep`. Sappiamo che le condizioni sperimentali non sono bilanciate (salviamo anche la numerosità dei livelli in `enne`), quindi **troveremo che m_w e m_{uw} non coincidono**:

```
enne<- table(ep$gruppo, ep$genoma)
           L/L  L/S  S/S
controllo    7   10   12
EP          10   14    8
```

Avevamo salvato le sei medie di interazione nell'oggetto `interazione`:

```
interazione
                L/L      L/S   S/S
controllo 6.285714 6.400000 6.000
EP        8.500000 6.642857 4.625
```

Cominciamo calcolando le **medie pesate** del fattore **Gruppo**, e confrontiamole con le medie che avevamo ottenuto con `tapply`:

```
((enne[1] * interazione[1]) + (enne[3] * interazione[3]) + (enne[5]
  * interazione[5]))/29
[1] 6.206897
((enne[2] * interazione[2]) + (enne[4] * interazione[4]) + (enne[6]
  * interazione[6]))/32
[1] 6.71875
```

```
tapply(ep$desiderio_sex,
ep$gruppo, mean)
 controllo       EP
 6.206897 6.71875
```

Ora vediamo le **medie non pesate** per il **Gruppo** usando le medie marginali, cioè facendo la media delle medie delle condizioni per i controlli e per i pazienti, e confrontiamole con le medie che avevamo usato per esprimere **l'effetto principale del Gruppo**:

```
(interazione[1]+interazione[3] +
  interazione[5])/3
 [1] 6.228571
(interazione[2]+interazione[4] +
 interazione[6])/3
 [1] 6.589286
```

```
(media_x1a<-(interazione[1] +
 interazione[3] + interazione[5])/3)
 [1] 6.228571
(media_x1b<-(interazione[2] +
 interazione[4] + interazione[6])/3)
 [1] 6.589286
```

Naturalmente, scopriremo lo stesso per il predittore **Genoma**: prima le medie pesate:

```
((enne[1]*interazione[1])+(enne[2]*interazione[2]))/17
[1] 7.588235
((enne[3]*interazione[3])+(enne[4]*interazione[4]))/24
[1] 6.541667
((enne[5]*interazione[5])+(enne[6]*interazione[6]))/20
[1] 5.45
```

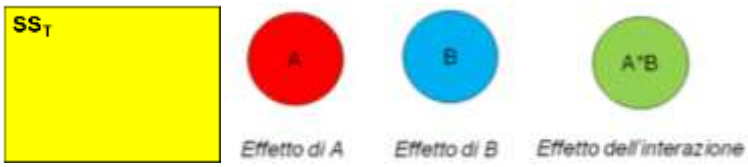
```
tapply(ep$desiderio_sex, ep$genoma, mean)
L/L
  7.588235
L/S
  6.541667
S/S
  5.45
```

… poi quelle non pesate:

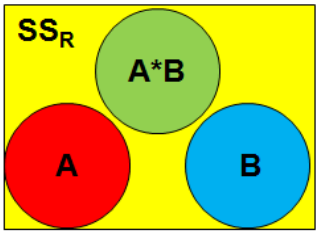
```
(interazione[1]+interazione[2])/2
[1] 7.392857
(interazione[3]+interazione[4])/2
[1] 6.521429
(interazione[5]+interazione[6])/2
[1] 5.3125
```

```
(media_x2a<-(interazione[1]+interazione[2])/2)
[1] 7.392857
(media_x2b<-(interazione[3]+interazione[4])/2)
[1] 6.521429
(media_x2c<-(interazione[5]+interazione[6])/2)
[1] 5.3125
```

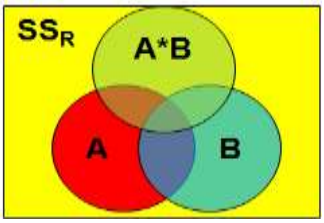
Perciò, quando gli effetti dei predittori si confondono l'uno nell'altro (i **predittori covariano**), è responsabilità di chi conduce l'analisi di tratta di attribuire questa porzione della variabilità di Y confusa tra le X all'uno o all'altro predittore, oppure di escluderla dalla quota di devianza di Y che viene ripartita tra modello ed errore - quindi sottrarla dall'ANOVA. La scelta segue **tre diverse logiche di partizione della devianza di Y da attribuire ai fattori**, chiamate **devianza di tipo I, tipo II e tipo III** (Overall e Spiegel, 1969): la **prima** usa **medie pesate**, la **seconda e la terza medie non pesate**. Usiamo una rappresentazione grafica (approssimazione un po' rozza, ma generalmente efficace). Rappresentiamo come un rettangolo giallo la **variabilità di Y** (SS_T), cui sovrapponiamo tre cerchi che rappresentano la quota di variabilità di Y spiegata dall'effetto principale di A (SS_{M_A}), dall'effetto principale di B (SS_{M_B}) e dall'interazione $A \times B$ ($SS_{M_{A*B}}$); per economia grafica, rappresentiamo i tre cerchi con ugual diametro, assumendo che i tre effetti si equivalgano. Avremo:



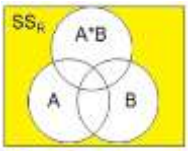
Quando il disegno è **bilanciato**, i tre fattori sono **ortogonali**, ovvero **non hanno varianza in comune** → le loro aree non si sovrappongono. La quota di varianza di Y che ciascuno spiega non è attribuita a nessun altro fattore (varianza unica). La porzione della variabilità di Y non spiegata da alcuno dei tra fattori (**area visibile in giallo**) rappresenta la devianza d'errore / residua SS_R.



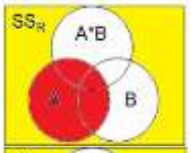
Quando il disegno **non** è **bilanciato**, i tre fattori **non** sono **ortogonali**, ovvero **hanno varianza in comune** → parte della variabilità di Y, corrispondente all'intersezione dei tre fattori non è univocamente assegnata a un solo effetto, poiché i fattori non sono indipendenti. Cosa fare di questa parte di variabilità di Y condivisa?
Le risposte sono (almeno) tre:



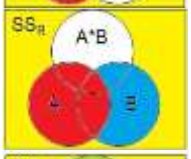
1) Partizione della devianza di tipo I (SS I) o sequential sum of squares



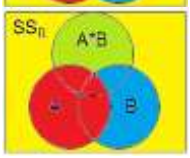
Partendo da questa situazione,

1. 

si assegna al **primo fattore inserito nel modello** (**effetto principale di A**) tutta la varianza unica spiegata da A e **tutta la varianza che A condivide con B e con $A \times B$**.

2. 

Poi si inserisce nel modello il secondo fattore (effetto principale di B), cui si assegna tutta la varianza unica di B residua (cioè quella non precedentemente assegnata ad A) e la variabilità che B condivide con $A \times B$.

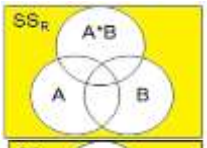
3. 

Infine, si inserisce nel modello il termine di interazione $A \times B$, cui è assegnata solo la varianza unica spiegata dall'interazione.

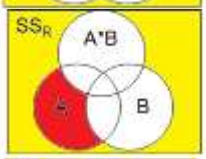
Formalizzando, quindi, si stabilisce prima la devianza attribuibile ad A (SS_{M_A}), poi la devianza attribuibile a B, esclusa quella assegnata ad A ($SS_{M_{B|A}}$) e infine quella di interazione, esclusa quella assegnata ai due effetti principali A e B ($SS_{M_{A\times B|A,B}}$). Come detto, questa partizione utilizza le medie pesate.

SS_M I tipo
1. $\boldsymbol{SS(A)}$
2. $\boldsymbol{SS(B|A)}$
3. $\boldsymbol{SS(A \times B|A,B)}$

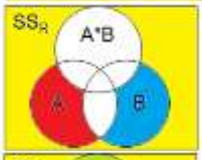
2) Partizione della devianza di tipo III (SS_M III) o marginal / orthogonal sum of squares



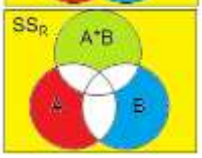
Partendo da questa situazione,

1. 

si assegna al **primo fattore inserito nel modello** (**effetto principale di A**) **solo la varianza** unica spiegata da A.

2. 

Poi si inserisce nel modello il secondo fattore (effetto principale di B), cui si assegna tutta la **varianza unica di B**.

3. 

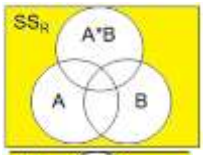
Infine, si inserisce nel modello il termine di interazione $A \times B$, cui è assegnata solo la varianza unica spiegata dall'interazione

Tutta la varianza condivisa tra i fattori viene quindi esclusa dall'analisi, dato che non è nemmeno assegnata alla varianza di errore. Naturalmente, questa soluzione drastica è particolarmente punitiva per l'emergere della significatività dei predittori quanto più questi sono sovrapposti: è definita orthogonal SS perché in questo modo gli effetti principali e di interazione sono **indipendenti** (non hanno varianza in comune), o marginal SS perché usa le **medie non pesate**, cioè la media delle medie marginali.

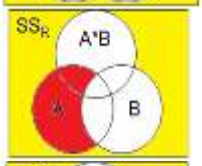
Formalizzando: si stabilisce prima la devianza unicamente attribuibile ad A (SS_{M_A}), poi la devianza unicamente attribuibile a B (SS_{M_B}) e infine quella unicamente attribuibile all'interazione ($SS_{M_{A\times B}}$):

SS_M III tipo
1. $\boldsymbol{SS(A|B,A\times B)}$
2. $\boldsymbol{SS(B|A,A\times B)}$
3. $\boldsymbol{SS(A\times B|A,B)}$

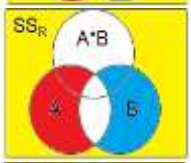
3) Partizione della devianza di tipo II (SS_M II) o partially sequential sum of squares



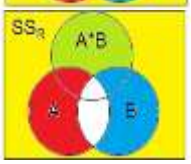
Partendo da questa situazione,

1. 

si assegna al **primo fattore inserito nel modello** (**effetto principale di A**) la varianza unica **di A e quella che A condivide con l'interazione**.

2. 

Poi si inserisce nel modello il secondo fattore (effetto principale di B), cui si assegna tutta la **varianza unica di B e quella che B condivide con l'interazione**.

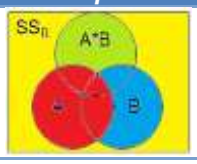
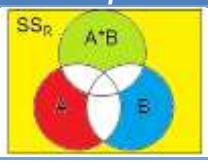
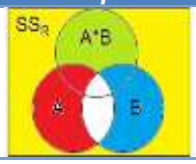
3. 

Infine, si inserisce nel modello il termine di interazione $A \times B$, cui è assegnata solo la varianza unica spiegata dall'interazione

Questo procedimento è un compromesso rispetto al precedente, dato che la varianza condivisa tra gli effetti principali (SS_{M_A}, SS_{M_B}) e l'interazione viene attribuita agli effetti principali, e resta esclusa dall'analisi solo la varianza condivisa tra gli effetti principali. All'interazione, di nuovo, si attribuisce solo la varianza unica di $A \times B$:

SS_M II tipo
1. $\boldsymbol{SS(A\|B)}$
2. $\boldsymbol{SS(B\|A)}$
3. $\boldsymbol{SS(A \times B\|A,B)}$

In sintesi, quindi:

Partizione di I tipo	*Partizione di III tipo*	*Partizione di II tipo*
		

In tutte le partizioni, la devianza residua $\boldsymbol{SS_R}$ è identica, così come la SS di interazione $\boldsymbol{SS_{M_{A*B}}}$. Nel tipo I, l'effetto principale di A, il primo inserito nel modello, si vede attribuita tutta la varianza di Y che riesce a spiegare, compresa quella non unica: perciò, **anche a parità di intensità rispetto a B**, come nel nostro esempio, in realtà gli viene attribuita una maggior varianza spiegata. Nelle altre due partizioni, a parità di intensità, ai due effetti principali è attribuita la stessa quota di varianza unica. La parte di variabilità attribuita agli effetti confusi dei fattori è totalmente attribuita al primo effetto nel tipo I, totalmente esclusa dalla devianza totale di Y (sia del modello sia di errore) nel tipo III, parzialmente attribuita ai due effetti principali, a scapito dell'interazione, nel tipo II.

Quindi, dato che la scelta della partizione della SS_M può avere ricadute pesanti sulla sorte delle ipotesi nulle, è il caso di ponderarla accuratamente.

Nella **partizione di tipo I** le SS dei predittori sono calcolate come in una regressione gerarchica: quella di B dopo che A è stato inserito e valutato, quella di $A \times B$ dopo che anche B stato inserito e valutato. **A meno che i predittori non siano completamente indipendenti, quindi, la SS_M di tipo I è da evitare quando il disegno non è bilanciato**, dato che il procedimento non presenta **invarianza**: l'ordine di inserimento dei fattori nel modello conta e il destino delle H_0 dipende da esso. Può capitare (ne vedremo degli esempi) che adattando due modelli ANOVA alla stessa Y, ma invertendo l'ordine dei predittori del modello, si ottengano SS_{M_A} e SS_{M_B} così diverse da determinare l'accettazione di H_0 in un caso e il suo rifiuto nell'altro. Non è nemmeno possibile predire con certezza se la SS dell'effetto aumenterà o

diminuirà, a seconda della sua posizione. È invece un approccio sensato per disegni bilanciati, con fattori ortogonali, o purché ci sia una **buona ragione teorica per determinare l'ordine dei fattori** (può essere il caso di una covariata, come vedremo nell'analisi della covarianza, capitolo 7).

Se i motivi per scegliere o evitare la partizione di tipo I sono piuttosto chiari e indiscussi, le raccomandazioni per una scelta tra partizione di tipo III e tipo II sono decisamente più sfumate e in letteratura soffrono di una qual certa partigianeria. Nella **partizione di tipo III** le SS_M sono calcolate prendendo in considerazione tutti gli altri effetti nel modello: la devianza attribuita a B è valutata parzializzando gli effetti di A e $A \times B$, la devianza attribuita ad A è stimata parzializzando gli effetti di B e $A \times B$, la devianza attribuita ad $A \times B$ è stimata dopo gli effetti principali A e B: quindi, l'effetto di ogni fattore è valutato dopo che tutti gli altri sono stati considerati. Dato che la devianza condivisa tra i fattori non viene utilizzata, nemmeno assegnandola alla SS_R, sommando tutte le devianze SS_M e la SS_R non si ottiene una somma uguale alla SS_T, come avviene nel caso precedente. In questa partizione sia gli effetti principali che le interazioni sono sempre chiaramente interpretabili: è **preferibile quando l'interesse è focalizzato sugli effetti principali** (*overall*; Spiegel e Cohen, 1975; Howell, 2006) ed è **equivalente alla partizione di tipo II se l'ipotesi è centrata sull'interazione**, anche se **esige contrasti ortogonali** (capitolo 6) e **non** funziona correttamente in presenza di **celle vuote**.

Nella partizione di tipo II, le SS_M sono calcolate prendendo in considerazione **tutti gli altri effetti** nel modello, **tranne gli effetti di ordine superiore che includono l'effetto da valutare**. B è valutato al netto di A, A al netto di B, $A \times B$ al netto degli effetti principali di A e B. Se si è interessati alle interazioni, SS_M di tipo II e III si equivalgono (Overall, Spiegel e Cohen, 1975; Howell, 2006). Questo approccio è appropriato per la costruzione di modelli, ed è quindi la scelta più "naturale" nel caso di regressioni multiple; è **più potente del tipo III se non c'è interazione significativa** e **non richiede contrasti ortogonali**. Tuttavia, per disegni fattoriali non bilanciati, la partizione di tipo II testa l'ipotesi che ci siano funzioni complesse delle numerosità di cella, che solitamente non hanno senso rispetto alle ipotesi sperimentali. Inoltre, ha come assunto (non sempre rispettabile) che le osservazioni mancanti siano state perse in maniera completamente casuale, per cui non ci sarebbe motivo per attribuire un peso maggiore a una cella con più osservazioni.

Ultimo dettaglio: esiste anche **la partizione di tipo IV** (SS_M **IV**; Goodnight, 1960), che procede come la partizione di tipo III e gestisce la presenza di celle vuote, ma: "*as Cochran and Cox suggested, "the only complete solution of the 'missing data' problem is not to have them*" (p. 82)".

In estrema sintesi, quindi:

Effetto	Disegno bilanciato	Disegno non bilanciato	Celle vuote
A	I=II=III=IV	III=IV	
B	I = II = III = IV	I=II;III=IV	I=II
$A \times B$	I = II = III = IV	I=II=III=IV	I=II=III=IV

Come **gestire in R** la scelta sulla partizione della SS_M? Ci sono almeno tre possibilità praticabili per fare un'analisi della varianza between groups:

1) `aov(Y~X*Y)` e `anova(Y~X*Y)` fanno parte delle statistiche di base e usano la partizione di **tipo I** come metodo di default, non modificabile dall'utente: meglio quindi usare queste due funzioni solo nel caso di un unico predittore (capitolo 6) o di disegni fattoriali bilanciati (o nell'analisi della covarianza, capitolo 7);
2) `ezANOVA` del package `ez`, che abbiamo usato nell'ANOVA a misure ripetute, gestisce anche ANOVA between groups e miste usando **la partizione di tipo II come metodo di default**, ma basta specificare `1`, `2` o `3`

nell'argomento `type=` per **cambiarla**. Inoltre, nel caso si scelga il metodo III, che richiede contrasti ortogonali, **non** è necessario specificarli: `ezANOVA` cambia da sé i contrasti dei fattori in modo che lo diventino;

3) `lm` usa la **partizione di tipo III come metodo di default**, e non è possibile cambiarla nella statistica di base. Tuttavia, si può usare `Anova` (occhio alla maiuscola) di `car`, che **lavora su oggetti di classe `lm` o `aov`** e con il cui argomento `type= 1-2-3` si può specificare l'approccio desiderato. Però, se scegliamo il tipo III dobbiamo anche indicare tra gli argomenti l'impostazione del **contrasto ortogonale** per ciascuno degli effetti principali, inserendo `contr.sum` o `contr.poly` o `contr.helmert`.

Applichiamo queste tre partizioni al modello $\boldsymbol{desiderio \sim gruppo \times genoma}$ e vediamo cosa comportano.
Il disegno è sbilanciato, quindi la partizione di tipo I non sarebbe raccomandabile, ma vediamola a fini didattici: usiamo `aov` inserendo **al primo posto** nel modello il **Gruppo**: è il suo effetto principale, quindi, a fare la parte del leone nella spartizione della SS_M:

```
summary(aov(ep$desiderio_sex~ep$gruppo*ep$genoma))
                    Df Sum Sq Mean Sq  F value  Pr(>F)
ep$gruppo            1   3.99   3.986    1.099 0.29901
ep$genoma            2  39.25  19.625    5.413 0.00715
ep$gruppo:ep$genoma  2  28.56  14.279    3.938 0.02521
Residuals           55 199.42   3.626
```

Il gruppo, indipendentemente dal genoma, non esercita un effetto significativo sul desiderio; il desiderio, indipendentemente dal gruppo di appartenenza, ha un effetto significativo sul desiderio – ma dovremo fare i test a coppie post hoc per conoscere le differenze a coppie; l'interazione è significativa: seconda del genoma del soggetto, la differenza tra controlli e pazienti cambia [oppure: a seconda del gruppo cui appartiene il soggetto, la differenza fra i tre genomi cambia]: il desiderio dei pazienti è maggiore nel caso dell'allele L/L, è sovrapponibile a quello dei controlli nel caso dell'allele L/S, è inferiore ai controlli nel caso dell'allele L/S [oppure: mentre il desiderio dei controlli sembra indipendente dal genoma, il desiderio dei pazienti cambia drasticamente a seconda del genoma del soggetto].
Cosa succede se **invertiamo l'ordine dei predittori**, dato che la partizione di tipo I non è invariante?

```
summary(aov(ep$desiderio_sex~ep$genoma*ep$gruppo))
                    Df Sum Sq Mean Sq  F value   Pr(>F)
ep$genoma            2  42.19  21.094    5.818  0.00511
ep$gruppo            1   1.05   1.049    0.289  0.59277
ep$genoma:ep$gruppo  2  28.56  14.279    3.938  0.02521
Residuals           55 199.42   3.626
```

In questo esempio, le decisioni su H_0 non cambiano, ma vediamo le SS_M attribuite a effetti principali e interazione: quella del genoma aumenta, passando da $SS_{M_A} = 39.25$ a $SS_{M_B} = 42.19$, e, contestualmente, quella del gruppo diminuisce (da 3.99 a 1.05). Come sappiamo, la SS_M di interazione resta invariata (28.56), così come quella d'errore ($SS_R = 199.42$).

Proviamo una partizione di tipo 3: la funzione probabilmente più facile è `ezANOVA`, in cui inseriremo l'argomento `between= .(X1,X2)` invece dell'argomento `within= X1` visto nel capitolo precedente, e specifichiamo il tipo di partizione con `type = 3`. Notate che l'elenco dei predittori non prevede l'usuale `c(X1,X2)`, ma la più insolita modalità `.(X1,X2)`.
Vediamo la prima parte dell'output:

```
ezANOVA(data = ep, dv = desiderio_sex, wid = soggetto, between = .(gruppo, genoma), type = 3, de
tailed = TRUE)

$ANOVA
        Effect DFn  DFd          SSn       SSd             F       p p<.05          ges
1 (Intercept)   1   55 2374.930804  199.4179   655.0125255  3.139165e-32    * 0.92253658
2 gruppo        1   55    1.880818  199.4179     0.5187349  4.744317e-01      0.00934342
3 genoma        2   55   39.235563  199.4179     5.4110525  7.157319e-03    * 0.16441445
4 gruppo:genoma 2   55   28.558781  199.4179     3.9382957  2.520732e-02    * 0.12527065
```

Di nuovo, risultano significativi gli effetti d'interazione (naturalmente, dato che la sua partizione non cambia) e di Genoma: la sua SS_M è molto simile a quella calcolata nel tipo I (39.235 versus 39.25). Evidentemente, il predittore non ha molta varianza in comune con l'effetto d'interazione, dato che la "restituzione" della varianza condivisa con $A \times B$ nella partizione di tipo I lascia quasi immutata la porzione di variabili spiegata da Genoma.

Il Generalized Eta Squared (η^2_G[103] - `ges`) nell'ANOVA a misure ripetute con un solo predittore era interpretabile come R^2 (proporzione di SS_T). Con **più di una X,** l'interpretazione del η^2_G cambia: è derivato dal **coefficiente eta quadrato parziale** (η^2_P[104]), che è un indicatore di **effect size relativo** (stima la relativa grandezza dell'effetto di un predittore rispetto agli altri) e rappresenta il rapporto tra la variabilità spiegata dal predittore rispetto alla variabilità non spiegata da alcun altro predittore ($\boldsymbol{SS_R} + \boldsymbol{SS_{M_{Xi}}}$): $\eta^2 = \frac{SS_{M_A}}{SS_T} \rightarrow \eta^2_p = \frac{SS_{M_A}}{SS_{M_A}+SS_R} \rightarrow \eta^2_G = \frac{SS_{M_A}}{\delta \times SS_{M_A} + \sum SS_{measured}}$.

In maniera del tutto **convenzionale**, e quindi da prendere con le opportune cautele, coefficienti $\eta^2_p \leq .06$ indicano un effetto parziale debole, da $\eta^2_p = .06$ a $\eta^2_p = .14$ un effetto moderato, $\eta^2_p > .14$ un effetto via via sempre più grande.

Soffermiamoci su un'altra informazione di questo output:

```
$ Levene's Test for Homogeneity of Variance
  DFn  DFd       SSn       SSd        F       p p<.05
1   5   55  12.84397  68.30357  2.068467  0.08328946
```

Come nelle misure ripetute, il test di Mauchly per la sfericità, nel caso dei campioni indipendenti abbiamo il **test di Levene per l'omoschedasticità** (capitolo 4) di A, B e $A \times B$ (contate i df…): in questo caso, con un po' di fortuna, l'omoschedasticità è rispettata.

Una curiosità: quando il disegno è sbilanciato e chiediamo una partizione di tipo II o III, `ezANOVA` dà sempre un **warning** sollecito, ma spassionato, sulla cautela necessaria per decidere quale partizione usare.

```
Warning: Data is unbalanced (unequal N per group). Make sure you specified a well-considered val
ue for type argument to ezANOVA().
```

Però, quando il disegno è sbilanciato e chiediamo incautamente una **partizione di tipo I**, pur producendo comunque un output, il `warning` di `ezANOVA` usa toni piuttosto accesi 😊

```
ezANOVA(data = ep, dv = desiderio_sex, wid = soggetto, between = .(gruppo, genoma), type = 1)
Warning: Using "type==1" is highly questionable when data are unbalanced and there is more
than one variable. Hopefully you are doing for demonstration purposes only!
```

Vediamo come si fa la stessa partizione di tipo 3 con `Anova` di **`car`**: l'applichiamo a un oggetto `lm`, esplicitiamo il tipo di partizione con `type= 3` e assegniamo contrasti ortogonali (`contr.sum`, `contr.poly` o `contr.helmert`) a entrambi i predittori con **`contrasts = list(X1= contrasto, X2= contrasto)`**:

```
Anova(lm(desiderio_sex~gruppo*genoma, data = ep, contrasts=list(gruppo = contr.poly,
  genoma = contr.poly)), type= 3)
ANOVA table (Type III tests)
Response: desiderio_sex
              Sum Sq  Df  F value    Pr (>F)
(Intercept) 2374.96   1  655.0125  < 2.2e-16
gruppo         1.88   1    0.5187   0.474432
genoma        39.24   2    5.4111   0.007157
gruppo:genoma 28.56   2    3.9383   0.025207
Residuals    199.42  55
```

[103] η^2_G inserisce al denominatore la variabilità individuale ($SS_{measured}$), e consente non solo il confronto di un fattore con gli altri nello stesso modello, ma anche il confronto dello stesso fattore in altri modelli riferiti alla stessa Y, purché le popolazioni siano le stesse ($\delta = 0$ nei disegni multilevel, $\delta = 1$ nei disegni fattoriali che trattiamo noi. Per dettagli sul calcolo: Molejnik & Algina, 2003; Bakeman, 2005).

[104] A differenza di η^2, che usa al denominatore tutta la devianza spiegata dal modello oltre a quella dell'errore, per ciascun effetto η^2_p **esclude** la devianza spiegata dagli **altri** effetti e somma solo la SS_M attribuita al predittore con la devianza d'errore,

A parte la disposizione, l'output è naturalmente identico a quello di ezANOVA.

Infine, vediamo una **partizione di tipo II**. Prima ezANOVA:

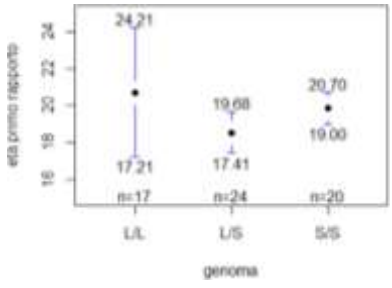
```
ezANOVA(data = ep, dv = desiderio_sex, wid = soggetto, between = .(gruppo, genoma),type = 2)
$ANOVA
          Effect DFn DFd         SSn       SSd        F      p p<.05          ges
1        gruppo   1  55  1.049342  199.4179  0.2894115  0.59276831   0.005234483
2        genoma   2  55 39.250732  199.4179  5.4127306  0.00714729   0.164457051
3 gruppo:genoma   2  55 28.558781  199.4179  3.9382957  0.02520732   0.125270648
```

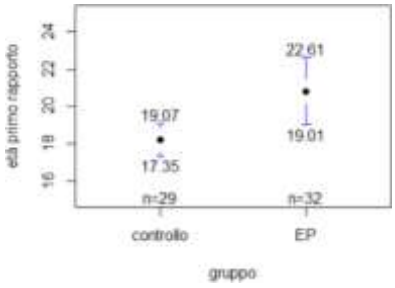
Poi Anova di car:

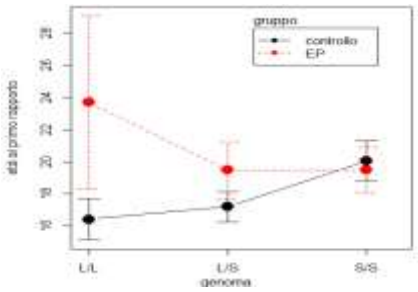
```
Anova(lm(desiderio_sex~gruppo*genoma, data = ep, contrasts=list(gruppo = contr.poly, genoma = co
 ntr.poly)), type= 2)
ANOVA table (Type II tests)
Response: desiderio_sex
                Sum Sq Df  F value  Pr (>F)
gruppo           1.049   1  0.2894 0.592768
genoma          39.251   2  5.4127 0.007147
gruppo:genoma   28.559   2  3.9383 0.025207
Residuals      199.418  55
```

Le SS_M di Gruppo e Genere sono abbastanza impercettibilmente diverse da quelle calcolate con il metodo III: evidentemente, la devianza comune ai due effetti principali, la cui sorte è diversa nei due metodi di partizione, è davvero poco rilevante.

Vediamo un caso in cui l'effetto principale significativo cambia, usando gli stessi predittori e un'altra Y: **l'età in cui i soggetti hanno avuto il primo rapporto sessuale** è diversa a seconda del gruppo di appartenenza, indipendentemente dal genoma? O a seconda del genoma, indipendentemente dal gruppo? O esiste un'interazione tra i due fattori? Sappiamo già che il disegno non è bilanciato, quindi ignoreremo la partizione di tipo I. Descriviamo i tre effetti (le barre di errore rappresentano i CI):








```
Desc(ep$eta_primo_rapporto~ep$genoma)
------------------------------------------------------------------------
            L/L      L/S      S/S
mean     20.706   18.542   19.850
median   19.000   18.000   20.000
sd        6.808    2.686    1.814
```

L'età al primo rapporto sessuale dei controlli sembra decisamente più bassa di quella dei pazienti, particolarmente variabili al loro interno, indipendentemente dal genoma. Meno definito sembra l'effetto principale del genoma, indipendentemente dal gruppo: l'età dei soggetti L/S sembra un po' più bassa di quella degli altri due gruppi, ma è soprattutto il gruppo L/L a sconcertare per la sua ampia variabilità. Infine, l'interazione: tra i controlli, i soggetti con genoma S/S tendono ad avere un'età più alta di quella degli altri due gruppi, tra loro molto simili. Al contrario, tra i pazienti i soggetti con genoma S/S tendono ad avere un'età al primo rapporto più bassa dei pazienti con genoma L/L (decisamente più variabili dei controlli con lo stesso genoma) e identica a quella dei pazienti con genoma L/S.

Poiché il disegno non è bilanciato, esprimiamo gli effetti calcolando le medie non pesate [all'esame andrà comunque benissimo usare le medie pesate per descrivere il dato!!!]:

```
controlli_LL<-ep[ep$genoma=="L/L" & ep$gruppo=="controllo",4]
controlli_LS<-ep[ep$genoma=="L/S" & ep$gruppo=="controllo",4]
controlli_SS<-ep[ep$genoma=="S/S" & ep$gruppo=="controllo",4]
pazienti_LL<-ep[ep$genoma=="L/L" & ep$gruppo=="EP",4]
pazienti_LS<-ep[ep$genoma=="L/S" & ep$gruppo=="EP",4]
pazienti_SS<-ep[ep$genoma=="S/S" & ep$gruppo=="EP",4]

(media_nonpesata_controlli<- (mean(controlli_LL) + mean(controlli_LS) + mean(controlli_SS))/3);
(media_nonpesata_pazienti<- (mean(pazienti_LL) + mean(pazienti_LS) + mean(pazienti_SS))/3)
[1] 17.90397
[1] 20.9
```

Per l'effetto principale del gruppo, Anova sarà applicata all'ipotesi nulla che l'età $\bar{x}_{controlli} = 17.91$ è solo casualmente diversa dall'età $\bar{x}_{pazienti} = 20.9$ (i due gruppi appartengono a una medesima popolazione).

```
(media_nonpesata_LL<- (mean(controlli_LL) + mean(pazienti_LL))/2);
(media_nonpesata_LS<- (mean(controlli_LL) + mean(pazienti_LS))/2);
(media_nonpesata_SS<- (mean(controlli_SS) + mean(pazienti_SS))/2)x
[1] 20.06429
[1] 17.96429
[1] 19.79167
```

Per l'effetto principale del genoma, Anova sarà applicata all'ipotesi nulla che le tre età medie 20.06, 17.96 e 19.79 sono solo casualmente diverse (i tre gruppi appartengono a una medesima popolazione).

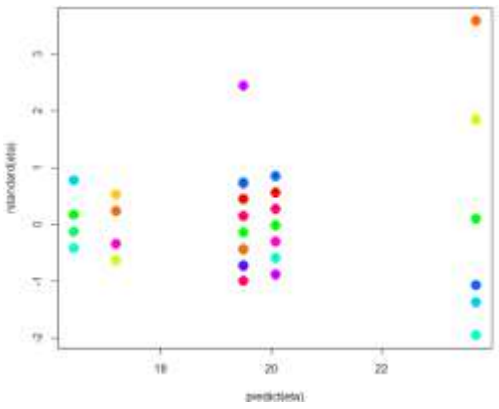
Per l'effetto dell'interazione genoma, Anova sarà applicata all'ipotesi nulla che le differenze tra le età 16.3, 17.20 e 20.08 dei controlli sono solo casualmente differenti dalle differenze tra le età 23.7, 19.5 e 19.5 nei pazienti (oppure, se preferite, che la differenza tra le età dei due gruppi 16.43 e 23.7 nel genoma L/L è solo casualmente differente dalla differenza tra 17.2 e 19.5 nel genoma L/S e dalla differenza tra 20.08 e 19.5 nel genoma S/S).

```
ezANOVA (data=ep, dv=eta_primo_rapporto, wid = soggetto, between = .(gruppo, genoma),type = 3)
$ANOVA
          Effect   DFn DFd      SSn      SSd           F         p p<.05         ges
1        gruppo     1  55   111.51385  719.831    8.520420   0.005080010   0.1341367
2        genoma     2  55    57.10973  719.831    2.181787   0.122510423   0.0735059
3 gruppo:genoma     2  55   138.69294  719.831    5.298544   0.007864092   0.1615481

$ 'Levene's Test for Homogeneity of Variance
  DFn  DFd    SSn          SSd         F       p p<.05
1   5   55  145.1056   491.3452  3.248554   0.01217753
```

Indipendente dal genoma, l'età del primo rapporto dei controlli è significativamente inferiore a quella degli EP. Indipendentemente dal gruppo, il genoma non determina differenze significative nell'età. L'interazione è significativa: quando il genoma è L/L, l'età del primo rapporto degli EP è molto più alta di quella dei controlli; la differenza si riduce quando il genoma è L/S e diminuisce ulteriormente, arrivando anzi a invertirsi, quando il genoma è S/S. L'entità dell'effetto principale del gruppo e quella dell'interazione sono sostanzialmente equivalenti.

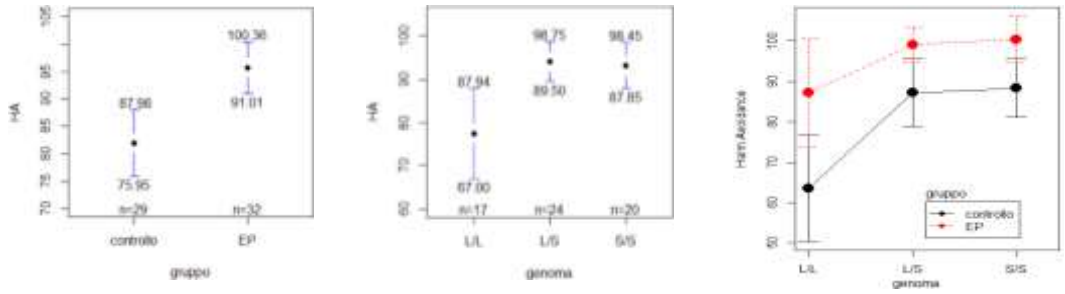
Purtroppo, **l'omoschedasticità è violata**, come si poteva sospettare osservando l'errore nel gruppo EP-L/L, con il suo CI tanto più ampio degli altri; d'altronde, il mai dimenticato grafico diagnostico predetti per residui del modello lineare lo dice chiaramente:

```
eta<-lm(ep$eta_primo_rapporto~ep$gruppo*ep$genoma)
plot(predict(eta),    rstandard(eta),    pch=19,    cex=2,
col=rainbow(15))
```



Vedremo nel paragrafo 7.3 come affrontare questo problema.

Chiudiamo, per completezza, con un esempio in cui **entrambi gli effetti principali sono significativi, ma l'interazione no**: usiamo i soliti predittori genoma e gruppo sulla Y **Harm Avoidance** (che, ricordiamo, è una misura di labilità emotiva).



Indipendentemente dal loro genoma, l'Harm Avoidance dei controlli sembra inequivocabilmente inferiore a quello dei pazienti, anche se questi soggetti sono un po' più variabili al loro interno. Indipendentemente dal gruppo, l'Harm Avoidance dei soggetti con genoma L/L è nettamente più basso, ma molto più variabile (di nuovo), di quello degli altri due gruppi. L'interazione non sembra promettere molto: le differenze tra L/L, L/S e S/S nei controlli sono molto simili alle differenze tra L/L, L/S e S/S nei pazienti.

```
ezANOVA(data=ep, dv = HA, wid = soggetto, between = .(gruppo, genoma),detailed= TRUE, type = 3)
$ANOVA

         Effect  DFn DFd        SSn       SSd           F         p p<.05         ges

1(Intercept)       1  55 443954.157  8022.385 3043.668444  7.694750e-50     0.9822504
2 gruppo           1  55  36307.165  8022.385   24.887749  6.446207e-06     0.3115340
3 genoma           2  55   3991.770  8022.385   13.683424  1.502387e-05     0.3322556
4 gruppo:genoma    2  55    405.030  8022.385    1.388406  2.580779e-01     0.0480610

$ 'Levene's Test for Homogeneity of Variance
  DFn  DFd       SSn     SSd          F  p p<.05
1   5   55  280.0675 4704.67 0.6548264 0.6590809
```

Indipendente dal genoma, il tratto HA degli EP è significativamente maggiore di quello dei controlli. Indipendentemente dal gruppo, il tratto HA è significativamente differente a seconda del genoma. Il genoma pesa solo impercettibilmente più del gruppo. L'interazione non è significativa: la differenza tra genoma L/L, L/S e S/S è uguale in EP e controlli – ovvero, la differenza nel tratto HA tra EP e controlli è la stessa quando il genoma è L/L, L/S e S/S. L'omoschedasticità, per fortuna, è rispettata.

Anche con una partizione di tipo II arriviamo alle stesse conclusioni:

```
Anova(lm(HA~gruppo*genoma, data = ep, contrasts= list(gruppo=contr.sum, genoma= contr.sum)),
   type=2)
Anova Table (Type II tests)
Response: HA
              Sum Sq Df F value    Pr(>F)
gruppo        3412.0  1 23.3920 1.105e-05
genoma        3780.4  2 12.9590 2.448e-05
gruppo:genoma  405.0  2  1.3884    0.2581
Residuals     8022.4 55
```

Oltre al *generalized eta squared*, altri coefficienti di effect size stimano l'effetto di una X al netto delle altre. Per esempio, possiamo usare il coefficiente f **di Cohen**, che **varia da 0 a infinito**; Cohen suggerisce di interpretare effetti con $f < .10$ come trascurabili, con f **da** $f = .11$ **a** $f = .25$ deboli, da $f = .25$ **a** $f = .40$ medi, con $f > .40$ grandi. Possiamo usare il noto package `effectsize` con la funzione `cohens_f(modello)`, che accetta oggetti `lm` o `aov` (non `ezAnova`, perciò) e fornisce anche un utile CI (di default, al 90%):

```
cohens_f(model = Anova(lm(HA~gruppo*genoma, data = ep, contrasts= list(gruppo=contr.sum, genoma=
contr.sum)), type=3))
# Effect Size for ANOVA (Type III)

Parameter     | Cohen's f (partial) |       90% CI
-------------------------------------------------
gruppo        |                0.67 | [0.42, 0.92]
genoma        |                0.71 | [0.44, 0.94]
gruppo:genoma |                0.22 | [0.00, 0.42]
```

Il coefficiente f concorda con il ges di ezANOVA nel definire gli effetti principali di intensità praticamente equivalente – e piuttosto grande -, mentre l'effetto di interazione è debole.

Notate bene: si può calcolare il coefficiente ***f*** **anche per modelli con una sola *X*** e con **una o più *X* continue**: in tutti questi casi, l'interpretazione delle soglie di intensità è la stessa.

```
cohens_f(model = aov(ep$HA~ep$genoma))
For one-way between subjects designs, partial eta squared is equivalent to eta squared.
Returning eta squared.
# Effect Size for ANOVA

Parameter | Cohen's f |       90% CI
-----------------------------------
ep$genoma |      0.52 | [0.27, 0.74]

cohens_f(model = lm(ep$IIEF~ep$HA), ci=.95)
For one-way between subjects designs, partial eta squared is equivalent to eta squared.
Returning eta squared.
# Effect Size for ANOVA

Parameter | Cohen's f |       95% CI
-----------------------------------
ep$HA     |      0.55 | [0.27, 0.82]

cohens_f(model = lm(ep$IIEF~ep$HA*ep$frequenza_sex_mese), ci=.95)
# Effect Size for ANOVA (Type I)

Parameter                   | Cohen's f (partial) |       95% CI
-------------------------------------------------------------------
ep$HA                       |                0.70 | [0.39, 1.00]
ep$frequenza_sex_mese       |                0.72 | [0.35, 0.99]
ep$HA:ep$frequenza_sex_mese |                0.32 | [0.00, 0.53]
```

Possiamo ora approfondire i confronti a coppie tra i livelli. Per gli effetti principali, è semplice: se il fattore ha più di due livelli si possono impostare i contrasti desiderati, oppure (ma solo se risulta significativo nel modello *overall*), i test post hoc. Nel primo esempio, l'effetto del genoma sul desiderio è significativo, quindi possiamo impostare:

```
pairwise.t.test(ep$desiderio_sex, ep$genoma, paired = FALSE, p.adjust.method = "b")
       Pairwise comparisons using t tests with pooled SD
data:  ep$desiderio_sex and ep$genoma
     L/L     L/S
L/S  0.3061  -
S/S  0.0056  0.2243

P value adjustment method: bonferroni
```

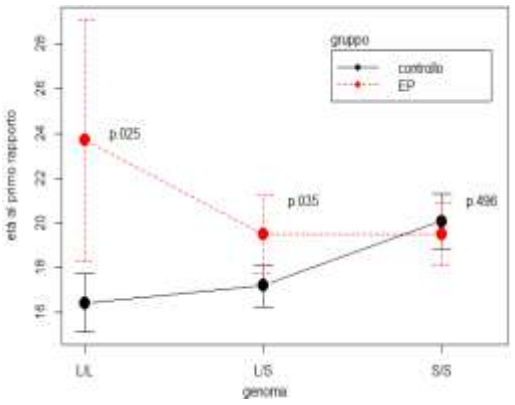
L'unica differenza significativa tra i livelli è tra gruppo L/L e gruppo S/S.

invece, quando la variabile dipendente è il tratto di Harm Avoidance, il gruppo L/L ha un tratto significativamente inferiore a quello di entrambi gli altri gruppi:

```
pairwise.t.test(ep$HA, ep$genoma, paired = FALSE, p.adjust.method = "b")
       Pairwise comparisons using t tests with pooled SD
data:  ep$HA and ep$genoma
     L/L     L/S
L/S  0.0016  -
S/S  0.0046  1.0000

P value adjustment method: bonferroni
```

Per l'interazione, i risultati più interessanti sono solitamente quelli relativi al **confronto tra livelli appartenenti a un fattore, tenuto fisso l'altro**: per esempio, nel modello Gruppo×Genoma riferito all'età al primo rapporto sarebbe interessante sapere se l'apparente grande differenza tra gruppi con genoma L/L si annulla per il genoma L/S e S/S → **tenuto fisso il genoma**, c'è differenza a seconda del gruppo di appartenenza? **L'analisi degli effetti semplici**, o *simple effects analysis*, valuta l'effetto di X_1 per tutti i singoli livelli dell'altra X_2 e viceversa (*l'effetto del gruppo in ogni genoma*, e/o l'effe*tto del genoma in ogni gruppo*). La *simple effects analysis* **potrebbe assomigliare** al fare ***k* ANOVA univariate** su subset differenti (tre ANOVA univariate in cui si valuta la differenza, in ogni genoma, tra i due gruppi, e due ANOVA in cui si ricerca la differenza, per ogni gruppo, tra i genomi). In realtà, l'analisi degli effetti semplici è **preferibile**, perché **fare più analisi univariate NON è un metodo consigliabile**: in queste ANOVA, infatti, le MS_M sono confrontate con MS_R e relativi df_R diversi. Verifichiamolo con la variabile Età al primo rapporto; cominciamo a vedere se ci sono differenze tra i due gruppi quando il genoma è L/L, L/S e S/S con la sintetica informazione di `aov(Y~X)`:

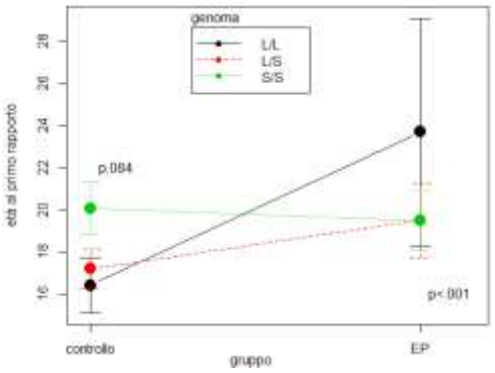
```
L_L<-subset(ep, ep$genoma=="L/L")
L_S<-subset(ep, ep$genoma=="L/S")
S_S<-subset(ep, ep$genoma=="S/S")

summary(aov(L_L$eta_primo_rapporto~L_L$gruppo))
             Df Sum Sq Mean Sq F value Pr(>F)
L_L$gruppo    1  217.7  217.72   6.235 0.0247
Residuals    15  523.8   34.92

summary(aov(L_S$eta_primo_rapporto~L_S$gruppo))
             Df Sum Sq Mean Sq F value Pr(>F)
L_S$gruppo    1  30.86  30.858   5.025 0.0354
Residuals    22 135.10   6.141

summary(aov(S_S$eta_primo_rapporto~S_S$gruppo))
             Df Sum Sq Mean Sq F value Pr(>F)
S_S$gruppo    1   1.63   1.633   0.483  0.496
Residuals    18  60.92   3.384
```



La differenza tra i gruppi è significativa quando il genoma è L/L e L/S, ma non quando il genoma è S/S; notate che la SS_R nel modello riferito al genoma S/S è la metà della SS_R del gruppo L/S e quasi 9 volte inferiore a quella del modello del gruppo L/L; notate anche la grande differenza tra i df_R.

Ora vediamo le differenze tra i genomi in ogni gruppo:

```
pz<-subset(ep, ep$gruppo=="EP")
crl<-subset(ep, ep$gruppo=="controllo")

summary(aov(pz$eta_primo_rapporto~pz$genoma))
               Df Sum Sq Mean Sq F value Pr(>F)
pazienti$genoma  2  121.3   60.64   2.699 0.0842
Residuals       29  651.6   22.47

summary(aov(crl$eta_primo_rapporto~crl$genoma))
                 Df Sum Sq Mean Sq F value   Pr(>F)
controlli$genoma  2  74.53   37.26    14.2 6.79e-05
Residuals        26  68.23   2.62
```



La differenza tra i genomi non è significativa tra i controlli, mentre lo è tra i pazienti (anche se pare abbastanza evidente che sia il genoma L/L a definirla come tale); di nuovo, la SS_R nel modello riferito ai controlli è quasi 10 volte inferiore a quella del modello dei pazienti, come sono diversi i df_R.

Qual è, quindi, il **problema** del fare più univariate tra i livelli di ogni fattore? Mentre, come detto, i contrasti pianificati non toccano la SS_R del modello – e di conseguenza la MS_R – in quanto sono diverse spartizioni della sola MS_M, con più univariate la MS_M di ogni modello è messa a confronto, volta per volta, con le **differenti MS_R dei rispettivi modelli**.

È meglio, invece, utilizzare la MS_R dell'intero modello **bivariato**, creando un **unico fattore composto dai livelli di X_1 e X_2** (nel nostro esempio, un fattore a 3×2 livelli) e inserendo in un modello $Y \sim fattore\ composto$, di cui si valutano i contrasti d'interesse o i confronti post hoc: questa è la *simple effect analysis*.

Ad esempio, costruiamo il "superfattore" con la funzione `paste(x1, x2)`, che concatena due variabili – stringa o factor in una nuova variabile `character`; ci servirà un fattore, quindi costruiamolo nidificando `paste` in `as.factor`:

```
ep$concatena<-as.factor(paste(ep$gruppo, ep$genoma))
class(ep$concatena)
[1] "factor"
```

Vediamo il "superfattore":

```
ep[1:6,c(2,3,13)]
  gruppo genoma concatena
1     EP    L/L    EP L/L
2     EP    L/L    EP L/L
3     EP    L/L    EP L/L
4     EP    L/L    EP L/L
5     EP    L/L    EP L/L
6     EP    L/L    EP L/L

ep[18:23,c(2,3,13)]
  gruppo genoma concatena
18     EP    L/S    EP L/S
19     EP    L/S    EP L/S
20     EP    L/S    EP L/S
21     EP    L/S    EP L/S
22     EP    L/S    EP L/S
23     EP    L/S    EP L/S

ep[56:61,c(2,3,13)]
   gruppo genoma     concatena
56 controllo    S/S controllo S/S
57 controllo    S/S controllo S/S
58 controllo    S/S controllo S/S
59 controllo    S/S controllo S/S
60 controllo    S/S controllo S/S
61 controllo    S/S controllo S/S
```

Inseriamo il fattore composito nel modello lineare $Y \sim superfattore$, **chiedendo i contrasti desiderati**; ad esempio, visto che non l'abbiamo ancora visto all'opera, usiamo il **contrasto SAS**, che considera come gruppo di controllo l'ultimo livello: `contr.SAS(n = numero di livelli)`:

```
contrasts(ep$concatena)<-contr.SAS(n = 6)
contrasts(ep$concatena)
             1 2 3 4 5
controllo L/L 1 0 0 0 0
controllo L/S 0 1 0 0 0
controllo S/S 0 0 1 0 0
EP L/L        0 0 0 1 0
EP L/S        0 0 0 0 1
EP S/S        0 0 0 0 0
```

La componente di errore di questo modello univariato è uguale a quella del modello bivariato Gruppo×Genoma:

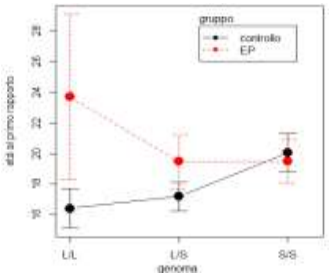
```
summary(aov(ep$eta_primo_rapporto~ep$gruppo*ep$genoma))
                   Df Sum Sq Mean Sq  F value  Pr(>F)
ep$gruppo           1  103.3  103.28   7.892  0.00686
ep$genoma           2   57.1   28.55   2.182  0.12251
ep$gruppo:ep$genoma 2  138.7   69.35   5.299  0.00786
Residuals          55  719.8   13.09

summary(aov(ep$eta_primo_rapporto~ep$concatena))
             Df Sum Sq Mean Sq F value  Pr(>F)
ep$concatena  5  299.1   59.82    4.57 0.00148
Residuals    55  719.8   13.09
```

Quindi possiamo passare a esaminare il contrasto che ci interessa usando `lm(Y~superfattore)`:

```
summary(lm(ep$eta_primo_rapporto~ep$concatena))

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.950e+01  1.279e+00  15.246   <2e-16
ep$concatena1 -3.071e+00  1.872e+00  -1.640   0.1066
ep$concatena2 -2.300e+00  1.716e+00  -1.340   0.1857
ep$concatena3  5.833e-01  1.651e+00   0.353   0.7252
ep$concatena4  4.200e+00  1.716e+00   2.448   0.0176
ep$concatena5  1.532e-15  1.603e+00   0.000   1.0000
```

```
> contrasts(ep$concatena)
             1 2 3 4 5
controllo L/L 1 0 0 0 0
controllo L/S 0 1 0 0 0
controllo S/S 0 0 1 0 0
EP L/L        0 0 0 1 0
EP L/S        0 0 0 0 1
EP S/S        0 0 0 0 0
```

È il **significativo** solo **il contrasto 4**, che la tabella dei contrasti ci dice essere "pazienti L/L versus pazienti S/S". I pazienti EP con alleli S/S hanno un'età al primo rapporto significativamente inferiore solo a quella dei pazienti EP con alleli L/L. non sono invece significativamente diversi dagli altri gruppi.



Se, invece, volessimo applicare i confronti post hoc al fattore composto:

```
pairwise.t.test(ep$eta_primo_rapporto, ep$concatena, p.adjust.method = "b")
```

```
        Pairwise comparisons using t tests with pooled SD

data:  ep$eta_primo_rapporto and ep$concatena

             controllo L/L controllo L/S controllo S/S EP L/L   EP L/S
controllo L/S 1.0000        -             -           -        -
controllo S/S 0.5725        1.0000        -           -        -
EP L/L        0.0022        0.0027        0.3485      -        -
EP L/S        1.0000        1.0000        1.0000      0.1044   -
EP S/S        1.0000        1.0000        1.0000      0.2641   1.0000
```

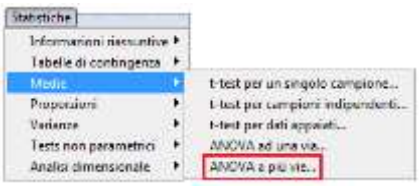
Tenuto fisso il gruppo, tra i pazienti EP non si rilevano differenze a coppie tra i genomi (.104, .264, 1.0), così come tra i controlli (1.0, .573, 1.0). Tenuto fisso il con genoma, si riscontrano differenze significative tra i due gruppi quando il genoma è L/L (.002), ma non quando è L/S (1.0) o S/S (1.0).

Si perde la significatività della differenza del precedente contrasto, a causa della severa correzione per il family-wise error rate: guardate cosa succede eliminando la correzione:

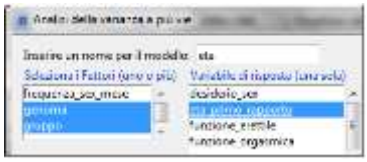
```
pairwise.t.test(ep$eta_primo_rapporto, ep$concatena, p.adjust.method = "none")
```

```
        Pairwise comparisons using t tests with pooled SD
data:  ep$eta_primo_rapporto and ep$concatena

             controllo L/L controllo L/S controllo S/S EP L/L   EP L/S
controllo L/S 0.66693       -             -           -        -
controllo S/S 0.03817       0.06803       -           -        -
EP L/L        0.00015       0.00018       0.02323     -        -
EP L/S        0.07206       0.13039       0.68349     0.00696  -
EP S/S        0.10663       0.18566       0.72524     0.01761  1.00000

P value adjustment method: none
```

Per concludere, se volete usare `Rcommander` →



→



Il prodotto di default è un'ANOVA di tipo 2 (con `Anova` di `car` applicata a un modello creato con `lm`), con l'aggiunta di medie, *sd* e numerosità dei livelli. Nello script appare:

```
età <- lm(eta_primo_rapporto ~ genoma*gruppo, data=ep, contrasts=list(genoma ="contr.Sum", gruppo
="contr.Sum"))
Anova(età, type=3)
with(ep, (tapply(eta_primo_rapporto, list(genoma, gruppo), mean, na.rm=TRUE))) # means
with(ep, (tapply(eta_primo_rapporto, list(genoma, gruppo), sd, na.rm=TRUE))) # std. deviations
xtabs(~ genoma + gruppo, data=ep) # counts
```

Nell'ouput si leggono:

```
età <- lm(eta_primo_rapporto ~ genoma*gruppo, data=ep, contrasts=list(genoma ="contr.Sum", gruppo
="contr.Sum"))
Anova(età)
Anova Table (Type II tests)

Response: eta_primo_rapporto
             Sum Sq Df F value   Pr(>F)
genoma        57.11  2  2.1818 0.122510
gruppo       111.51  1  8.5204 0.005080
genoma:gruppo 138.69  2  5.2985 0.007864
Residuals     719.83 55
with(ep, (tapply(eta_primo_rapporto, list(genoma, gruppo), mean, na.rm=TRUE))) # means
    controllo   EP
L/L  16.42857 23.7
L/S  17.20000 19.5
S/S  20.08333 19.5
```

```
with(ep, (tapply(eta_primo_rapporto, list(genoma, gruppo), sd, na.rm=TRUE))) # std. deviations
controllo       EP
L/L  1.397276 7.543209
L/S  1.316561 3.031882
S/S  1.928652 1.690309
xtabs(~ genoma + gruppo, data=ep) # counts
gruppo
genoma controllo EP
   L/L         7 10
   L/S        10 14
   S/S        12  8
```

Per cambiare il tipo di partizione, si modifica nello script l'argomento `type=` di `Anova`:

```
età <- lm(eta_primo_rapporto ~ genoma*gruppo, data=ep, contrasts=list(genoma ="contr.Sum", gruppo ="contr.Sum"))

Anova(età, type=3)
Anova Table (Type III tests)
Response: eta_primo_rapporto
Sum Sq Df   F value    Pr(>F)
(Intercept)   21765.7  1 1663.0471 < 2.2e-16
genoma           35.3  2    1.3494  0.267858
gruppo          129.8  1    9.9139  0.002650
genoma:gruppo   138.7  2    5.2985  0.007864
Residuals       719.8 55
```

13.2 ANOVA fattoriale a misure ripetute e a misure ripetute mista

*In questo paragrafo useremo i dataframe **simon** e **sicurezza***

Nell'analisi della varianza **fattoriale a misure ripetute**, il modello lineare prevede una sola Y predetta da almeno due X categoriali, **entrambe somministrate entro soggetti**: conterrà quindi, oltre a b_0, anche i b_1 degli effetti principali e della loro interazione (per modelli con interazione). Nell'analisi della varianza a **misure ripetute** mista, il modello lineare vede una Y predetta da almeno due X categoriali, di cui **almeno una** somministrata **between groups** e **almeno una** somministrata **entro** soggetti.

Nella logica della partizione della devianza, dei contrasti e dei post hoc ben poco cambia rispetto all'analisi fattoriale per gruppi indipendenti (ricordiamo solo che per i fattori a misure ripetute la partizione di SS_T prevede che sia la devianza entro i soggetti a essere ripartita in SS_M e SS_R), così come la verifica della sfericità è del tutto analoga a quella dell'ANOVA a misure ripetute a una via. Concentriamoci quindi su una cosa nuova: useremo l'ANOVA mista per vedere anche un **esempio di interazione a tre vie:** $X_1 \times X_2 \times X_3$.

13.2.1 ANOVA fattoriale a misure ripetute

Vediamo **l'ANOVA fattoriale a misure ripetute** usando un classico esperimento sui tempi di reazione che sfrutta il paradigma sperimentale noto come **Simon effect** (dataframe **simon**) in una delle sue modalità più semplici: 22 soggetti hanno risposto alla comparsa di uno stimolo su uno schermo in quattro condizioni sperimentali: usando l'effettore in **posizione compatibile** (X_{1_a}: mano dx - stimolo a dx, mano sx - stimolo a sx) o **incompatibile** (X_{1_b} -mano dx - stimolo a sx, mano sx - stimolo a dx) con la posizione dello stimolo sullo schermo; ogni volta lo stimolo era preceduto da un *cue* visivo che poteva avere la **stessa forma** (X_{2_a}) o una **forma diversa** (X_{2_b}) dello stimolo – target.

$X_{1_a}X_{2_a}$

$X_{1_b}X_{2_a}$

$X_{1_a}X_{2_b}$

$X_{2_a}X_{2_b}$

Il modello lineare prevede quindi:

- effetto principale della **Posizione** (X_1) → H_0: $\bar{X}_{X_{1a}} = \bar{X}_{X_{1b}}$, indipendentemente dalla forma del *cue;*
- effetto principale della **Forma** (X_2) → H_0: $\bar{X}_{X_{2a}} = \bar{X}_{X_{2b}}$, indipendentemente dalla posizione dello stimolo;
- interazione **Posizione per Forma** ($X_1 \times X_2$): → H_0: $\bar{X}_{X_{1a}X_{2a}} - \bar{X}_{X_{1b}X_{2a}} = \bar{X}_{X_{1a}X_{2b}} - \bar{X}_{X_{2a}X_{2b}}$, ovvero la differenza nei RT tra posizione compatibile e incompatibile quando il *cue* ha la stessa forma dello stimolo è uguale alla differenza nei RT tra posizione compatibile e incompatibile quando la forma del *cue* è diversa *[oppure:* H_0: $\bar{X}_{X_{1a}X_{2a}} - \bar{X}_{X_{1a}X_{2b}} = \bar{X}_{X_{1b}X_{2a}} - \bar{X}_{X_{1b}X_{2b}}$*, cioè la differenza nei RT tra cue uguale e diverso quando la posizione è compatibile è uguale alla differenza nei RT tra cue uguale e diverso quando la posizione è incompatibile]*.

Le ipotesi alternative dovrebbero prevedere una facilitazione – ovvero una riduzione dei RT – quando la posizione stimolo – effettore è compatibile e il *cue* ha la stessa forma dello stimolo che anticipa.

Nel dataset in formato *wide* **ogni colonna rappresenta una condizione** $x_{1j}x_{2j}$:

```
head(simon)
```

	$X_{1_a}X_{2_a}$	$X_{1_b}X_{2_a}$	$X_{1_a}X_{2_b}$	$X_{2_a}X_{2_b}$
sogg	compatibile_uguale	compatibile_diverso	incompatibile_uguale	incompatibile_diverso
s1	468.6	529.2	477.1	546.1
s2	435.0	502.5	487.5	535.5
s3	493.7	599.1	539.8	586.8
s4	496.4	512.0	470.5	463.6
s5	410.7	442.7	393.2	477.6
s6	423.9	473.8	408.5	478.8

Prepariamo il dataset in formato long e descriviamo i dati:

```
simon2<-melt(data = simon, id.vars = "sogg", measure.vars = c ("compatibile_uguale", "compatibil
e_diverso", "incompatibile_uguale", "incompatibile_diverso"))
```

```
 head(simon2)
   sogg           variable  value
1    s1 compatibile_uguale  468.6
2    s2 compatibile_uguale  435.0
3    s3 compatibile_uguale  493.7
4    s4 compatibile_uguale  496.4
5    s5 compatibile_uguale  410.7
6    s6 compatibile_uguale  423.9
```

```
tail(simon2)
   sogg             variable  value
83  s17 incompatibile_diverso  598.7
84  s18 incompatibile_diverso  515.5
85  s19 incompatibile_diverso  624.4
86  s20 incompatibile_diverso  572.5
87  s21 incompatibile_diverso  531.4
88  s22 incompatibile_diverso  490.8
```

I fattori $X_{1\ Posizione}$ e $X_{2\ Forma}$ sono "intrecciati" nel `factor` a misure ripetute `variable`: dobbiamo quindi separarli in due variabili diverse da inserire nel modello lineare. Potete verificare facilmente (`View(simon2)`) che le prime 44 righe del dataframe in long format sono composte dalle misure in posizione compatibile, da 1 a 22 con *cue* di stessa forma, da 23 a 44 con *cue* di diversa forma; le successive 44 righe contengono invece le misure delle posizioni incompatibili: da 45 a 66 con *cue* di uguale forma, da 67 a 88 con *cue* diverso. Perciò, possiamo inserire il fattore `$posizione` replicando per le prime 44 volte l'etichetta "`compatibile`" e per le seconde 44 "`incompatibile`":

```
simon2$posizione<-c(rep("compatibile",44), rep("incompatibile",44))
```

Il fattore **$forma** sarà invece creato ripetendo per le prime 22 volte "uguale", per le successive 22 "diverso", poi ancora per 22 volte "uguale" concludendo con 22 "diverso":

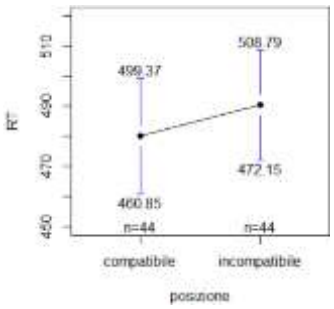
```
simon2$forma<-c(rep("uguale", 22), rep("diversa",22), rep("uguale",22), rep("diversa",22))
```

Diamo nomi e verifichiamo:

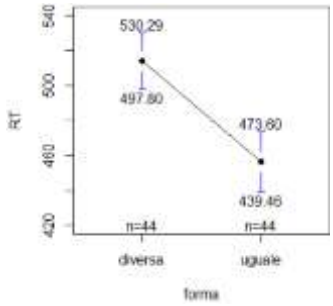
```
names(simon2)<-c("sogg", "ripetute","RT", "posizione", "forma")
 head(simon2,3); tail(simon2,3)
   sogg           ripetute          value   posizione        forma
 1   s1 compatibile_uguale          468.6   compatibile      uguale
 2   s2 compatibile_uguale          435.0   compatibile      uguale
 3   s3 compatibile_uguale          493.7   compatibile      uguale
    sogg              ripetute      value     posizione       forma
 86  s20 incompatibile_diverso      572.5   incompatibile    diversa
 87  s21 incompatibile_diverso      531.4   incompatibile    diversa
 88  s22 incompatibile_diverso      490.8   incompatibile    diversa
```

Le due nuove variabili sono di classe `character`: **riclassifichiamole** con `as.factor` e descriviamo i dati.

```
tapply(simon2$RT, simon2$posizione, mean)
   compatibile incompatibile
     480.1068       490.4705
tapply(simon2$RT, simon2$posizione, sd)
  compatibile incompatibile
     63.35526       60.25722
```



I RT nella posizione compatibile, indipendentemente dalla forma del *cue,* sono un po' più rapidi, anche se la differenza è piccola e la variabilità ampia.



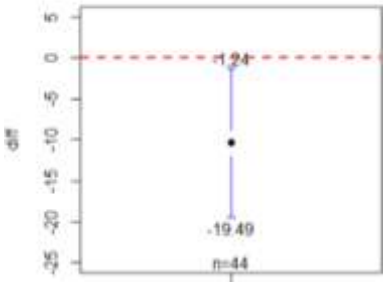

```
tapply(simon2$RT, simon2$forma, mean)
  diversa    uguale
 514.0455  456.5318
tapply(simon2$RT, simon2$forma, sd)
  diversa   uguale
 53.42250 56.15547
```

Quando il *cue* è uguale I RT sono decisamente più rapidi.

Naturalmente, ci ricordiamo che con **misure ripetute** non possiamo interpretare la sovrapposizione dei CI delle medie per stimare la significatività della differenza, perché non prendono in considerazione la **correlazione** tra le misure. Dovremmo / potremmo, invece, rappresentare il CI della media della differenza tra i livelli contro il valore atteso da H_0: dato che la correlazione positiva tra le misure è forte per entrambi i predittori, questi grafici non sono informativi sulla reale significatività:

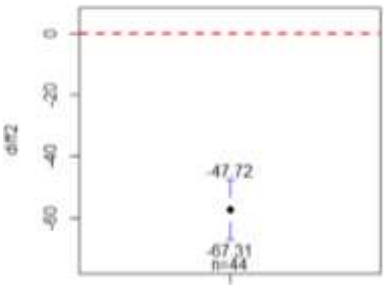
```
compatibile<-simon2$RT[1:44]
incompatibile<-simon2$RT[45:88]
diff<-compatibile-incompatibile
cor(compatibile, incompatibile)
[1] 0.8832552
```

effetto Posizione



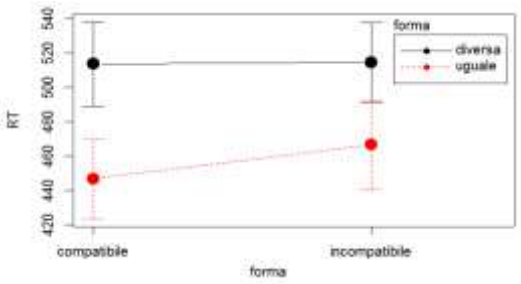

```
uguale<-simon2$RT[c(1:22,45:66)]
diverso<-simon2$RT[c(23:44,67:88)]
diff2<-uguale-diverso
cor(uguale,diverso)
[1] 0.8282781
```

effetto Forma



Entrambi gli effetti principali dovrebbero essere significativi, anche se l'effetto della forma del *cue* pare più incisivo.

```
tapply(simon2$RT,list(simon2$posizionesimon2$forma),mean)
                 diversa    uguale
compatibile       513.5682  446.6455
incompatibile  514.5227  466.4182

tapply(simon2$RT,list(simon2$posizione, simon2$forma),sd)
                diversa   uguale
compatibile    55.543425  2.80042
incompatibile  52.519355  8.85885
```



Quando la forma del *cue* è diversa da quella dello stimolo (traccia nera), i RT della posizione compatibile e di quella incompatibile sono praticamente identici; invece, quando la forma del *cue* è uguale a quella dello stimolo (traccia rossa), la posizione compatibile della mano ha un effetto di facilitazione più sensibile rispetto a quello della posizione incompatibile.

Usiamo `ezANOVA`: essendo un disegno a sole misure ripetute senza NA, perfettamente bilanciato (stesso numero di soggetti in tutte le condizioni), la partizione della devianza non è un problema; possiamo usare il tipo II di default. Dato che entrambi i fattori hanno solo due livelli, inoltre, anche la verifica della sfericità non è eseguita. I due predittori within sono indicati nell'argomento **`within = .(X1, X2)`**:

```
ezANOVA(data = simon2, dv = RT, wid = sogg, within = .(posizione,forma))
$ANOVA
            Effect DFn DFd          F         p p<.05          ges
2       posizione   1  21   5.058065  3.536943e-02   0.009216772
3           forma   1  21 119.842139  3.882222e-10   0.222694413
4 posizione:forma   1  21   5.372634  3.063079e-02   0.007609460
```

La posizione ha un effetto **significativo** (indipendentemente dal *cue,* i RT della posizione compatibile sono più rapidi), così come **l'interazione** (la differenza tra posizione compatibile e incompatibile a seconda del tipo di *cue*), ma l'η_p^2 dice che è **molto più forte, oltre che significativo, l'effetto della forma del *cue***: quando il *cue* ha la stessa forma dello stimolo, i RT del soggetto sono molto più veloci, indipendentemente dalla compatibilità delle posizioni stimolo – effettore.

In analogia a quanto visto nel paragrafo precedente, ***eseguite le opportune analisi post hoc*** *per questo modello lineare: ora il dataframe contiene tutto quello che vi serve...*

13.2.2 ANOVA fattoriale a misure ripetute mista

Eseguiamo un'ANOVA **fattoriale a misure ripetute mista** partendo dal caso più semplice: un predittore a misure ripetute e un predittore a gruppi indipendenti.

Riprendiamo il dataset **`sicurezza`** e finalmente vediamo all'opera anche il gruppo di controllo il controllo:

```
addmargins(table(sicurezza$gruppo))
  Controllo  formazione non obbligatoria     formazione obbligatoria   Sum
         32                          35                          56    23
```

Il disegno non è bilanciato.

Verifichiamo se, indipendentemente dal gruppo di appartenenza, si è verificato un cambiamento significativo nelle **conoscenze** ($\boldsymbol{Y}$) da T_0 a T_2 [$\boldsymbol{X_{1\,tempo}}$, misure ripetute]; se, indipendentemente dal momento in cui sono rilevate, esiste una differenza nelle conoscenze dei tre gruppi (formazione obbligatoria, formazione non obbligatoria, nessuna formazione: $\boldsymbol{X_{2\,Gruppo}}$]; se il cambiamento nelle conoscenze da T_0 a T_2 avviene in maniera differente a seconda che il soggetto appartenga all'uno o all'altro dei tre gruppi [**interazione a due vie** $X_{1\,Tempo} \times X_{2Gruppo}$]. Poiché $X_{1\,Tempo}$ è a

misure ripetute, dobbiamo cambiare il formato del dataframe. Ci sono **due variabili uniche**, ovvero due variabili per cui la misura del soggetto è unica e non ripetuta: il **codice** che lo identifica e il **gruppo** cui appartiene: lei concateniamo nell'argomento **id.vars=**:

```
melt_sic<-melt(data = sicurezza, id.vars = c("codice","gruppo"),measure.vars = c("conoscenze_t0"
  , "conoscenze_t1", "conoscenze_t2"))
names(melt_sic)<-c("sogg", "gruppo","tempo", "conoscenze")
```

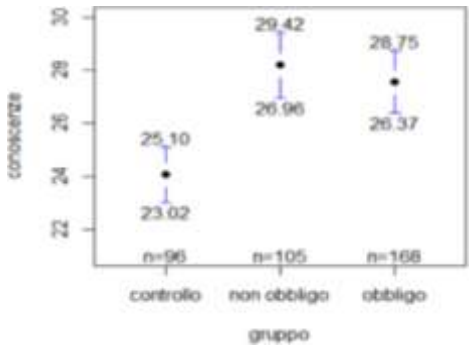
Rinominiamo le etichette dei gruppi, fastidiosamente lunghe, con levels(X1):

```
levels(melt_sic$gruppo)<- c("controllo", "non obbligo","obbligo")
levels(melt_sic$tempo)<- c("t0", "t1","t2")
```

```
 head(melt_sic,4)
    sogg   gruppo     tempo     conoscenze
  1 BC0Y29  obbligo      t0              34
  2 BG3M14  obbligo      t0              28
  3 BL9D18  obbligo      t0              32
  4 BR3E44  obbligo      t0              21
```

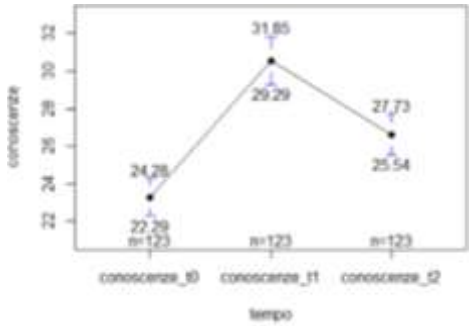
```
tail(melt_sic,4)
          sogg      gruppo   tempo     conoscenze
  366SL6B14  controllo      t2              28
  367SN3J09  controllo      t2              26
  368SP7U25  controllo      t2              20
  369TR3D18  controllo      t2              27
```

```
round(tapply(melt_sic$conoscenze, melt_sic$gruppo, mean),2)
   controllo  non obbligo  obbligo
       24.06           28.19    27.56
round(tapply(melt_sic$conoscenze, melt_sic $gruppo, sd),2)
   controllo  non obbligo  obbligo
        5.14         6.36     7.80
```



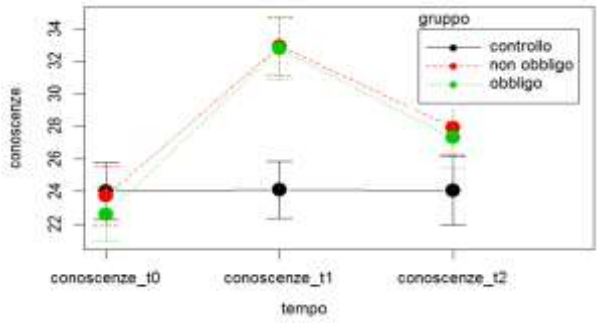
Indipendentemente dal momento della rilevazione, le conoscenze del gruppo di controllo sembrano inferiori a quelle degli altri due gruppi, tra loro molto simili.

```
round(tapply(melt_sic$conoscenze, melt_sic$tempo, mean),2)
    t0     t1    t2
 23.28  30.57  26.63
round(tapply(melt_sic$conoscenze, melt_sic$tempo, sd),2)
       t0     t1     t2
     5.55   7.16   6.15
```



Indipendentemente dal gruppo, le conoscenze T_0 sembrano nettamente inferiori a quelle rilevate a T_1 e T_2, ma queste ultime sembrano tracciare una spiacevole inversione di tendenza.

```
round(tapply(melt_sic$conoscenze,
   list(melt_sic$gruppo,ms$tempo), mean),2)
                  t0     t1     t2
controllo     24.00  24.12  24.06
obbligo       23.71  32.91  27.94
non obbligo   22.61  32.79  27.29

round(tapply(melt_sic$conoscenze, list(melt_sic
   $gruppo,ms$tempo), sd),2)
                t0    t1    t2
controllo     4.83  4.84  5.86
obbligo       5.27  5.41  4.81
non obbligo   6.09  7.05  6.69
```



La volontarietà della partecipazione al corso certamente non determina un diverso apprendimento; tuttavia, a T_0 i tre gruppi sono identici, a T_1 i controlli sono decisamente inferiori e a T_2, nonostante il brusco calo dei gruppi sperimentali, sembrano ancora abbastanza nettamente più bassi.

Applichiamo ezAnova, aggiungendo il predittore between e ricordandoci, dato che i gruppi sono sbilanciati, di scegliere un adeguato tipo di partizione della SS_M:

```
ezANOVA(data = melt_sic, dv = conoscenze, wid = sogg, within = tempo, between = gruppo, type= 3, detailed=FALSE)
```

```
$Mauchly Test for Sphericity

        Effect             W         p p<.05
3        tempo   0.9886695 0.5076245
3 gruppo:tempo   0.9886695 0.5076245
```

La sfericità per il fattore within e l'interazione non è un problema (quindi qui non visualizziamo le correzioni):

```
$ANOVA
        Effect DFn  DFd           F        p p<.05          ges
2       gruppo   1  120    5.897571 3.605044e-03    0.07549227
3        tempo   1  240  139.407380 6.666075e-41    0.16431512
4 gruppo:tempo   4  240   31.854570 2.726606e-21    0.08244827
```

Tutti gli effetti sono significativi, anche se il più forte è il tempo, indipendentemente dal gruppo. Usiamo l'analisi degli effetti semplici per approfondire i confronti d'interazione.

```
melt_sic$concatena<-as.factor(paste(melt_sic$gruppo, melt_sic$tempo))
pairwise.t.test(melt_sic$conoscenze,melt_sic$concatena,p.adjust.method="b")
```

```
data:  melt_sic$conoscenze and melt_sic$concatena

            controllo t0  controllo t1 controllo t2 non obbligo t0 non obbligo t1 non obbligo t2  obbligo t0  obbligo t1
controllo t1     1.00000             -            -              -              -              -           -           -
controllo t2     1.00000       1.00000            -              -              -              -           -           -
non obbligo t0   1.00000       1.00000      1.00000              -              -              -           -           -
non obbligo t1   5.9e-08        9.7e-08      7.6e-08        7.8e-09              -              -           -           -
non obbligo t2   0.23357       0.30148      0.26557        0.10248        0.01676              -           -           -
obbligo t0       1.00000       1.00000      1.00000        1.00000        2.6e-13        0.00118           -           -
obbligo t1       2.4e-09        4.2e-09      3.2e-09        1.7e-10        1.00000        0.00572    < 2e-16           -
obbligo t2       0.44007       0.57262      0.50248        0.18516        0.00044        1.00000     0.00119     4.3e-05
```

Teniamo fisso $X_{2\ Gruppo}$ (in rosso): i controlli sono sempre uguali a se stessi nel tempo; entrambi i gruppi sperimentali aumentano significativamente da T_0 a T_1 e calano significativamente da T_1 a T_2; tra T_0 e T_2, il gruppo con formazione obbligatoria mostra un calo significativo, quello senza obbligo no.

Teniamo fisso $X_{1\ Tempo}$ (in blu): a T_0 i tre gruppi non sono significativamente differenti; a T_1, i Controlli sono significativamente inferiori a entrambi i gruppi sperimentali, tra loro identici; a T_2, i tre gruppi tornano tristemente non significativamente differenti.

Concludiamo con un esempio di **interazione a tre vie**. Aggiungiamo ai precedenti un ulteriore predittore:

- $X_{1\ Tempo}$, misure ripetute: indipendentemente dal gruppo di appartenenza, si è verificato un cambiamento significativo nelle **conoscenze** ($\boldsymbol{Y}$) da T_0 a T_2;
- $X_2\ Gruppo$ (between groups): indipendentemente dal momento in cui sono rilevate, esiste una differenza nelle conoscenze dei tre gruppi (formazione obbligatoria, non obbligatoria, nessuna formazione);
- $\boldsymbol{X_{3\ Formazione}}$ (between groups - $formatori): indipendentemente dal momento in cui sono rilevate e dal gruppo, esiste una differenza nelle conoscenze a seconda che il soggetto sia stato precedentemente formato per essere un *peer tutor* alla sicurezza (X_{3a} formatore, X_{3b} non formatore).

```
table(sicurezza$formatori)
    formatori non formatori
           81            42
```

L'aggiunta di un predittore fa sì che i termini di **interazione** siano **quattro**, tre a due vie e una a tre vie:

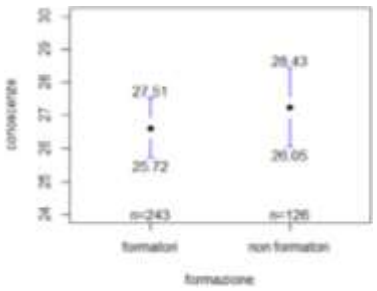
- $X_{1Tempo} \times X_{2Gruppo}$: la variazione nel tempo delle conoscenze è diversa a seconda del gruppo di appartenenza;
- $X_{1Tempo} \times X_{3Formazione}$: la variazione nel tempo delle conoscenze è diversa a seconda che il soggetto sia un *peer tutor* o che non lo sia;
- $X_{2Gruppo} \times X_{3Formazione}$: la diversa performance nel compito di conoscenza nei tre gruppi è differente a seconda che il soggetto sia un *peer tutor* o che non lo sia;
- $X_{1Tempo} \times X_{2Gruppo} \times X_{3Formazione}$: la diversità nel cambiamento delle conoscenze nel tempo (X_1), rilevata a seconda del gruppo di appartenenza (X_2), risente del fatto che il soggetto sia stato precedentemente addestrato come formatore alla sicurezza (X_3).

Aggiungiamo il predittore che manca al dataframe precedente, tra le variabili **uniche**:

```
melt_sic<-melt(data = sicurezza, id.vars = c("codice","gruppo", "formatori"), measure.vars = c("
  conoscenze_t0", "conoscenze_t1", "conoscenze_t2"))
names(melt_sic)<-c("sogg", "gruppo", "formazione", "tempo", "conoscenze")
levels(melt_sic$gruppo)<- c("controllo", "non obbligo","obbligo")
levels(melt_sic$tempo)<- c("t0", "t1","t2")

head(melt_sic)
    sogg  gruppo      formazione  tempo  conoscenze
1 BC0Y29 obbligo       formatori     t0          34
2 BG3M14 obbligo   non formatori     t0          28
3 BL9D18 obbligo       formatori     t0          32
4 BR3E44 obbligo   non formatori     t0          21
5 BR3P27 obbligo       formatori     t0           7
6 BR3S10 obbligo       formatori     t0          15
```

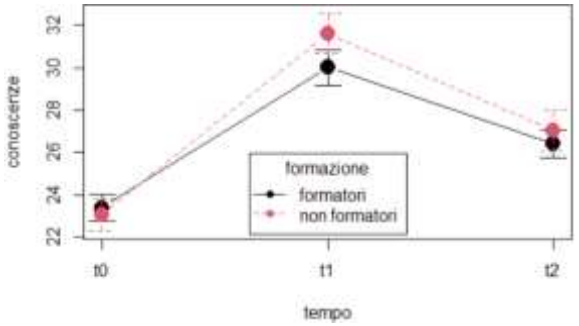
Descriviamo solo gli effetti **nuovi** rispetto all'esempio precedente, cioè l'effetto principale della formazione, le interazioni Tempo*Formazione e Gruppo*Formazione, più l'interazione a tre vie:

```
tapply(melt_sic$conoscenze,melt_sic$formazione,mean)
    formatori non formatori
     26.61728      27.23810
tapply(melt_sic$conoscenze,melt_sic$formazione,sd)
    formatori non formatori
     7.093067      6.742615
```



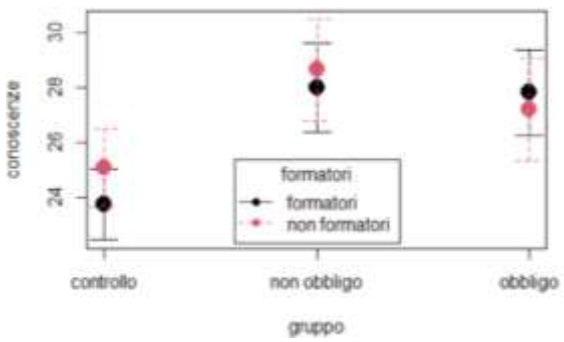
Indipendentemente da tempo e gruppo, le conoscenze di formatori e non formatori sono molto simili.

```
tapply(melt_sic$conoscenze, list(melt_sic$tempo,
melt_sic$formazione), mean)
   formatori non formatori
t0  23.39506      23.07143
t1  30.02469      31.61905
t2  26.43210      27.02381
tapply(melt_sic$conoscenze, list(melt_sic$tempo,
melt_sic$formazione), sd)
   formatori non formatori
t0  5.915402      4.825708
t1  7.617374      6.132490
t2  6.072352      6.341840
```



Indipendentemente dal gruppo, la variazione nel tempo di formatori e non formatori è molto simile.

```
tapply(melt_sic$conoscenze, list(melt_sic$gruppo,
   melt_sic$formazione), mean)
            formatori non formatori
controllo    23.77333      25.09524
non obbligo  27.98611      28.63636
obbligo      27.81250      27.22222
tapply(melt_sic$conoscenze, list(melt_sic$gruppo,
   melt_sic$formazione), sd)
            formatori non formatori
controllo    5.561871      3.112953
non obbligo  6.867956      5.164895
obbligo      7.717802      7.938535
```

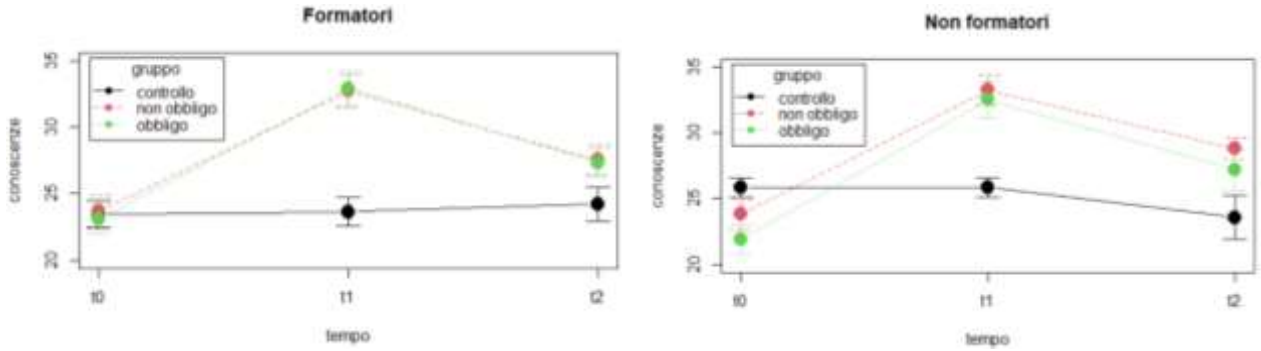


Indipendentemente dal tempo, la differenza formatori e non formatori è molto simile nei tre gruppi (con la parziale eccezione del gruppo di controllo.

Per descrivere l'interazione a tre vie, il modo più conveniente è fare **due grafici d'interazione a due vie $X_1 \times X_2$ per ogni livello di X_3**, ovvero un grafico $\boldsymbol{Tempo} \times \boldsymbol{Gruppo}$ per i **formatori** e uno per i **non formatori**. Se i due grafici d'interazione sono uguali nei due livelli di X_3, allora probabilmente l'essere un *peer tutor* non ha un effetto sull'interazione, e quindi l'interazione a tre vie non risulterà significativa. Se invece i due grafici di interazione a due vie **sono diversi nei due livelli di X_3,** allora probabilmente l'essere stati formati per il *peer tutoring* ha un diverso impatto sull'interazione fra tempo e gruppo, quindi **l'interazione a tre vie $Tempo \times Gruppo \times Formazione$ potrebbe essere significativa**.

Creiamo i subset per fare i due grafici d'interazione:

```
formatori<-subset(melt_sic,melt_sic$formazione=="formatori")
non_formatori<-subset(melt_sic,melt_sic$formazione=="non formatori")
```



L'interazione $Tempo \times Gruppo$ nei formatori è decisamente simile a quella rilevata tra i non formatori: probabilmente, non troveremo una interazione a tre vie significativa.

Verifichiamo:

```
ezANOVA(data = melt_sic, dv = conoscenze, wid = sogg, within = tempo, between = .(gruppo,formazi
one), type= 3)
```

```
$Mauchly Test for Sphericity
                   Effect         W   p p<.05
4                   tempo   0.99153  0.610576
5            gruppo:tempo   0.99153  0.610576
6       formazione:tempo    0.99153  0.610576
7 gruppo:formazione:tempo   0.99153  0.610576
```

La sfericità per il fattore within e le interazioni non è un problema (quindi non sono qui visualizzate le correzioni):

```
$ANOVA
                   Effect DFn  DFd           F        p p<.05         ges
2                  gruppo   2  117    3.5603561  3.154887e-02   0.0482524125
3              formazione   1  117    0.1673286  6.832448e-01   0.0011899452
4                   tempo   2  234  112.5125340  5.799574e-35   0.1383527525
5       gruppo:formazione   2  117    0.2877811  7.504550e-01   0.0040812196
6            gruppo:tempo   4  234   24.0529278  1.108445e-16   0.0642419767
```

```
7         formazione:tempo   2  234    0.2369659  7.892073e-01    0.0003380621
8 gruppo:formazione:tempo    4  234    1.1794344  3.205309e-01    0.0033550704
```

L'effetto di $X_{1\,Tempo}$ è decisamente il più forte tra quelli significativi ($X_{2\,Gruppo}$ e interazione $X_{1\,Tempo} \times X_{2\,Gruppo}$). $X_{3\,Formazione}$ ha alcun effetto, né indipendentemente dagli altri predittori, né nelle interazioni a due e a tre vie.

13.3 L'analisi della covarianza: ANCOVA

L'analisi della **covarianza (ANCOVA)** rappresenta l'estensione dell'ANOVA a **una o più covariate**, cioè variabili X, che hanno una relazione con Y, ovvero esercitano un effetto su Y, ma non fanno parte dell'ipotesi sperimentale: la loro influenza, non controllata dal disegno sperimentale, può **incidere sull'effetto che la (o le) variabili indipendenti X** hanno su Y. Per questo motivo sono infatti definite anche **variabili di disturbo**. l'ANCOVA "tradizionale" usa covariate **continue**, ma possono essere anche **categoriali**: in questo caso sono definite **variabili di blocco** (***blocking variables***), e, se hanno più di due livelli, vengono dummizzate come accade per un qualsiasi modello lineare. Indipendentemente dalla loro natura, quando le covariate sono previste dal ricercatore, e quindi misurate, il loro effetto sulla relazione $Y \sim X$ può essere inserito nel modello lineare e **parzializzato**, cioè escluso: in questo modo, la **relazione $Y \sim X$ viene stimata al netto dell'effetto della relazione tra la covariata (o le covariate) e Y**.

Avremo allora un modello lineare **additivo**, in cui il valore $i-esimo$ di Y è dato da intercetta, effetto della covariata ed effetto di X, più l'errore:

$$y_i = b_0 + b_1 cov_i + b_1 X_i + e_i$$

L'inclusione della covariata nel modello lineare fa sì che si compia una **sorta di regressione gerarchica a blocchi**. Nel **primo** blocco, Y viene fatta regredire sulla covariata[105], misurando quindi la **relazione tra la covariata e Y:** se questa relazione esiste, la covariata spiegherà una parte più o meno consistente della varianza di Y. Nel blocco **successivo**, viene condotta **l'ANOVA** vera e propria (*between groups*, a misure ripetute, mista), ovvero si **misura la relazione tra la/le X e Y**, calcolando **quanta della varianza di Y non predetta dalla covariata** è **attribuibile all'effetto di X.** Perciò, se la covariata inserita al primo blocco spiegasse il 20% della variabilità di Y, i fattori inseriti nel blocco successivo dovrebbero ripartire tra sé e con l'errore il restante 80% della variabilità di Y. Ne consegue che se la covariata ha un forte effetto su Y, alla/alle X resterà poco da spiegare, e probabilmente il loro effetto non risulterà significativo: la differenza il risultato di un'ANOVA e quello di un'ANCOVA, condotte sugli stessi dati, potrebbe essere rilevante. Se invece la covariata ha uno scarso effetto su Y, ovvero se ne spiega poca varianza, ai fattori X resterà molto da spiegare: la differenza tra i risultati di ANOVA e ANCOVA sugli stessi dati sarà probabilmente trascurabile.

Il principale vantaggio di ANCOVA su ANOVA, quindi, è che la proporzione di MS_R di Y confrontata con quella spiegata da X è diminuita, dato che parte della MS_R di Y è precedentemente attribuita alla covariata: **la variabilità residua è ridotta**, consentendo di valutare meglio quella sperimentale (MS_M).

Tuttavia, ANCOVA esige che siano rispettati due requisiti affinché il modello sia affidabile:

1. **indipendenza tra la covariata e predittori X**: se la covariata è **continua, non deve essere differente tra i livelli di X**: **t-test per campioni indipendenti o ANOVA eseguiti sulla covariata come Y e la variabile indipendente X come predittore ($covariata \sim X$) non dovranno essere significativi**. Se la covariata è **categoriale**, **non deve essere significativamente associata a X**. Quando il requisito è soddisfatto, la varianza di Y spiegata dalla covariata è solo "rumore", cioè errore, dal punto di vista di X: eliminato il rumore nel primo blocco del modello, la relazione tra Y e X potrà delinearsi più chiaramente. **Attenzione**: **NON si deve usare l'ANCOVA per "controllare" differenze tra gruppi non bilanciati proprio nella covariata**. Se, per esempio, avessimo costituito grupppi sperimentali secondo fasce di ritardo intellettivo ($X_1: lieve, X_2: moderato, X_3: grave$)

[105] Per non appesantire il discorso, si fa genericamente riferimento a **una** covariata, ma il modello può contenere **più covariate**, che in ogni caso sono inserite nel primo blocco.

e scoprissimo che questi gruppi non sono bilanciati per età, non potremmo inserire l'età come covariata, come se questo cancellasse lo sbilanciamento (Cohen e Cohen, 1975; Miller e Chapman, 2001).

2. **omogeneità dei coefficienti angolari**: in ANCOVA, nel primo blocco si adatta la retta di regressione $Y \sim covariata$ **all'intero campione**, indipendentemente dal gruppo cui appartiene un soggetto. Ne deriva che, affinché il modello di regressione sia valido, la **relazione tra covariata e Y deve essere la stessa in tutti i gruppi corrispondenti ai livelli di X** → i b_1 che descrivono questa relazione devono essere gli stessi in tutti i livelli di X: se così non fosse, il modello ricavato nel primo blocco sarebbe invalido e la successiva ANOVA inaffidabile. L'uguaglianza dei b_1 di $Y \sim covariata$ nei livelli di X si traduce in una **interazione non significativa tra covariata e X**, che indica come l'effetto della covariata sia lo stesso per tutti i livelli di X. **Attenzione**: anche se statisticamente disgraziata, una significativa interazione $covariata \times X$ può essere interpretativamente interessante (ad esempio, può essere interessate rilevare che diverse dosi di un farmaco (X) hanno un diverso effetto a seconda dell'età o del peso del paziente): solo, per fare l'analisi non useremmo ANCOVA, ma un **multivel / mixed linear model** (che, peraltro, è preferito ad ANCOVA anche se il requisito di omogeneità è rispettato, soprattutto nel caso di ANCOVA su disegni within o misti).

Vediamo il requisito di omogeneità con un esempio su dati inventati. Ipotizziamo che la **performance** in un compito di calcolo a memoria (Y = da 0 a 10) sia positivamente influenzata dall'entità della **ricompensa** ricevuta per la loro partecipazione all'esperimento (X = di piccola, media e grande entità), ma supponiamo anche che la **motivazione** intrinseca al compito, autoriferita dai soggetti ($covariata$ = da 0 a 10), possa in qualche modo interferire sulla relazione tra performance e ricompensa. I dati sono:

```
performance<- c(2,2,3,4,5,7,3,4,5,5,7,7,2,2,3,4,4,4,5,6,6,6,8,9)
ricompensa<-c(rep("piccola",6), rep("media",6), rep("grande",12))
motivazione<-c(2,4,5,5,6,7,6,7,8,9,10, 10, 0,3,4, 1,2,3,0,0,3,5,1,2)
d<-data.frame(performance, ricompensa, motivazione)
```

```
d[1:6,]
  performance ricompensa motivazione
1           2    piccola           2
2           2    piccola           4
3           3    piccola           5
4           4    piccola           5
5           5    piccola           6
6           7    piccola           7
```

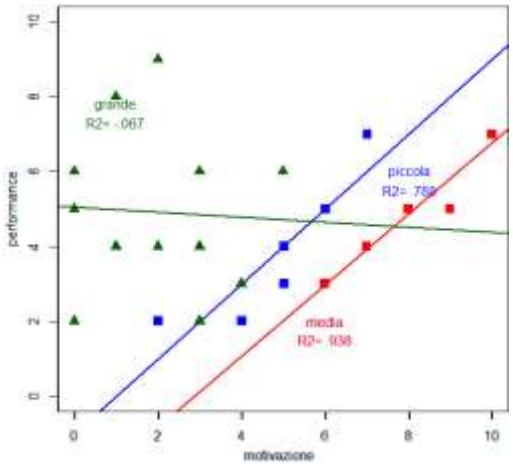
```
d[7:12,]
   performance ricompensa motivazione
7            3     media           6
8            4     media           7
9            5     media           8
10           5     media           9
11           7     media          10
12           7     media          10
```

```
d[13:18,]
   performance ricompensa motivazione
19           5    grande           0
20           6    grande           0
21           6    grande           3
22           6    grande           5
23           8    grande           1
24           9    grande           2
```

Creiamo i tre modelli della relazione $Y_{performance} \sim covariata_{motivazione}$ corrispondenti ai tre livelli di X ricompensa (usiamo l'argomento `subset` in `lm`) e vediamone i b_1:

```
piccola<-lm(d$performance~d$motivazione, subset = d$ricompensa=="piccola")
media<-lm(d$performance~d$motivazione, subset = d$ricompensa=="media")
grande<-lm(d$performance~d$motivazione, subset = d$ricompensa=="grande")
piccola[1];media[1];grande[1]
```

```
$coefficients
  (Intercept) d$motivazione
           -1             1
$coefficients
  (Intercept) d$motivazione
        -2.75          0.95
$coefficients
  (Intercept) d$motivazione
   5.05000000   -0.06666667
```

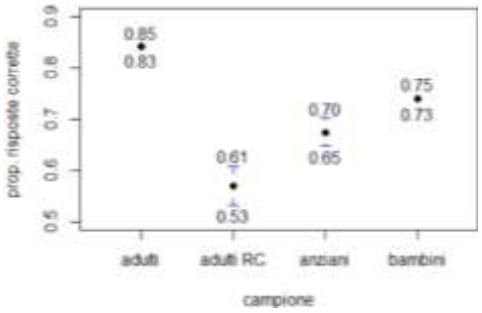


I b_1 dei gruppi che hanno ricevuto ricompense di scarsa e media entità sono molto simili tra loro: in entrambi i gruppi la motivazione intrinseca ha un forte potere predittivo sulla performance. Invece, nel gruppo che ha ricevuto una forte ricompensa la relazione tra motivazione intrinseca e performance viene praticamente azzerata

Per vedere l'applicazione di un'ANCOVA su dati veri usiamo il dataframe **denominazione**: sono le risposte corrette (espresse come proporzione su un totale di 400 stimoli) date in un **compito di denominazione** di stimoli visivi da **quattro diversi campioni**: **adulti** senza deterioramento cognitivo (poiché i dati servivano alla standardizzazione del set di stimoli a uso neuropsicologico, questo gruppo rappresenta il riferimento normativo), **anziani** senza deterioramento cognitivo, **bambini** a sviluppo tipico (in età prescolare e scolare), **adulti con ritardo cognitivo**. Le figure sono categorizzabili in vario modo: nel dataframe sono presenti solo alcune delle categorie. Naturalmente, ci aspettiamo che tra i gruppi ci siano differenze nelle capacità di denominazione corretta, ma anche che entro ciascun gruppo alcune **variabili non controllate dallo sperimentatore** (**genere**, **età**, **titolo di studio**) possano incidere sulla prestazione.
Usiamo la **proporzione sul totale** di denominazioni corrette e, con ANOVA, vediamo se si rilevano differenze tra i gruppi nella capacità di denominare correttamente gli stimoli. Rinominiamo il dataframe come **den**:

```
den <- denominazione
```

```
Desc(den$prop_totale~den$campione, digits = 2)
den$prop_totale ~ den$campione
Summary:
n pairs:557,valid:557 (100.0%), missings: 0 (0.0%), groups: 4
           adulti  adulti RC    anziani    bambini
mean         0.84       0.57       0.67       0.74
median       0.86       0.60       0.66       0.75
sd           0.06       0.17       0.09       0.08
IQR          0.07       0.22       0.10       0.11
n             199         76         45        237
np         35.73%     13.64%      8.08%     42.55%
```



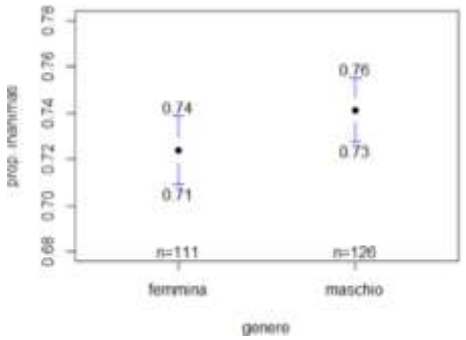

```
summary(aov(den$prop_totale ~ deno$campione))
              Df Sum Sq Mean Sq F value Pr(>F)
den$campione   3  4.417  1.4723   180.1 <2e-16
Residuals    553  4.521  0.0082
```

Come da ipotesi, l'appartenenza al gruppo incide in maniera significativa sulla capacità di denominazione.

Calcolate la proporzione di varianza spiegata da X; fate i contrasti semplici usando il gruppo degli adulti come riferimento; fate i test post hoc.

Concentriamoci solo sui 237 **bambini**. È un dato abbastanza noto, in letteratura, che le **competenze linguistiche delle bambine siano mediamente migliori delle competenze dei coetanei maschi**: questo vale anche per il compito di denominazione? Vediamo cosa succede per la categoria degli **oggetti inanimati**.

```
bimbi<-subset(den,  den$campione=="bambini")
round(tapply(bimbi$prop_inanimati, bimbi$genere, mean),2)
femmina maschio
   0.724    0.741
round(tapply(bimbi$prop_inanimati, bimbi$genere, sd),2)
femmina maschio
   0.08    0.08
summary(lm(bimbi$prop_inanimati~bimbi$genere))
[...]
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.723874   0.007433  97.392   <2e-16
bimbi$generemaschio 0.017396   0.010194   1.707   0.0892
[…]
Multiple R-squared: 0.01224,Adjusted R-squared: 0.008038
F-statistic: 2.912 on 1 and 235 DF,  p-value: 0.08923
```



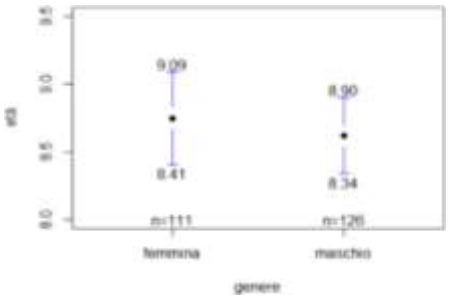
Secondo ANOVA, **no**: la minima differenza tra le medie non è significativa, nonostante la numerosità del campione renda il test molto potente, e la varianza spiegata dal genere è poco più dell'1%. **Questo potrebbe risentire dell'età dei bambini**? Forse le differenze di genere sono più marcate quanto più l'età è ridotta, e tendono ad annullarsi con la crescita. Verifichiamo quindi la **relazione tra X_{Genere} e $Y_{Denominaizone}$**, una volta **tolto l'effetto della relazione tra $covariata_{Età}$ e $Y_{denominazione}$ di oggetti inanimati**.

```
summary(bimbi$eta)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.000   8.000   8.000   8.679  10.000  12.000
cor(bimbi$eta, bimbi$prop_inanimati)
[1] 0.307014
```

In effetti, **la relazione tra età e denominazione di oggetti inanimati esiste**, ed è positiva. Quindi l'inserimento dell'età come covariata nel modello è sensato.

Verifichiamo gli assunti: c'è una **relazione tra la $covariata_{Età}$ e X_{Genere}**? Maschi e femmine **non** dovrebbero avere età significativamente diverse:

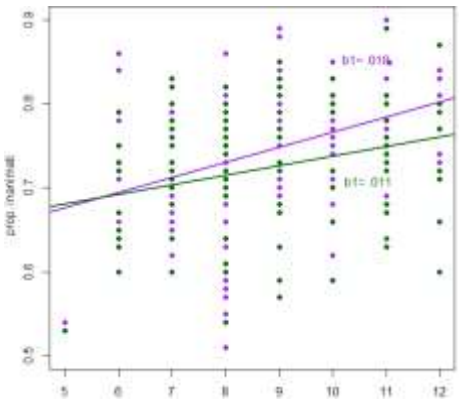
```
summary(aov(bimbi$eta~bimbi$genere))
              Df Sum Sq Mean Sq F value Pr(>F)
bimbi$genere   1    1.0  0.9775   0.338  0.561
Residuals    235  678.7  2.8879
```



Il primo assunto è verificato: la covariata Età non è significativamente differente nei diversi livelli di X.

Rappresentiamo il secondo assunto di **omogeneità dei b_1** della relazione $Y_{denominazione} \sim covariata_{Età}$ in ciascun livello di X_{Genere}.

```
maschi<-lm(bimbi_M$prop_inanimati~bimbi_M$eta,
  subset=bimbi$genere=="maschio")
femmine<-lm(bimbi_F$prop_inanimati~bimbi_F$eta,
  subset=bimbi$genere=="femmina")

maschi$coefficients[2]; femmine$coefficients[2]
bimbi_M$eta
 0.01796229
bimbi_F$eta
 0.01137517
```



I b_1 sono entrambi positivi: sappiamo che i b_1 saranno significativamente differenti se **l'interazione $covariata \times X$ risulterà significativa**.

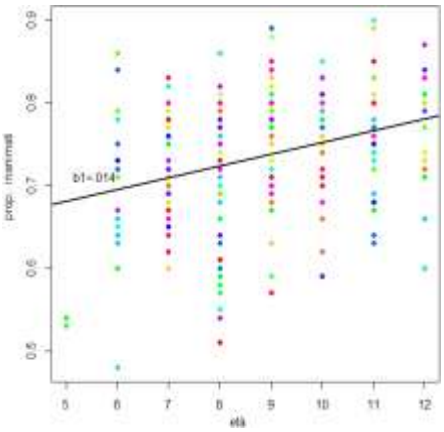
Poiché il **modo più veloce** per verificarlo è aggiornare il modello ANCOVA **additivo** $Y \sim covariata + X$ che useremo per l'ipotesi sperimentale, prima facciamo ANCOVA, poi attribuiamo un $p - value$ all'interazione $covariata \times X$.

L'analisi della covarianza in R è molto semplice: ci servirà, fondamentalmente, **aov**, in cui inseriremo la covariata come primo fattore della formula, cui si aggiungono i predittori: **aov(Y ~ covariata + X)**. Sappiamo già (§7.2) che **aov** usa la **partizione della SS_M di Tipo I**, e quindi attribuirà tutta la variabilità di Y possibile al primo fattore inserito nel modello, sia quella unica sia quella condivisa: la covariata prenderà la fetta più grossa che può spiegare, mentre il fattore X inserito al secondo passo cercherà di spiegare la restante variabilità di Y.

```
covariata<-aov(prop_inanimati~ eta+genere, data= bimbi)
summary(covariata)
             Df  Sum Sq  Mean Sq  F value   Pr(>F)
eta           1  0.1375  0.13751   24.761 1.26e-06
genere        1  0.0218  0.02185    3.934   0.0485
Residuals   234  1.2995  0.00555
```

Tolto l'effetto (significativo) della covariata età, la relazione tra genere e capacità di denominazione degli oggetti inanimati è ora significativa (per quanto resti debole). Per conoscere il coefficiente angolare della covariata e il suo segno (nonché i contrasti del predittore X, se è a più di due livelli), possiamo applicare **`summary.lm(modello)`**, che riporta il modello alla sua "originaria" struttura `lm`:

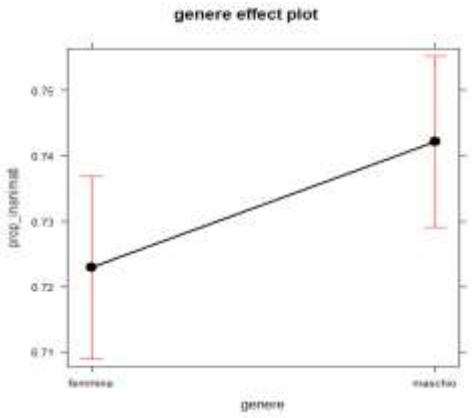
```
summary.lm(covariata)
Call:
aov(formula = prop_inanimati ~ eta + genere, data = bimbi)
Residuals:
      Min        1Q    Median        3Q       Max
-0.222331 -0.047516  0.006548  0.054350  0.156548

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.597560   0.026005  22.979  < 2e-16
eta               0.014440   0.002861   5.048 8.99e-07
genere[T.maschio] 0.019254   0.009708   1.983   0.0485
```



La **relazione** tra covariata Età e proporzione di denominazioni corrette di oggetti inanimati è – come ragionevole – **positiva**: al crescere dell'età, cresce la proporzione di risposte corrette (.01 per ogni anno in più). La relazione si vede bene nel `plot(covariata, X)`.

Possiamo ora descrivere le **medie marginali corrette**, ovvero le **medie dei livelli di X stimate dopo aver tolto l'effetto della covariata**, usando `effect("predittore", modello)` del package `effects`: la funzione crea un oggetto per uno dei termini (di solito un effetto principale) di un modello lineare (o lineare generalizzato, cap. 8), che assorbe i marginali dei termini inferiori del modello e fa la media per tutti gli altri termini del modello. Aggiungendo l'argomento `se= TRUE`, si può anche avere la stima dello SE e dei CI delle medie marginali. Inoltre, con `plot(medie_stimate)`, si ottiene il grafico delle medie stimate con relative barre di errore:

```
medie_stimate<-effect("genere", covariata, se=TRUE)
summary(medie_stimate)
 genre effect
 genere
  femmina    maschio
0.7228859  0.7421402
 Lower 95 Percent Confidence Limits
genere
  femmina   maschio
0.708945  0.729056
 Upper 95 Percent Confidence Limits
genere
   femmina    maschio
0.7368268  0.7552244
plot(medie_stimate)
```



Ricordiamo le medie che avevamo calcolato precedentemente:

```
round(tapply(bimbi$prop_inanimati, bimbi$genere, mean),2)
femmina maschio
  0.724   0.741
```

Per costruire il modello con interazione $covariata \times X$ e verificare il secondo assunto, ricordiamo **`update(oggetto da aggiornare, .~. + nuovo elemento)`** che abbiamo visto nel capitolo 5: aggiorniamo il vecchio modello specificando che **va lasciato tutto uguale tranne** (`.~.`) l'aggiunta (`+`) del nuovo elemento che modifica il modello, cioè il termine di **interazione** $X_1:X_2$.

```
covariata_interazione<-update(covariata, .~.+ genere:eta)
summary(covariata_interazione)
             Df  Sum Sq  Mean Sq  F value   Pr(>F)
eta           1  0.1375  0.13751   24.795 1.24e-06
genere        1  0.0218  0.02185    3.939   0.0483
eta:genere    1  0.0073  0.00733    1.321   0.2516
Residuals   233  1.2922  0.00555
```

L'interazione $covariata$ ×non è significativa, il secondo assunto è rispettato.

Se la nostra X avesse più di due livelli e il suo effetto fosse significativo, potremmo decidere di fare post hoc tra i livelli: tuttavia, `pairwise.t.test` o le altre funzioni dedicate viste nel §7.1.2 non potrebbero essere applicate alle medie marginali corrette. Potremo allora usare **`glht`**`(modello, linfct=mcp(predittore="contrasti"))` del package **`multcomp`**, che crea un oggetto basato sul modello `aov` o `lm` (e, come vedremo nel capitolo 9, su altri tipi di modello). L'argomento `linfct=mcp()` specifica il tipo di contrasti: per fare confronti a coppie fra tutti i livelli, secondo la logica dei post hoc, sono utilizzati i contrasti di Tukey. Per visualizzare i **post hoc** del predittore specificato, useremo `summary(oggetto glht)`, in cui è possibile specificare il tipo di correzione per il family-wise error aggiungendo l'argomento `test=adjusted(type= "metodo di correzione"))`; per la correzione possiamo scegliere fra `none`, `bonferroni`, `BH` (Benjamini-Hochberg), `holm` (e diversi altri, qui non trattati). Per visualizzare i CI dei contrasti, si usa l'usuale `confint(oggetto glht)`.

Ne vediamo un esempio con il dataframe **epde**: verifichiamo l'esistenza di differenze significative nella soddisfazione per l'esperienza sessuale di coppia (totale IIEF) a seconda della frequenza mensile dei rapporti, una volta escluso l'effetto dell'ansia di tratto. Per maggiore chiarezza, soprattutto nei grafici, riordiniamo i livelli del fattore:

```
epde$attivita<-ordered(epde$attivita_sessuale_mese, levels=c("1-2 volte","3-4 volte","5-6
volte",">=7 volte"))
Desc(epde$IIEF~epde$attivita, digits = 2)
-------------------------------------------------------------------
        1-2 volte  3-4 volte  5-6 volte  >=7 volte
mean        28.52      36.49      52.09      56.40
median      24.00      37.00      53.00      53.50
sd          13.28      14.38       7.46       5.82
```

Usiamo `Etasq(modello, anova=TRUE, type=1)` invece di `summary(aov)`, per meglio evidenziare la variazione di X; prima ANOVA:

```
EtaSq(aov(epde$IIEF~epde$attivita),type = 1, anova = TRUE)
                 eta.sq eta.sq.part       SS df        MS         F            p
epde$attivita 0.3901673   0.3901673  8437.47  3 2812.4902  17.70097 5.688819e-09
Residuals     0.6098327          NA 13187.79 83  158.8891        NA           NA
```

La soddisfazione è significativamente differente nelle diverse classi di frequenza; X spiega il 39.1% di variabilità, e la devianza del modello è quasi diciotto volte maggiore della devianza residua. Costruiamo il modello ANCOVA:

```
ancova_attivita<-aov(IIEF~STAI_tratto+attivita, data=epde)
EtaSq(ancova_attivita,type = 1, anova = TRUE)
                eta.sq eta.sq.part         SS df          MS          F           p
STAI_tratto 0.81069893   0.8370187 17531.5786  1 17531.57861 421.125194 0.000000000
attivita    0.03144466   0.1661093   679.9991  3   226.66638   5.444742 0.001829916
Residuals   0.15785641          NA  3413.6866 82    41.63032         NA          NA
```

La soddisfazione è ancora significativamente differente nelle diverse classi di frequenza, ma l'effetto di X è drasticamente calato: in effetti, l'effetto della covariata, inserito al primo step, si prende ben l'81% della variabilità di Y. È probabile trovare, quindi, una differenza tra le medie dei livelli corrette per l'effetto della covariata e quelle non corrette:

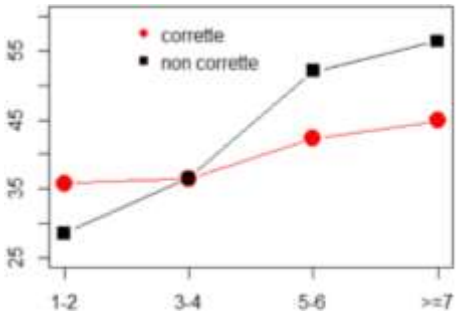
```
marginali_attivita<-effect(term = "attivita",mod = ancova_attivita, se=TRUE)
summary(marginali_attivita)
attivita effect
attivita
   1-2 volte 3-4 volte 5-6 volte >=7 volte
    35.78323  36.44903  42.23841  44.83813
Lower 95 Percent Confidence Limits
attivita
   1-2 volte 3-4 volte 5-6 volte >=7 volte
    33.29233  34.27945  38.16247  40.51055
 Upper 95 Percent Confidence Limits
attivita
   1-2 volte 3-4 volte 5-6 volte >=7 volte
    38.27413  38.61862  46.31434  49.16571
```

Potremmo sovrapporre in un grafico le medie corrette e quelle non corrette, per enfatizzarne le differenze. Le medie marginali corrette sono uno dei termini della lista `marginali_attivita`: `marginali_attivita$fit`

```
non_corrette<-tapply(epde$IIEF, epde$attivita, mean)
corrette<-marginali_attivita$fit
```

Abbiamo già usato `lines` per sovrapporre un grafico a un altro; aggiungiamo la nuova funzione `legend(x, y, legend="testo", pch=simbolo, bty="n")` per sovrascrivere anche la legenda: `bty="n"` cancella il bordo esterno (altrimenti: `bty="o"`). aggiungiamo qualche altro sfizio grafico: cancelliamo l'asse X di default (`xaxt= "n"`), che presenterebbe una fittizia distribuzione continua, e con `ann=FALSE` eliminiamo anche le etichette; sostituiamo X con uno di nostra creazione con `Axis(side=1, at=c(posizioni in X), labels=c("etichette asse")`. In entrambi i grafici uniamo i punti mediante linee (`lty="b"`) per evidenziare come cambi la linearità della relazione $Y \sim X$ dopo l'intervento della covariata:

```
plot(corrette, pch=19, cex=2, ylim=c(25,60), ann = FALSE,
 ylab="IIEF",type="b",xaxt="n", col="red")
Axis(side=1, at= c(1,2,3,4), labels = c("1-2", "3-4", "5-
 6",">=7"))
lines(non_corrette, pch=15, type = "b", cex=1.5)
legend(x = 1.5, y = 60, legend = "corrette", pch = 19,
 col="red",bty = "n")
legend(x = 1.5, y = 56, legend = "non corrette", pch = 15,
 bty="n")
```



Le medie marginali corrette, con l'eccezione della categoria "3-4 volte", sono piuttosto diverse da quelle non corrette, e la linearità della relazione, parzializzato l'effetto della covariata, è meno marcata.

Poiché l'effetto del predittore è significativo, vediamo i post hoc, applicando la correzione di Bonferroni:

```
posthoc_attivita<-glht(ancova_attivita, linfct=mcp(attivita= "Tukey"))
summary(posthoc_attivita, test = adjusted(type="bonferroni"))
       Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: Tukey Contrasts
Fit: aov(formula = IIEF ~ STAI_tratto + attivita, data = epde)
Linear Hypotheses:
                          Estimate Std. Error t value Pr(>|t|)
3-4 volte - 1-2 volte == 0   0.6658     1.6612   0.401  1.00000
5-6 volte - 1-2 volte == 0   6.4552     2.5250   2.556  0.07451
>=7 volte - 1-2 volte == 0   9.0549     2.6488   3.419  0.00590
5-6 volte - 3-4 volte == 0   5.7894     2.3204   2.495  0.08762
>=7 volte - 3-4 volte == 0   8.3891     2.4327   3.448  0.00536
>=7 volte - 5-6 volte == 0   2.5997     2.8214   0.921  1.00000

(Adjusted p values reported -- bonferroni method)
```

Nonostante la severità della correzione, quasi tutti i confronti a coppie sono significativi o appena sopra soglia; naturalmente, le significatività delle differenze tra le medie da cui non è stato parzializzato l'effetto della covariata, pur con la stessa correzione di Bonferroni, sono più evidenti:

```
pairwise.t.test(epde$IIEF, epde$attivita, p.adjust.method = "b")
      Pairwise comparisons using t tests with pooled SD
data:  epde$IIEF and epde$attivita
          1-2 volte 3-4 volte 5-6 volte
3-4 volte 0.07299   -         -
5-6 volte 5.0e-06   0.00345   -
>=7 volte 2.1e-07   0.00019   1.00000

P value adjustment method: bonferroni
```

Se, invece di fare post hoc, si vogliono applicare contrasti a priori in ANCOVA, bisogna costruire la matrice dei contrasti desiderati e inserirla come oggetto dell'argomento `mcp`: `contrasti<-glht(modello, linfct = mcp(fattore =`

`matrice dei contrasti))`. Ad esempio, confrontiamo le frequenze più rare (accorpiamo 1-2 volte con 3-4 volte): con 5-6 volte e ≥7 accorpati, con 5-6 volte ignorando ≥7, con ≥7 ignorando 5-6 volte.

```
contrasti<-rbind("rari versus frequenti"=c(1,1,-1,-1),"rari versus 5-6"= c(1,1,-2,0), "rari versus
 >=7"=c(1,1,0,-2))
contrasti_attivita<-glht(ancova_attivita, linfct=mcp(attivita=contrasti))
summary(contrasti_attivita)
```

```
       Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: User-defined Contrasts
Fit: aov(formula = IIEF ~ STAI_tratto + attivita, data = epde)
Linear Hypotheses:
                          Estimate Std. Error t value Pr(>|t|)
rari versus frequenti == 0  -14.844      3.738  -3.971  < 0.001
rari versus 5-6 == 0        -12.245      4.556  -2.687  0.02031
rari versus >=7 == 0        -17.444      4.807  -3.629  0.00131
(Adjusted p values reported -- single-step method)
```

Le differenze pianificate sono tutte significative – ma, non essendo questi contrasti ortogonali, attenzione al family-wise error rate.

Concludiamo: avremo rispettato i requisiti di applicabilità per questo modello? Purtroppo, solo uno: l'interazione $covariata \times X$ non è significativa, ma la covariata STAI è significativamente differente nei livelli di X:

```
summary(aov(epde$STAI_tratto~epde$attivita))
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)
epde$attivita  3   6260  2086.5   11.44 2.36e-06
Residuals     83  15137   182.4
```

```
summary(aov(epde$IIEF~epde$STAI_tratto*epde$attivita))
```

```
                              Df Sum Sq Mean Sq F value  Pr(>F)
epde$STAI_tratto               1  17532   17532 417.247 < 2e-16
epde$attivita                  3    680     227   5.395 0.00198
epde$STAI_tratto:epde$attivita 3     94      31   0.748 0.52656
Residuals                     79   3319      42
```

Dovremmo seriamente pensare a trasformare questa ANCOVA in un multilevel model.

E per i disegni a misure ripetute? È sconsigliato usare l'ANCOVA tradizionale (anche se è possibile: §12.3.1), ed è invece opportuno ricorrere ai disegni multilevel (capitolo 15) .

13.4 ANOVA fattoriale e ANCOVA robuste

Se non sono rispettati i requisiti di normalità multivariata e omoschedasticità, o di indipendenza delle misure, è possibile ricorrere alle statistiche robuste dell'ormai ben noto package **WRS2**, di facile lettura.

Per un'ANOVA between a due vie su medie *trimmed*, si usa **`t2way(formula, dataframe, trimmed=)`**; i relativi post hoc sono forniti da `mcp2a(formula, dataframe)`.

Recuperiamo i dati di `ep`:

```
t2way(conoscenze_t0~gruppo*responsabilita, data=sicurezza, tr=.2)
```

```
Call:
t2way(formula = conoscenze_t0 ~ gruppo * responsabilita, data = sicurezza,
    tr = 0.2)
                      value p.value
gruppo                3.0005   0.243
responsabilita        3.6354   0.063
gruppo:responsabilita 0.1302   0.939
```

```
mcp2a(conoscenze_t0~gruppo*responsabilita, data=sicurezza)
Aggregation requires fun.aggregate: length used as default
Call:
mcp2a(formula = conoscenze_t0 ~ gruppo * responsabilita, data = sicurezza)

                               V1  ci.lower ci.upper p-value
gruppo1                  14.66667 -18.00000 21.33333 0.37896
gruppo2                 -25.33333 -40.00000 -6.00000 0.00167
gruppo3                 -40.00000 -42.00000 -9.33333 0.00000
responsabilita1         -14.66667 -37.33333  2.00000 0.04007
gruppo1:responsabilita1 -14.66667 -30.66667 18.00000 0.24708
gruppo2:responsabilita1 -14.66667 -22.00000  5.33333 0.06177
gruppo3:responsabilita1   0.00000 -20.00000 16.00000 0.19199
```

L'analogo a tre vie è **t3way(formula, dataframe, trimmed=)**.

Se invece si vuol calcolare l'ANOVA a due vie sulle mediane, si può usare **med2way(formula, dataframe)**:

```
med2way(conoscenze_t0~gruppo*responsabilita, data=sicurezza)
Call:
med2way(formula = conoscenze_t0 ~ gruppo * responsabilita, data = sicurezza)

                      value p.value
gruppo                0.6890  0.5021
responsabilita        0.2967  0.5859
gruppo:responsabilita 0.2654  0.8757
```

Per ANOVA a misure ripetute mista a due vie su medie *trimmed*, usiamo **bwtrim(formula, dataframe, trimmed=)**; naturalmente, si applica su dataframe in formato *long*.

```
sicu <- melt(data = sicurezza, id.vars = c("codice", "gruppo"), measure.vars = c("conoscenze_t0",
   "conoscenze_t1", "conoscenze_t2"))

names(sicu)<-c("sogg", "gruppo", "tempo", "conoscenze")

bwtrim(conoscenze~gruppo*tempo, data= sicu, id=sogg,tr= .2)
Call:
bwtrim(formula = conoscenze ~ gruppo * tempo, id = sogg, data = sicu,    tr = 0.2)
               value p.value
gruppo         6.3479  0.0042
tempo        142.4262  0.0000
gruppo:tempo  70.1786  0.0000
```

Infine, per un'ANCOVA robusta per due gruppi indipendenti e una covariata, si usa **ancova(formula, data, tr=.2, fr1=1, fr2=1)**, dove `fr1=1` e `fr2=1` indicano che lo span della covariata per ogni gruppo non è specificato.

```
ancova(conoscenze_t0~eta+genere, data= sicurezza, fr1=1, fr2=2)
Call:
ancova(formula = conoscenze_t0 ~ eta + genere, data = sicurezza, fr1 = 1, fr2 = 1)
         n: F n: M trimmed mean diff     se lower CI upper CI statistic p-value
eta = 25   21   17           -1.0490 1.6035  -5.5639   3.4660    0.6541  0.5204
eta = 34   56   22           -0.8782 1.1100  -3.8757   2.1194    0.7911  0.4340
eta = 39   58   22            0.0516 1.1441  -3.0279   3.1311    0.0451  0.9643
eta = 45   49   22            0.5829 1.1324  -2.4744   3.6403    0.5148  0.6098
eta = 59   15   20           -2.1111 2.4982  -9.8194   5.5971    0.8450  0.4173
```

Capitolo 14

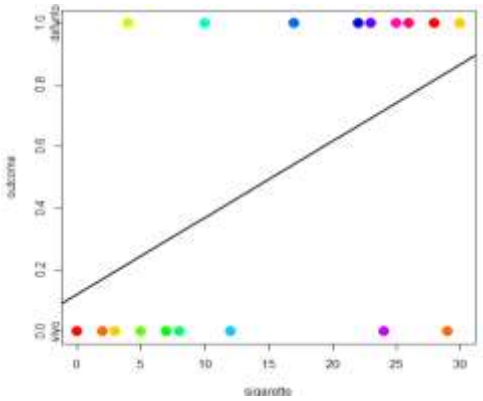
Modelli lineari generalizzati: la regressione logistica

In questo capitolo useremo il dataframe ***cuore***.

Nei modelli lineari visti nei capitoli precedenti, si assume (e si verifica!) che la relazione tra Y e X sia – appunto – lineare: quando questa **condizione non si verifica**, sia perché la natura di Y **è categoriale** (questo è il caso trattato in questo capitolo), sia perché la relazione **con Y continue** proprio non ne vuole sapere di essere lineare, il **modello** lineare generale (definito dal **metodo dei minimi quadrati**) **non** può essere applicato.

Per esempio, rappresentiamo la relazione definita dal **modello $outcome \sim sigarette$**: rappresentiamo il numero medio di sigarette fumate al giorno dal 2000 al 2014 (X) da un gruppo di soggetti di cui ricaviamo all'anagrafe lo stato biologico ($Y_1 = defunto$; $Y_0 = vivo$):

```
sigarette<-c(0,2,3,4,5,7,8,10,12,17,22,23,24,25,26,28,29,30)
outcome<-c(0,0,0,1,0,0,0,1,0,1,1,1,0,1,1,1,0,1)
mod<-lm(outcome~sigarette)

plot(sigarette, outcome, col=rainbow(15), pch=19, cex=2)
abline(mod, lwd=2)
mtext(text = c("vivo", "defunto"),side = 2,at = c(0,1))
```



La **retta di interpolazione** – ovvero il modello lineare – è calcolabile, ma **non può adattarsi ai punti**, perché gli errori (distanza tra i punti y_i e $\hat{y}_i$ sulla retta) dipendono solo dalla variabilità in X (numero di sigarette), non dalla relazione tra X e Y; inoltre, i valori $\hat{y}_i \neq 0$ o $\hat{y}_i \neq 1$ non hanno senso.

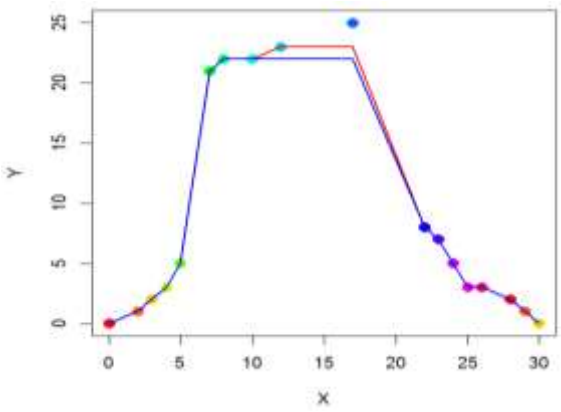
Per rappresentare la corretta relazione tra sigarette ed outcome, evidentemente non lineare, si possono usare **funzioni di smoothing**: uno **smoother** è un processo algoritmico non parametrico che non impone una forma alla relazione sul grafico, ma al contrario consente di far **emergere la forma della relazione tra Y e X**.

Tra i molti disponibili in R, proviamo a usare `runmed(x= variabile dipendente, k= )`, che utilizza le **running medians** o **mediane mobili**[106]: sono le mediane di **sottoinsiemi adiacenti di dati ordinati**, di **ampiezza definita da k**, che deve essere **dispari**: "**3**" è **l'ampiezza minima suggerita** per eliminare gli outliers che creano il "rumore di fondo" e impediscono l'emergere della forma della relazione. Al crescere di k, gli outliers vengono via via sempre più esclusi dalla forma della relazione definita nel grafico: lo smoothing ("smussatura") della curva si accentua e la curva si regolarizza.

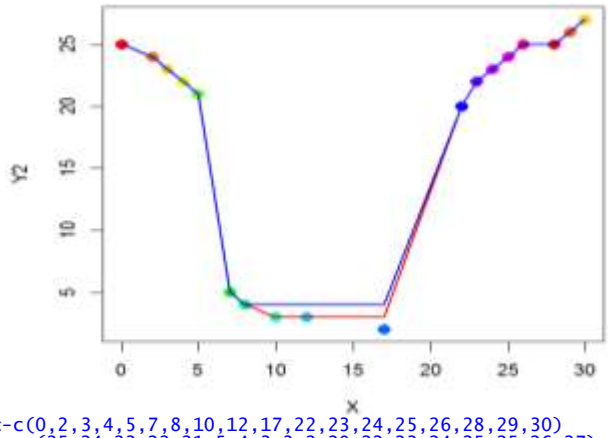
[106] Se volete un esempio di vettore di mediane mobili, provate a eseguire questo piccolo script:

```
V<-c(0,2,3,4,5,7,8,10,12,17,22,23,24,25,26,28,29,30)
mediane_mobili<-c(median(V[1:3]),        median(V[2:4]),median(V[3:5]),median(V[4:6]),median(V[5:7]),        median(V[6:8]),
  median(V[7:9]), median(V[8:10]), median(V[9:11]), median(V[10:12]), median(V[11:13]), median(V[12:14]), median(V[13:15]),
  median(V[14:16]), median(V[15:17]), median(V[16:18]))
mediane_mobili
```

Vediamo due esempi con dati inventati, in cui la relazione tra X e Y non è monotonica (in rosso la curva individuata da k=3, in blu quella definita da k=7):



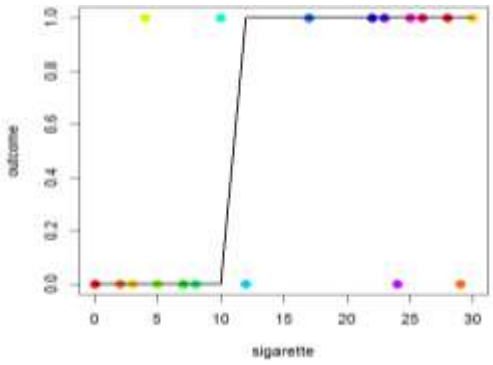

```
X<-c(0,2,3,4,5,7,8,10,12,17,22,23,24,25,26,28,29,30)
Y<-c(0,1,2,3,5,21,22,22,23,25,8,7,5,3,3,2,1,0)
plot(X, Y, pch=19, cex=1.5, col=rainbow(15))
lines(X, runmed(Y, k=3), lwd=2, col="red")
lines(X, runmed(Y, k=7), lwd=2, col="blue")
```




```
X<-c(0,2,3,4,5,7,8,10,12,17,22,23,24,25,26,28,29,30)
Y2<-c(25,24,23,22,21,5,4,3,3,2,20,22,23,24,25,25,26,27)
plot(X, Y2, pch=19, cex=1.5, col=rainbow(15))
lines(X, runmed(Y2, k=3), lwd=2, col="red")
lines(X, runmed(Y2, k=7), lwd=2, col="blue")
```

Rifacciamo ora il plot della relazione `sigarette~outcome`, e, con `lines`, sovrapponiamogli lo smoother[107]:

```
plot(sigarette, outcome, pch=19, cex=1.5, col=rainbow(15))
lines(sigarette, runmed(outcome, k=3), lwd=2)
```



La forma della relazione tra sigarette ed outcome è, con ogni evidenza, decisamente **logistica.**

Ora: non potendo modificare la natura di Y, è necessario **modificare il modello**, ovvero la relazione che lega Y a X, passando dal modello lineare generale (***GLM***) al **modello lineare generalizzato** (***GGLM***), che consente di interpretare la relazione tra variabili con la stessa logica del modello lineare generale. Vengono generalizzati **due parametri** del modello:

1. **la funzione che lega X e Y (*link function LF*): $f(y) = b_0 + b_1 x_i + e_i$**. Applichiamo una funzione che renda lineare la relazione tra X e Y, e che, nel caso del GLM, è una funzione d'identità: $f(y) = y$.
2. La **forma della distribuzione di Y**, che non è normale.

Il criterio per scegliere la link function che linearizzi la relazione $Y \sim X$ è quello di usare **la link function inversa della link function tra le variabili**. Per esempio, se la relazione che lega X e Y è "$Y = X^2$", allora la funzione inversa a questa relazione è "$Y = \sqrt{X}$)". A seconda del tipo di relazione esistente tra X e Y, quindi, serviranno di volta in volta *link functions* diverse per linearizzare la relazione nel $GGLM$.

In questo capitolo, vedremo un caso particolare di modello lineare generalizzato, in cui Y è una variabile categoriale, a due (§8.1) o più livelli (§8.2), predetta da una o più indifferentemente continue o categoriali: la regressione logistica.

[107] Le funzioni di base `lines(loess(x~y))` e `lines(smothspline(x,y))`, come anche `lines(SmothSpline(x,y))` di `DescTools`, aggiungono anche l'utile ***CI*** dello smoother; i loro oggetti sono particolari forme di interpolazione – ma dovremmo entrare molto più a fondo nell'amplissimo mondo della regressione non parametrica per capirle per bene: limitiamoci a segnalarle ai curiosi o per usi futuri.

14.1 Regressione logistica binaria (o dicotomica)

Nella regressione logistica si valuta la relazione di **predicibilità** esistente tra una variabile $\boldsymbol{Y}$ **categoriale** (*criterio o dipendente*) a partire dai valori di **almeno una** variabile $\boldsymbol{X}$ (*predittore* o *indipendente*), indifferentemente **continua o categoriale**. Nel caso della regressione logistica binaria, $\boldsymbol{Y}$ **è composta da due categorie**, cioè è **dicotomica**.
La domanda fondamentale cui si risponde con la regressione logistica è: **conoscendo il valore x_j** assunto in X dal soggetto j, sono in grado di **predire la probabilità che il soggetto j appartenga a una data categoria Y_j piuttosto che a una categoria $Y_{\neg j}$**? Ad esempio, la regressione logistica semplice binaria risponde alla domanda: "conoscendo il numero di sigarette al giorno fumate da Tizio (x_j), posso predire se è più probabile che Tizio tra vent'anni appartenga alla categoria "$Y_{1\ Defunto}$" della variabile Y Stato biologico, piuttosto che alla categoria "$Y_{0\ Vivo}$"?
Oppure: "Conoscendo solo il numero di partite di calcio viste in TV all'anno (x_{1j}), il suo essere padrone di un cane (X_{2j}) e lo stipendio annuo (x_{3j}) di Caio, posso predire se Caio è più probabilmente maschio (Y_1) che femmina (Y_0)?": questo quesito risponde la regressione logistica multipla binaria. Invece, per: "conoscendo la quantità di cortisolo di Sempronio (x_j), posso predire se Sempronio è più probabilmente un monaco zen (Y_0) che un broker di Borsa (Y_1) o uno psicologo libero professionista (Y_3)?": serve la regressione logistica (semplice) multinomiale, oggetto del §8.2.

Se la funzione di relazione tra X e Y nel modello lineare generale è data da:
$y_i = (b_0 + b_1 x_i) + e_i$ o $y_i = (b_0 + b_1 x_i + b_2 x_{2i} \ldots + b_n x_{ni}) + e_i$, nel modello di predizione tra X e Y categoriale la **probabilità che i appartenga** alla **categoria Y** è data dai **valori assunti da i nella X (o nelle X)** – più l'errore di predizione:

Semplice
$$P(Y) = \frac{1}{1 - e^{-(\beta_0 + \beta_1 X_i)}}$$

Multipla
$$P(Y) = \frac{1}{1 - e^{-(\beta_0 + \beta_1 X_i + \beta_2 X_{2i} \ldots \beta_1 X_m)}}$$

$\boldsymbol{P(Y)}$ è la **probabilità di Y** (quando tende a 0, è molto improbabile che Y si verifichi sotto condizione di H_0)**, e** è **la base del logaritmo natural**e, e i **coefficienti b_0 e b_1** sono identici a quelli del modello lineare: ogni predittore nell'equazione ha il proprio b_1. Per calcolare i parametri del modello generalizzato, però, **non usiamo il metodo dei minimi quadrati** ($\boldsymbol{OLS}$: *ordinary least squares*), ma la **stima della massima verosimiglianza** (**Maximum likelihood – ML**), che **seleziona quei coefficienti che rendono i valori osservati massimamente probabili**.

Una legittima domanda è: **perché usare i logaritmi**? Perché la $\boldsymbol{LF}$ che unisce $\boldsymbol{X}$ **a** $\boldsymbol{Y}$ in questo $GGLM$ **è il $\boldsymbol{logit}$**. Vediamolo.
Predire Y significa attribuire una probabilità a che Y si verifichi ($Y = 1$) rispetto alla probabilità che Y non si verifichi ($Y = 0$), ovvero **stimare la probabilità dell'evento atteso (p_1) rispetto alla probabilità che esso non si verifichi (p_0)**, corrispondente al numero di casi che verificano Y_1 (n_1) rispetto al numero di casi che verificano Y_0 (n_0).

Questo rapporto si chiama odds e l'abbiamo già trovato, in un altro contesto, nel Capitolo 7: $odds = \frac{p_1}{p_0} = \frac{n_1}{n_0}$

Ricordiamo: se l'odds di **essere astinenti dal fumo (Y_1)** dopo un certo numero di sedute di una costosa terapia è **odds=2**, vuol dire che la probabilità di essere astinenti (Y_1) è il doppio della probabilità di essere ancora fumatori (Y_0), ovvero che per ogni persona ancora fumatrice ci sono due astinenti.

Dalla formula dell'odds, è facile dedurre che:

- se $p_1 < .5$, allora $odds < 1$
- Se $p_1 = .5$, allora $odds = 1$
- Se $p_1 > .5$, allora $odds > 1$

Quindi, conoscendo il valore in X di un dato soggetto i, vorremmo poter predire il suo odds, cioè la probabilità p_1 che appartenga a una categoria Y_1 invece che alla categoria Y_0 (p_0): conoscendo il suo grado di dipendenza di nicotina (X_i), siamo in grado di predire quanto è più probabile / meno probabile che dopo una terapia un soggetto sia astinente (Y_1) invece che fumatore (Y_0)?

La relazione tra Y e la/le X, lo sappiamo, non è lineare, essendo Y categoriale. La **link function che linearizza la relazione tra Y e X è il $logit$**: la **funzione $logit$ è l'inverso della curva logistica** che descrive la relazione tra Y dicotomica e X continua, e consiste nella **trasformazione in logaritmo naturale dell'odds ($\log(x)$)**.

Ricordando le proprietà dei logaritmi (se avete bisogno di un rapido ripasso, guardate l'Appendice VI), abbiamo:

- se $x < 1$, allora **logaritmo negativo** → `round(log(.75),2)` → `-0.29`
- se $x = 1$, allora **logaritmo $= 0$** → `log(1)` → `-0`
- se $x > 1$, allora **logaritmo positivo** → `round(log(2.5),2)` → `0.92`

Pertanto, l'odds, trasformato nel suo logaritmo naturale, assumerà valori positivi e negativi corrispondenti a una probabilità dell'evento atteso maggiore (segno $+$) o minore (segno $-$) a quella dell'evento non atteso:

- se $p_1 < .5$, allora il **$logit$ dell'odds è negativo**
- se $p_1 = 1$, allora il **$logit$ dell'odds $= 0$**
- se $p_1 > .5$, allora il **$logit$ dell'odds è positivo**

Il modello $GGLM$ che utilizza il $logit$ come link function tra X e Y per linearizzare la loro relazione si chiama, per l'appunto, **logistico**: la regressione logistica è quindi un $GGLM$ in cui la LF tra X e Y è il $logit$ (logaritmo dell'odds) della probabilità di appartenere ad una categoria (p_1) rispetto alla probabilità di non appartenervi (p_0).

$$\log\left(\frac{p_1}{p_0}\right) = a + b_{xy}$$

I **valori predetti** $\hat{Y}$, cioè i **valori $\hat{logit}$ di Y**, per ogni valore X_i sono:

$$logit_{x=j} = b_0 + b_1 X_j$$

L'interpretazione dei coefficienti b_0 e b_1 nel $GGLM$ è analoga a quella di una regressione OLS, con qualche specificazione: per interpretarli, va **eliminata la scala logaritmica** in cui sono espressi usando il suo inverso[108]: la **funzione esponenziale exp^b**.

$$exp^b(ln_n) = x$$

- b_1 esprime il $logit$ dell'odds, ovvero il $logit$ della probabilità di appartenere alla categoria Y_1 rispetto alla categoria Y_0. **L'esponenziale di b_1** è un **odds ratio** (OR), ovvero un **rapporto tra odds** (capitolo 7): indica di **quante volte dobbiamo moltiplicare l'odds atteso per ogni unità in più / in meno di X** (per ogni seduta in più di psicoterapia, la probabilità di astenersi dal fumare aumenta/diminuisce di N volte; per ogni lezione seguita in più, la probabilità di essere promossi aumenta/diminuisce di N volte).
    - se b_1 è positivo, all'aumentare di X aumenta la probabilità di rientrare nella categoria Y_1;

[108] L'esponenziale di una somma di logaritmi è uguale al prodotto degli argomenti dei logaritmi: $exp^b(ln_x + ln_y) = x \times y$; l'esponenziale di una differenza tra logaritmi è uguale al rapporto tra gli argomenti dei logaritmi $exp^b(ln_x - ln_y) = x/y$

- se b_1 è negativo, all'aumentare di X diminuisce la probabilità di rientrare nella categoria Y_1;
- se $b_1 = 0$, la probabilità di rientrare nella categoria Y_1 è uguale alla probabilità di rientrare nella categoria Y_0: Y non è influenzata da X, la relazione non è significativa.

- b_0 indica il valore **$logit$ atteso quando $X = 0$**; quindi, **l'exp^b di b_0** esprime **l'OR atteso quando $X = 0$**.

Come si calcolano, allora, i parametri b_0 e b_1 del modello lineare generalizzato con il metodo della Massima Verosimiglianza? Con un complesso metodo iterativo.

La ML valuta la **verosimiglianza dei parametri del modello sotto condizione di H_0**: $y_i = b_0 + e_i$. Come usuale, H0 prevede l'assenza di relazione tra X e Y ($b_1 = 0$), usando come stima del modello la ***likelihood function*** LF, che indica quanto è probabile ottenere i dati osservati in un modello in cui le misure siano determinate solo dal fatto che i soggetti appartengono a una data popolazione (b_0, $grand\ mean$) e su di essi agisce l'errore casuale (e_i), ovvero in un modello in cui $b_1 = 0$ o **modello nullo**.

Nella regressione multipla OLS abbiamo visto che, per valutare se un modello descrive bene i dati, possiamo confrontare i valori osservati e i valori predetti (ricordate che R^2 multiplo è la correlazione tra valori predetti e valori osservati?). Similmente, possiamo usare i valori osservati e i valori predetti di un'equazione di un $GGLM$ per stimare il fit del modello.

L'analogo della SS_R nel $GGLM$ è la **log – likelihood (LL)**, cioè la **somma delle probabilità associate agli outcome predetti ed osservati**: indica quanta informazione **rimane non spiegata** dopo aver adattato il modello ai dati, e quindi è **inversamente proporzionale al fit**: **tanto minore è, tanto migliore è il modello**.

$$LL = \sum_{i=1}^{N} \left[Y_i ln\left(P(Y_i)\right) + (1 - Y_i) \ln\left(1 - P(Y_i)\right) \right]$$

Legata alla log-likelihood è la **devianza**, pari a **meno due volte la LL** → indicata come **-2 log-likelihood ($-2LL$)**, è quindi anch'essa inversamente proporzionale al fit (più è piccola, meglio è). Dato che $-2LL$ **segue la distribuzione χ^2**, al contrario di LL che non segue una distribuzione di probabilità nota, la usiamo quando dobbiamo attribuire $p - value$ per stabilire significatività. In particolare, usiamo la **$-2LL$ di modelli per stimare differenze** tra loro, perlopiù tra un modello "di baseline" e un modello in cui sono inclusi i predittori previsti dall'ipotesi.

Non sorprenderà che il modello "di baseline" è, come nella regressione OLS, il **modello nullo**, in cui è inserita solo b_0. Nella regressione lineare, b_0 nel modello nullo rappresentava la media della distribuzione Y, ovvero il miglior modello a disposizione per stimare una data Y_i. Poiché nel $GGLM$ Y è categoriale, e quindi non è descrivibile da medie, la stima migliore della probabilità di Y_i, **in assenza di altri predittori**, è la **categoria con la maggior frequenza di osservazioni.**

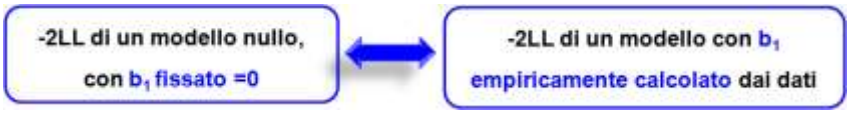
Ricordate il dataset `fumatori`?

```
table(fumo$outcome_3_mesi)
astinente    fumatore
       86          40
```

Se dopo tre mesi di terapia gli Astinenti (1) sono 86 e i Fumatori (0) sono 40, in assenza di qualsiasi altra informazione il modo migliore per scommettere se Gino sia un Astinente (1) o un Fumatore (0) è puntare sulla sua appartenenza alla categoria più numerosa…

L'informazione non spiegata da questo modello nullo ($-2LL$) diminuisce se aggiungo (almeno) un predittore?

Ovvero, l'informazione veicolata dal predittore migliora la capacità del modello di predire correttamente l'appartenenza del soggetto alla categoria attesa? Per rispondere, si confronta la $-2LL$ del modello nullo con quella del modello con almeno un predittore:



Esprimeremo quindi la **differenza fra le devianze dei modelli** come un **quantile** χ^2: $\chi^2 = 2LL_{predittore} - 2LL_{nullo}$ → $\chi^2 = \left(-2LL_{nullo} - \left(-2LL_{predittore}\right)\right)$

con $df = k_{predittore} - k_{nullo}$, dove k è il numero di parametri inseriti nel modello (uno solo, cioè b_0, nel modello nullo; due nel modello con b_0 e un b_1, ecc.): in questo modo si può **attribuire un** $p - value$ **alla differenza tra i due modelli sotto condizione di ipotesi nulla**: se si respinge H_0 la devianza del modello con un predittore è significativamente differente da quella del modello nullo → la X **contribuisce significativamente a predire l'appartenenza dell'osservazione alla categoria di riferimento**.

Questo appena descritto è il **Likelihood Ratio Test**[109] **(*LRT*)**: più il valore della statistica del test è $> \mathbf{0}$, più il modello che comprende b_1 è maggiormente probabile[110] di quello in cui esso non agisce (modello nullo), e quindi è più verosimile che $b_1 \neq 0$.

Il LRT è un test overall, come F nel metodo dei minimi quadrati. Per stimare se **l'apporto dei singoli predittori** inseriti nel modello sia significativo ($b_1 \neq 0$) e se $b_0 \neq 0$ si usa il **test *z*** di **Wald**, che segue la distribuzione normale.

$$Z_{b_0} = \frac{b_0}{SE_{b_0}} \quad Z_{b_1} = \frac{b_1}{SE_{b_1}}$$

Attenzione, però: quando il coefficiente b_0 o b_1 è grande, gli SE relativi tendono a gonfiarsi, e quindi il test di Wald può risultare sottostimato, rendendo più probabile un errore di II tipo (Menard, 1995)

Anche nel $GGLM$, oltre a stimare se $LRT \neq 0$, dobbiamo **quantificare il fit**, dato che anche la **distribuzione** χ^2 risente della numerosità campionaria: LRT che esprimono un potere predittivo debole, purché il campione sia ampio, risultano facilmente $\neq 0$. In analogia al coefficiente R^2 del GLM, si usano **coefficienti di godness of fit** definiti $pseudo - R^2$, basati sulla devianza $-2LL$, il cui range di variazione va **da 0 a 1**: più tendono a 1, più il modello è adeguato. In letteratura sono stati proposti molti coefficienti di questo tipo (si veda, per esempio, la bella rassegna di Menard, 2000): qui ne approfondiamo solo alcuni fra i più facilmente reperibili negli articoli di ricerca - in parte, più che per meriti intrinseci, perché forniti negli output dei più diffusi software statistici.

Nonostante la loro ampia diffusione in letteratura, **Hosmer e Lemeshow** (1989) invitano alla **cautela** nell'uso e nell'interpretazione dei coefficienti $pseudo - R^2$: in primis, obiettano a che possano essere considerati veri e propri indici di *goodness of fit*, dato che non mettono in relazione i valori osservati con quelli predetti dal modello (cosa che fanno, invece, R^2 e R_M^2), ma possono al più servire per stimare il diverso grado di fit dei modelli inseriti in una procedura di selezione per individuare il migliore del set, rispetto a una medesima Y. Inoltre, **tendono a essere perlopiù bassi**, se interpretati con gli standard usuali per R^2 e R_M^2 dei modelli lineari, il che potrebbe indurre un'indebita valutazione negativa dei modelli GLM cui si applicano.

Tuttavia, **almeno** un indice pseudo R^2, quello di **McKelvey – Zavoina**, si propone di essere una stima di valori latenti sulla base delle probabilità stimate dai dati, e dimostra di essere la migliore approssimazione a R_M^2 quando applicato a Y continue (ad esempio, Veall e Zimmermann, 1996), proponendosi quindi come, **probabilmente, il migliore stimatore**

[109] Più precisamente: il **logaritmo in base naturale del rapporto tra le *LF* esprime la differenza tra le due *LF***, per proprietà dei logaritmi: il logaritmo del rapporto tra due numeri corrisponde alla differenza tra i loro logaritmi: $log(50/10) = 1.609$; $log(50) - log(10) = 1.609$. Quindi: $-2 \times (log(50/10)) = -2 \times (log(50) - log(10))$.

[110] Si può trovare la probabilità di LRT espressa secondo la distribuzione F, ma, per quanto con N ampi le distribuzioni χ^2 e F tendano a sovrapporsi, in questo caso l'uso di F darà solo stime approssimate. La correzione all'approssimazione richiede la modifica dei df (**metodo di Kenward-Rogers**).

anche per modelli logistici. La sua formula è complessa, ma dato che R lavora per noi, concediamoci il lusso di imparare a usarlo.
Opportunamente avvertiti, quindi, vediamo i quattro indici su cui ci potremo concentrare.

L'indice R_L^2 di **McFadden** (1974) è uno stimatore logicamente **corrispondente all'analogo** GLM: si calcola dividendo il χ^2 del LRT (che rappresenta il cambiamento dalla baseline), ovvero la differenza tra le devianze dei modelli, per la devianza del modello nullo.

$$R_L^2 = \frac{\left(-2LL_{nullo} - \left(-2LL_{predittore}\right)\right)}{-2LL_{nullo}}$$

Indica la **riduzione nel valore assoluto della devianza**, quindi di quanto migliori il fit grazie all'inserimento dei predittori nel modello. Tuttavia, può sottostimare la forza della relazione se Y è una dicotomizzazione di una variabile continua (ad esempio, promosso – bocciato; Hagle e Mitchell, 1992). Anche se la critica è meno applicabile se Y è realmente dicotomica (De Maris, 1992), spesso il fit del modello risulta sottostimato.

Lo $pseudo - R^2$ **di Cox e Snell** (1989) tiene in conto anche la **numerosità del campione**; tuttavia, il suo valore massimo può essere < 1 **anche nel caso di un modello perfetto**[111], e ciò rende complicata l'interpretazione del fit, che è perlopiù sottostimato (Nagelkerke, 1991; Ryan, 1997).

$$R_{CS}^2 = 1 - exp\frac{\left(-2LL_{predittore} - \left(-2LL_{prednullo}\right)\right)}{N}$$

Lo $pseudo - R^2$ **di Nagelkerke** (1991) riconduce il coefficiente di determinazione di Cox nel **range** $0 - 1$, dividendolo per il massimo ottenibile rispetto ai dati a disposizione. Il fit stimato è in genere **maggiore rispetto** alle sottostime di R_L^2 **e** R_{CS}^2.

$$R_N^2 = \frac{R_{CS}^2}{1 - exp\left(-\frac{-2LL_{nullo}}{N}\right)}$$

Lo $pseudo - R^2$ **di McKelvey - Zavoina** (1975; 1991) sottintende l'esistenza di una sottostante variabile latente, non osservabile, continua (Finney, 1962).

$$R_{MZ}^2 = \frac{\hat{V}ar_{y*}}{\hat{V}ar_{y*} + Var_{err}}$$

Poiché la variabile continua non è disponibile, i coefficienti del modello sono utilizzati per stimare la varianza spiegata tramite la varianza dei valori predetti, che viene divisa per la somma della varianza delle risposte predette e della varianza di errore (formula di Veal e Zimmermann, 1996). Nei modelli di regressione logistica, la varianza di errore è fissata a $\pi^2/6$. L'indice è difficile da stimare, ma grazie a R non ce ne dovremo preoccupare.

Per **confrontare il fit di modelli riferiti alla medesima** Y **e con un diverso numero di predittori**, come nella regressione OLS possono essere usati AIC (e $AICc$) o BIC, in cui k **sono i parametri del modello** e N **il numero di osservazioni**.

$$AIC = -2LL + 2k \quad BIC = -2LL + ln(N) \times 2k$$

Nuovamente, il criterio per scegliere il modello sarà quello di **selezionare il modello con** AIC **(o** BIC**) inferiore**. Nel capitolo 5 i due coefficienti sono stati descritti un po' sommariamente, ma meritano qualche riga – e qualche critica in più. La letteratura su questi coefficienti è ampia, ma, come spesso abbiamo detto in altri contesti, piuttosto discorde sui loro meriti relativi. Un punto piuttosto sicuro, basato su ricerche di simulazione di adattamento di modelli aleatori, è che AIC, indipendentemente da N, soffre del bias di individuare modelli troppo grandi, **over fitted** (con parametri ridondanti,

[111] Il limite superiore di R^2_{CS} corrisponde a $1 - LF_0^{2/n}$

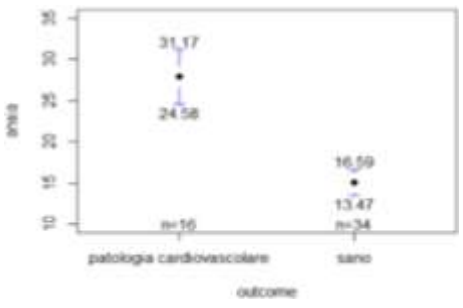
eccessivamente complessi). Al contrario, ***BIC*** non porta praticamente mai a scegliere un modello over fitted (purché N sia sufficientemente grande), ma, al contrario, porta più facilmente a scegliere modelli troppo piccoli, cioè **under fitted**, rilevando un peggioramento del fit all'aumentare dei parametri, di nuovo indipendentemente da N.
In effetti, nonostante la somiglianza tra le due formule sia palese, e porti a pensare che la differenza tra loro sia solo nella quantità di penalizzazione apportata al modello, i due coefficienti rispondono a scopi diversi. Detta semplicemente, mentre AIC tenta di selezionare il modello che più si adatta a una realtà sconosciuta (la popolazione) in un set di modelli che non potranno mai essere il modello generatore dei dati atteso in popolazione, BIC è una stima della **funzione della probabilità a posteriori che il modello sia vero** (ecco perché "Bayesian": capitolo 6): un BIC più basso implica che il modello deve essere considerato con maggiore probabilità il modello vero.
Concretamente, comunque, quando AIC e BIC individuano per gli stessi dati due diversi modelli come il migliore (evento non troppo frequente, ma possibile: ne vedremo un esempio nel Capitolo 9), la scelta del modello è responsabilità del ricercatore: si può suggerire che sarebbe meglio preferire AIC quando un falso negativo (l'esclusione di un predittore significativo) porta a conseguenze più gravi per la ricerca di un falso positivo (l'inclusione di un predittore non significativo), mentre si preferirebbe BIC nella situazione opposta, in cui l'inclusione di un predittore non significativo porta a distorsioni più gravi dell'esclusione di un predittore significativo.

La teoria alle spalle della regressione logistica può essere un po' sconcertante, dopo l'abitudine al modello lineare, ma l'esecuzione dell'analisi in R è molto semplice – anche se l'output della funzione di base **`glm(Y~X, dataframe, distribuzione di probabilità)`** trascura qualche informazione importante che dovremo calcolare separatamente. Vedremo successivamente una diversa funzione (**`mlogit`**: la useremo diffusamente nella regressione logistica multinomiale), che dà più informazioni, ma costringe a una preliminare modica del dataframe: potrete scegliere la funzione che preferite.
Con `glm` useremo, come primo esempio, i dati di **`cuore`**, ricavati da una ricerca su 50 uomini adulti, che hanno manifestato un episodio di cardiomiopatia tako-tsubo (o sindrome del cuore spezzato): questa sindrome si manifesta come attacco cardiaco in stretta concomitanza con uno stress acuto, e produce una modificazione morfologica, **reversibile**, del muscolo cardiaco, cui deve il nome. Una delle ipotesi della ricerca era che una predisposizione a manifestare ansia, operazionalizzata con il punteggio alla scala STAI–Tratto (X: range $0-40$) fosse predittiva della probabilità di aver sviluppato, sei mesi dopo l'episodio, una patologia cardiovascolare, invece di godere di una remissione completa: Y è, quindi, lo stato di salute, categorizzato in $Y_1 = sano$ e $Y_0 = patologia$)

```
table(cuore$outcome)
patologia cardiovascolare  sano
16                         34
```

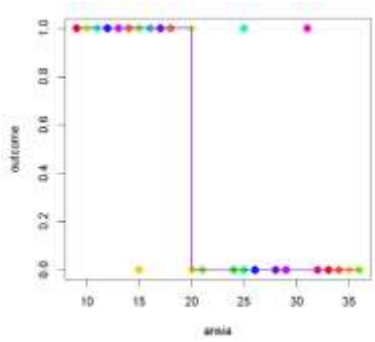
*Come sempre, se non cambiamo il livello di riferimento, R interpreterà il livello che precede in ordine alfanumerico come Y_0; nel nostro esempio la categoria Y_1 è **"Sano"**. Nella letteratura biomedica e psicologica, l'odds ha di solito un significato protettivo o preventivo: per questo si usa associare la categoria Y_1 a quella che prevede un esito favorevole – ma nessuno vieta di usare un'altra logica*

```
round(tapply(cuore$ansia, cuore$outcome, mean),2)
patologia cardiovascolare                      sano
                    27.88                     15.03
round(tapply(cuore$ansia, cuore$outcome, sd),1)
patologia cardiovascolare                      sano
                      6.2                       4.5
```



Effettivamente, l'ansia di stato dei pazienti è significativamente maggiore di quella rilevata nei soggetti sani.

Vediamo la forma della relazione $Y \sim X$ e per creare il modello lineare generalizzato applichiamo `glm`: **`glm(Y~X, data=, family= binomial)`**.



Poiché Y è dicotomica, la distribuzione di probabilità che descrive la distribuzione degli errori e la link function (`family`) che ci serve è quella **binomiale**. Se nel dataset ci fossero dati mancanti, aggiungeremmo: **`na.action= na.omit`** o **`na.action=na.exclude`**.

```
glm(outcome~ansia, data = cuore, family = binomial)
Call:  glm(formula = outcome ~ ansia, family = binomial, data = cuore)

Coefficients:
(Intercept) ← b0     ansia ← b1
      8.066        -0.357
df del modello: N - parametri nel modello
↓
Degrees of Freedom: 49 Total (i.e. Null) ; 48 Residual
                     ↑                      ↑
                     df modello nullo: N-1      df modello con predittore: N-1-1
Null Deviance:      62.69 ← -2LL del modello nullo
Residual Deviance:  27.51 ← -2LL del modello con predittore outcome   AIC: 31.51
```

Gli elementi del `Call` dicono già quasi tutto quel che serve, ma vediamo il più completo `summary` del modello, simile a quello di `lm`: dividiamolo nelle due parti corrispondenti a: sintesi degli errori del modello + parametri del modello e informazioni relative al modello nel complesso. Iniziamo da questa seconda parte:

```
solo_ansia<-glm(outcome~ansia, data = cuore, family = binomial)
summary(solo_ansia)
[...]
Null deviance:     62.687  on 49  degrees of freedom
Residual deviance: 27.507  on 48  degrees of freedom
AIC: 31.507
```

Come abbiamo visto nel `Call`, la null deviance è la $-2LL$ (devianza) del modello nullo, che contiene solo b_0 e non b_1 (quindi, i df corrispondono a $N-1$ parametro $\rightarrow 50-1$). La residual deviance è la devianza non spiegata del modello con il predittore Ansia (i suoi df corrispondono a $N-2\ parametri \rightarrow 50-2$): è inferiore alla devianza non spiegata del modello nullo, il che significa che **l'aggiunta del predittore ansia ha migliorato la capacità del modello di prevedere la probabilità del soggetto *j* di appartenere alla categoria Sano (Y_{1_j}) conoscendo il suo punteggio di ansia (X_j)**.

L'AIC, di per sé, non dice nulla, a meno di non **confrontarlo con quello del modello nullo** (con la sola intercetta), che può essere ricavato dal `summary`: se assistiamo a una riduzione dell'AIC passando dal modello nullo a quello con predittore, sapremo che l'ansia apporta un significativo miglioramento nella capacità del modello di prevedere l'outcome.

Nel modello nullo l'unico parametro è b_0, quindi $\boldsymbol{k = 1}$:

```
(AIC_nullo<-62.69+(2*1))
[1] 64.69
```

Conosciamo dal `summary` l'AIC del modello con predittore ansia, ma volendo calcolarlo (b_0 e b_1, quindi $\boldsymbol{k = 2}$):

```
(AIC_ansia<-27.51+(2*2))
[1] 31.51
```

Sì, l'AIC si dimezza aggiungendo il b_1 al modello: il predittore sembra davvero efficace. Con la formula, possiamo anche ricavare $\boldsymbol{BIC}$, che non è riportato nel modello:

```
(BIC_nullo<-62.69+(log(50)*1))
[1] 66.60202
(BIC_ansia<-27.51+(log(50)*2))
[1] 35.33405
```

Per fortuna, BIC conferma la variazione positiva del fit.

Disgraziatamente, R **non** riporta nel summary il likelihood ratio test (***LRT***), che ci può dire se **la differenza tra $-2LL$ del modello nullo e $-2LL$ del modello con predittore è significativa**: è come se `lm` ci fornisse solo MS_M e MS_R e poi dovessimo ricavarne noi F e $p-value$. Possiamo però facilmente ottenere LRT, sia trascrivendo a mano i valori delle due devianze e relativi gdl, sia, più elegantemente, ricavandoli dal modello `solo_ansia`, di cui sono **oggetti**:

```
LRT<-solo_ansia$null.deviance-solo_ansia$deviance
LRT
[1] 35.17963
```

Cioè:

```
62.687-27.507
[1] 35.18
```

Ora, basta ricordare che il risultato di LRT è un quantile di una distribuzione χ^2 con df corrispondenti alla differenza tra i df del modello nullo e i df del modello con predittore, per associare un $p-value$ al quantile ottenuto o a uno più estremo. Non dovreste aver dimenticato la funzione di ripartizione:

```
pchisq(LRT, 49-48, lower.tail = FALSE)
[1] 3.006515e-09
```

Sarebbe un risultato davvero eccezionale ottenere questa differenza tra devianze residue, se il predittore Ansia inserito nel secondo modello non avesse una relazione di predizione con l'outcome Y: l'ansia incide in maniera non casuale sulla probabilità di essere sano invece che sofferente di una patologia cardiaca tako-tsubo.

Quanto è forte questa relazione significativa? Per calcolare un indice di effect size usiamo `PseudoR2(modello generalizzato, which=)` di `DescTools`, specificando in `which` il tipo di $pseudo-R^2$; tra gli altri, troviamo `"McFadden"`, `"CoxSnell"`, `"Nagelkerke"`, `"McKelveyZavoina"` – attenti alle maiuscole:

```
PseudoR2(solo_ansia, which = "McFadden")
 McFadden
0.5611955
PseudoR2(solo_ansia, which = "CoxSnell")
 CoxSnell
0.5051956
PseudoR2(solo_ansia, which = "Nagelkerke")
Nagelkerke
 0.7069979
PseudoR2(solo_ansia, which = "McKelveyZavoina")
McKelveyZavoina
      0.7010983
```

Se vi danno fastidio le etichette di `PseudoR2`, in `performance` trovate `r2_coxsnell(modello)`, `r2_nagelkerke(modello)`, `r2_mckelvey(modello)` e `r2_mcfadden(modello)`, con output identico (`r2_mcfadden` dà anche uno $pseudo-R^2$ *adjusted*, ma resta il problema della sottostima).

Infine, osserviamo l'ultima riga:

```
Number of Fisher Scoring iterations: 6
```

Abbiamo detto che nel metodo della Massima Verosimiglianza il calcolo dei parametri del modello è **iterativo**: semplificando parecchio la reale procedura di calcolo, si procede per approssimazioni successive finché una stima non è significativamente differente da quella precedente. A questo punto il calcolo si interrompe (la **soluzione converge**, ovvero si raggiunge la convergenza) e nell'output è indicato il numero di iterazioni necessarie per giungere alla convergenza.

Ora possiamo dedicarci alla prima parte dell'output: essendoci un solo predittore, non ci riserva sorprese:

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3564  -0.1765   0.2121   0.4258   2.4690

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  8.06647    2.05404   3.927  8.6e-05
ansia       -0.35696    0.09653  -3.698 0.000217
```

b_1 indica la variazione nel $logit$ (logaritmo dell'odds) di $Y_1 = sano$ associata a una unità di cambiamento in X (ansia). **La relazione è negativa: per ogni punto in più di ansia, il $logit$ della probabilità di appartenere alla categoria $Y_{1\ Sano}$ diminuisce**. Di quanto diminuisce? Nel `summary` non è detto: si ottiene l'odds ratio facendo **l'esponenziale di b_1**, usando l'oggetto `$coefficients` del modello `solo_ansia`, che contiene i parametri del modello. Poiché ci serve solo il secondo dei due coefficienti, cioè b_1, specifichiamo:

```
exp(solo_ansia$coefficients[2])
    ansia
0.6998002
```

Oppure, più piattamente;

```
exp(-0.35696)
[1] 0.6998005
```

Per ogni punto di ansia in più, la **probabilità** di essere sano (Y_1) **diminuisce** di .**67 volte**. Dato che, come sempre, siamo interessati al dato in popolazione, chiediamo anche i CI dei parametri del modello, nella loro scala esponenziale:

```
round(exp(confint(solo_ansia)),3)
               2.5 %     97.5 %
(Intercept) 113.782 453837.155
ansia         0.555      0.818
```

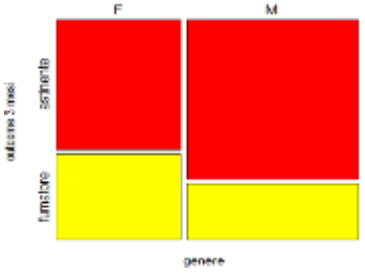
In popolazione, per ogni punto di ansia in più, la probabilità di essere sano (Y_1) **diminuisce** tra le .56 e le .82 volte. Notate che il CI del b_1 **non comprende $H_0 = 1$**, cioè il valore atteso dall'H_0 relativa a un OR.

Un'ultima nota sul test z associato ai parametri: in entrambi i casi respingiamo l'ipotesi nulla secondo cui $logit = 0 \rightarrow OR = 1$. In letteratura si può trovare, invece di z, uno **z al quadrato (z^2)**, che è la scelta di default di altri software di calcolo: la sua probabilità segue una distribuzione χ^2 e si definisce **statistica di Wald**. È possibile usare la statistica di Wald per calcolare un analogo di R^2, ma il suo uso è sconsigliato in presenza di ampi errori standard[112].

Vediamo il caso di una **regressione logistica con un predittore categoriale**: riapriamo il dataset `fumo` e rispolveriamo una vecchia ipotesi: la probabilità di essere astinenti a tre mesi dal termine della terapia ($Y_0 = astinente, Y_1 = fumatore$) è influenzata dal genere del paziente ($X_0 = femmina, X_1 = maschio$)?

Descriviamo i dati:

```
round(prop.table(table(fumo$genere, fumo$outcome_3_mesi),1) *100,1)
      astinente fumatore
  F        60.4     39.6
  M        74.0     26.0
```



La proporzione di maschi (X_1) astinenti (Y_0) è un po' più alta.

Se volessimo cambiare la categoria di riferimento da astinenti a fumatori, potremmo ricodificarla con `relevel(variabile, "nuova categoria 0")`.

[112] $pseudoR^2 = \sqrt{\frac{z^2 - 2df}{-2LL_{nullo}}}$

```
solo_genere<-glm(outcome_3_mesi~genere, data=fumo, family= binomial)
summary(solo_genere)
  Call:
  glm(formula = outcome_3_mesi ~ genere, family = binomial, data = fumo)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0046  -0.9475  -0.7765   1.3607   1.6407

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4212     0.2808  -1.500    0.134
genereM      -0.6233     0.3873  -1.609    0.108


    Null deviance: 157.48  on 125  degrees of freedom
Residual deviance: 154.88  on 124  degrees of freedom
AIC: 158.88


exp(-.6233)
[1] 0.5361721
exp(confint(solo_genere))
                2.5 %   97.5 %
(Intercept) 0.3731564 1.129966
genereM     0.2489693 1.143470
```

La relazione è negativa: passando da $X_{0\ Femmina}$ a $X_{1\ Maschio}$, la probabilità di appartenere alla categoria $Y_{1\ fumatore}$ si abbassa di .54 volte; però, sappiamo dal test z che il $logit$ dell'OR non è significativamente diverso da 0, e infatti il LRT non è significativo:

```
pchisq(157.48-154.88, 1, lower.tail=FALSE)
[1] 0.1068637
pchisq(solo_genere$null.deviance-solo_genere$deviance, 1, lower.tail=FALSE)
[1] 0.1066666
```

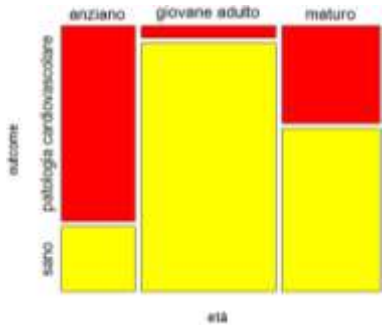
... e la relazione, secondo $lo\ pseudo - \boldsymbol{R^2_{MZ}}$, è di infima entità:

```
PseudoR2(solo_genere, which = "McKelveyZavoina")
McKelveyZavoina
     0.02797655
```

Notate che nel CI dell'odds ratio è contenuto il valore previsto da $H_1 = 1$ (la probabilità di essere astinente invece che fumatore è la stessa in uomini e donne).

Ora vediamo un predittore categoriale a più di due livelli, in cui torneremo a confrontarci con i **contrasti a coppie**. Nel dataframe **cuore**, un altro plausibile predittore dello sviluppo della sindrome tako-tsubo (Y; la categoria di riferimento è ancora $Y_{1\ Sano}$) è l'**età**, che è stata **suddivisa in tre categorie** ($\boldsymbol{X_a} = giovane\ adulto$; $\boldsymbol{X_b} = maturo$; $\boldsymbol{X_c} = anziano$).

```
mosaicplot(table(cuore$eta, cuore$outcome), col=rainbow(6),
  xlab="età", ylab="outcome")

round(prop.table(table(cuore$eta,  cuore$outcome),1) *100,1)
                 patologia cardiovascolare  sano
  anziano                             75.0  25.0
  giovane adulto                       4.5  95.5
  maturo                              37.5  62.5
```



In effetti, la patologia cardiovascolare è evidentemente più presente tra gli anziani, e anche tra giovani adulti e maturi la differenza sembra piuttosto chiara.

Il predittore `età` è un fattore, e i contrasti semplici previsti (grazie alle etichette) coincidono con contrasti sensati per le ipotesi: i più anziani saranno confrontati con i più giovani e con i maturi:

```
contrasts(cuore$eta)
              giovane adulto  maturo
anziano                    0       0  ← livello di riferimento di X
giovane adulto             1       0  ← I contrasto: probabilità anziani vs probabilità giovani adulti
maturo                     0       1  ← II contrasto: probabilità anziani vs probabilità maturi
```

Se volessimo **cambiare i contrasti**, procederemmo comunque **esattamente come fatto** nel modello lineare generale.

Vediamo quindi il modello logistico:

```
solo_eta<-glm(data=cuore, outcome~eta, family=binomial)
```

Prima ci concentriamo sull'ultima sezione del `summary`, che riguarda l'effetto del predittore nel complesso:

```
summary(solo_eta)
[…]
    Null deviance: 62.687  on 49  degrees of freedom
Residual deviance: 42.802  on 47  degrees of freedom
AIC: 48.802
```

La $-2LL$ del modello con i **due predittori dummizzati** (notate i df **della residual deviance**: nel modello abbiamo inserito **due** b_1) è **più bassa** del modello nullo: questa differenza è significativa?

```
pchisq(solo_eta$null.deviance-solo_eta$deviance, 2, lower.tail=FALSE)
[1] 4.808806e-05
```

Sì: l'età ha un effetto sulla probabilità di sviluppare una patologia cardiaca cronica; valutiamo l'intensità di questo effetto con R^2_{MZ}:

```
PseudoR2(solo_eta, which = "McKelveyZavoina")
McKelveyZavoina
      0.4698352
```

Potremmo dire che il predittore età ha un impatto (al più) medio, comunque inferiore a quello dell'ansia di tratto.

Ora vediamo nei **contrasti** se la probabilità degli anziani (X_0) di essere sani (Y_1) è significativamente diversa della probabilità dei giovani adulti (I contrasto) e della probabilità dei maturi (II contrasto).

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.0986     0.6667  -1.648 0.099369
etagiovane adulto  4.1431     1.2214   3.392 0.000694
etamaturo          1.6094     0.8433   1.909 0.056319
```

Vediamo subito gli OR:

```
round(exp(solo_eta$coefficients),2)
      (Intercept) etagiovane adulto         etamaturo
             0.33             63.00              5.00

exp(confint(solo_eta))
                       2.5 %      97.5 %
(Intercept)        0.0739549    1.117189
etagiovane adulto 8.0958744 1414.333137
etamaturo         1.0331158   30.144072
```

Passando dalla categoria anziano (X_0) alla categoria "**giovane adulto**" (X_1 del I contrasto), la probabilità di essere **sano** (Y_1) **aumenta** ($+$) di **63 volte** (OR), e questa variazione è significativa ($p < .001$) – ma il CI in popolazione è enorme!

Passando dalla categoria anziano (X_0) alla categoria "**maturo**" (X_1 del II contrasto), la probabilità di essere **sano** (Y_1) **aumenta** ($+$) di **5 volte** (OR), e questa variazione è **pressoché** significativa ($p = .056$).

Prima di passare al caso **con più predittori**, vediamo come usare **mlogit** (package **mlogit**) con lo stesso esempio outcome~età: indipendentemente dal fatto che il **dataframe** sia in formato *wide* o *long*, dobbiamo **trasformarlo**. Il **nuovo** formato sarà strutturato da **una riga per ogni categoria Y possibile per ogni caso**: ogni alternativa di Y è identificata come **TRUE**, se corrisponde a quella cui il soggetto appartiene, e come **FALSE** in caso contrario. Perciò, in questo esempio ogni soggetto sano sarà ripetuto in una riga TRUE e in una FALSE (TRUE per l'alternativa sano, FALSE per l'alternativa patologia), come ogni malato (TRUE per patologia, FALSE per sano).
Usiamo **mlogit.data(data= dataframe, choice, shape)**: l'argomento choice= richiede la Y categoriale ($gruppo), shape= identifica il formato del dataframe da trasformare (*wide*, nel nostro esempio):

```
multi_cuore<-mlogit.data(data = cuore, choice = "outcome", shape = "wide")
multi_cuore[15:18,]
```

```
  soggetto ansia            eta outcome outcome_numero chid                      alt
1       S8    17         maturo   FALSE             1    8 patologia cardiovascolare
2       S8    17         maturo    TRUE             1    8                     sano

3       S9    26 giovane adulto    TRUE             0    9 patologia cardiovascolare
4       S9    26 giovane adulto   FALSE             0    9                     sano
```

```
multi_cuore[41:44,]
```

```
  soggetto ansia     eta outcome outcome_numero chid                      alt
1      S21    18 anziano   FALSE             1   21 patologia cardiovascolare
2      S21    18 anziano    TRUE             1   21                     sano

3      S22    25 anziano    TRUE             0   22 patologia cardiovascolare
4      S22    25 anziano   FALSE             0   22                     sano
```

Nel nuovo dataframe **multi_cuore**, $chid è il numero identificativo del soggetto, $alt è l'alternativa di Y.

Eseguiamo la regressione logistica applicando **mlogit(formula= y~b0|X1, dataframe)**. Notate che nella formula va **espressa l'intercetta**[113] del modello, che, come abbiamo visto nel modello nullo della regressione multipla, si indica con $Y\sim1$, e che il predittore si inserisce facendolo precedere da |.

```
eta<-mlogit(formula = outcome~1|eta, data= multi_cuore)
summary(eta)
```

Il summary inizia proponendo le frequenze relative nelle categorie di Y:

```
Frequencies of alternatives:
patologia cardiovascolare     sano
                     0.32     0.68
```

← `prop.table(table(cuore$outcome))`

```
patologia cardiovascolare     sano
                     0.32     0.68
```

Seguono i coefficienti b_0 e b_1:

```
Coefficients :
                      Estimate Std. Error z-value  Pr(>|z|)
sano:(intercept)      -1.09861    0.66667 -1.6479 0.0993694
sano:etagiovane adulto  4.14313    1.22150  3.3918 0.0006943
sano:etamaturo          1.60944    0.84327  1.9086 0.0563191
```

Infine, il modello overall: sono riportati direttamente il $\boldsymbol{LRT}$ con $p-value$ associato e il coefficiente $\boldsymbol{R_L^2}$

```
(Log-Likelihood: -21.401
McFadden R^2:  0.31721
Likelihood ratio test : chisq = 19.885 (p.value = 4.8088e-05)
```

Non è riportato $\boldsymbol{AIC}$, ma lo si può ricavare facilmente dalla log-likelihood del modello, ricordandoci che $k=3$:

```
(eta$logLik*(-2))+(2*3)
```

```
'log Lik.' 48.80199 (df=3)
```

[113] il motivo per cui inseriamo b_0 nella formula è in realtà più complesso: il suo ruolo qui è effettivamente quello di "segnaposto"; se volete approfondire quale tipo di variabile stia sostituendo, fate riferimento all'Help di mlogit

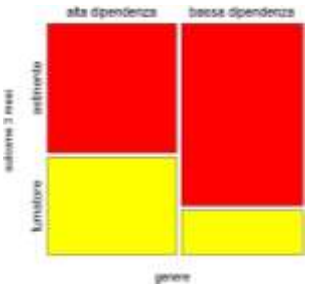
Vediamo ora il caso in **cui il modello ha più X**: il modello lineare generalizzato è flessibile tanto quanto il modello lineare, e accetta **anche predittori continui e categoriali** nello stesso modello.

Usiamo **due predittori categoriali** del dataframe `fumo`: l'outcome a 3 mesi ($Y_1 = fumatore$) è predetto dalla dipendenza a T0 ($X_{1_0} = alta\ dipendenza$; $x_{1_1} = bassa\ dipendenza$) e dal genere ($X_{2_0} = femmina$; $X_{2_1} = maschio$)?

```
contrasts(fumo$Fagerstrom_categorie)
                 bassa dipendenza
alta dipendenza                 0
bassa dipendenza                1
```

```
contrasts(fumo$genere)
  M
F 0
M 1
```

```
round(prop.table(table(fumo$Fagerstrom_categorie,fumo$outcome_3_mesi),1)
  *100,1)
                  astinente fumatore
 alta dipendenza       56.9     43.1
 bassa dipendenza      80.3     19.7
```



Sembra che tra chi ha bassa dipendenza i fumatori a tre mesi siano decisamente meno di quelli che si ritrovano tra i pazienti con forte dipendenza.

Un **modello a blocchi** si costruisce inserendo contemporaneamente i due predittori:

```
dipendenza_genere<-glm(data=fumo, outcome_3_mesi~Fagerstrom_categorie+genere, family=binomial)
```

Partiamo dal fondo del `summary`:

```
    Null deviance: 157.48  on 125  degrees of freedom
Residual deviance: 146.11  on 123  degrees of freedom
AIC: 152.11
pchisq(157.48-146.11,2, lower.tail=FALSE)
[1] 0.003396533
```

Nel complesso il modello è significativo: i due predittori, congiuntamente, hanno un effetto sulla probabilità di smettere di fumare. Vediamoli separatamente, ora:

```
summary(dipendenza_genere)
[...]
Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                        0.1484     0.3467   0.428  0.66871
Fagerstrom_categoriebassa dipendenza -1.1910     0.4165  -2.859  0.00425
genereM                           -0.7240     0.4050  -1.788  0.07384
```

Passando dalla categoria Alta dipendenza (X_{1_0}) alla categoria Bassa dipendenza (X_{1_1}) la probabilità di essere fumatori (Y_1) **diminuisce** in maniera **significativa**. Invece, passando dalla categoria Femmina (X_{2_0}) alla categoria Maschio (X_{2_1}), la probabilità di essere fumatori (Y_1) **diminuisce** in maniera **non pienamente significativa**. Quindi, mentre una bassa dipendenza a T0 è un solido fattore prognostico positivo a breve termine, l'essere maschi non lo è altrettanto chiaramente.

```
exp(dipendenza_genere$coefficients)
                 (Intercept) Fagerstrom_categoriebassa dipendenza     genereM
                   1.1599389                          0.3039192   0.4847978
exp(confint(dipendenza_genere))
                                      2.5 %    97.5 %
(Intercept)                       0.5879241 2.3141483
Fagerstrom_categoriebassa dipendenza 0.1302529 0.6733635
genereM                           0.2162178 1.0662362
```

Se avessimo adottato un approccio **gerarchico**, invece, avremmo costruito prima il modello con un predittore (dipendenza), poi quello con due predittori (dipendenza + fumo):

```
solo_dipendenza<-glm(data=fumo, outcome_3_mesi ~ Fagerstrom_categorie, family = binomial)
dipendenza_genere<-glm(data=fumo, outcome_3_mesi ~ Fagerstrom_categorie+genere,
   family=binomial)
```

... e avremmo **confrontato la variazione del fit** con **anova**: quando `anova` non si applica a modelli OLS, ma a $GGLM$ con la distribuzione χ^2, ne ricaviamo un output un po' diverso:

```
anova(solo_dipendenza, dipendenza_genere)
Analysis of Deviance Table

Model 1: outcome_3_mesi ~ Fagerstrom_categorie
Model 2: outcome_3_mesi ~ Fagerstrom_categorie + genere
  Resid. Df   Resid. Dev   Df   Deviance
1       124       149.35
2       123       146.11    1     3.2403
```

Ci viene restituita la **differenza tra le devianze** d'errore del modello 1 ($-2LL = 149.35$) e del modello 2 ($-2LL = 146.11$), ma **non la sua significatività**, che però possiamo facilmente ricavare, dato che sappiamo che la **differenza tra le devianze** d'errore (**3.2403**) si distribuisce come un quantile χ^2 per df corrispondenti a $df_1 - df_2 = 1$:

```
pchisq(3.2403, 1, lower.tail = FALSE)
[1] 0.07184748
```

La variazione tra i due modelli non è chiaramente significativa: aver aggiunto il genere non migliora significativamente la capacità predittiva dei due modelli. D'altronde, gli AIC diminuiscono quasi impercettibilmente:

```
AIC(solo_dipendenza); AIC(dipendenza_genere)-
[1] 153.3494
[1] 152.1092
```

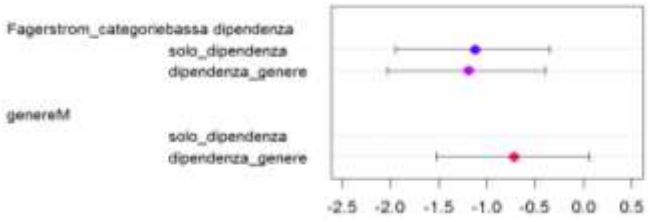
... e i BIC, più prudenti nel giudizio di variazione, addirittura si **alzano**:

```
BIC(solo_dipendenza); BIC(dipendenza_genere)
[1] 159.022
[1] 160.618
```

Ricordate `TMod(modello1, modello2,...)`, che abbiamo usato nella regressione lineare multipla? Se vi facilita il confronto tra modelli, potete usarla anche per modelli lineari generalizzati:

```
(gerarchico<-TMod(solo_dipendenza, dipendenza_genere)); plot(gerarchico)
```

	coef	solo_dipendenza	dipendenza_genere	
1	(Intercept)	-0.279	0.148	← b_0
2	Fagerstrom_categorie bassa dipendenza	-1.128 **	-1.191 **	← b_1 e significatività
3	genereM		-0.724 .	
4	---			
5	logLik	-74.675	-73.055	
6	logLik0	-78.742	-78.742	
7	G2	8.136	11.376	
8	AIC	153.349	152.109	
9	BIC	159.022	160.618	
10	numdf	2	3	
11	N	126	126	
12	n_vars	1	2	
13	McFadden	0.052	0.072	
14	McFaddenAdj	0.026	0.034	
15	Nagelkerke	0.088	0.121	
16	CoxSnell	0.063	0.086	
17	Kendall Tau-a	0.118	0.153	
18	Somers Delta	0.270	0.350	← coefficienti di associazione asimmetrica ($y \sim x$) per misure **ordinali**
19	Gamma	0.511	0.455	
20	Brier	0.203	0.199	← è una misura di accuratezza della previsione probabilistica, per outcome dicotomici: varia da 0 (del tutto accurata) a 1 (del tutto non accurata)
21	C	0.635	0.675	← misura di accuratezza della previsione equivalente all'AUC (*area under curve*) delle curve ROC: non è nel nostro programma



Per una **model selection non nidificata**, costruiamo la model class; due modelli li abbiamo già, ma rivediamo da capo:

```
nullo<-glm(data=fumo, outcome_3_mesi ~ 1, family = binomial)
dipendenza<-glm(data=fumo, outcome_3_mesi ~ Fagerstrom_categorie, family = binomial)
genere<-glm(data=fumo, outcome_3_mesi ~ genere, family = binomial)
dipendenza_genere<-glm(data=fumo, outcome_3_mesi ~ Fagerstrom_categorie+genere,
   family=binomial)

model.sel(nullo, dipendenza, genere,dipendenza_genere)
Model selection table
                    (Intrc) Fgr_ctg  gnr   df    logLik    AICc  delta  weight
dipendenza_genere    0.1484       +    +    3   -73.055   152.3   0.00   0.614
dipendenza          -0.2787       +         2   -74.675   153.4   1.14   0.347
genere              -0.4212            +    2   -77.441   159.0   6.67   0.022
nullo               -0.7655                 1   -78.742   159.5   7.21   0.017
```

Il modello migliore è quello con entrambi i predittori, che è 1.77 volte più verosimile, come migliore approssimazione al modello generatore dei dati, del modello con la sola dipendenza (ma entrambi i modelli entrano nel *confidence set*).

Se, infine, volessimo adottare un **approccio per passi**, mai molto raccomandabile, usiamo la procedura **step**, **backward dal modello completo** o **forward dal modello nullo**, proprio come nel modello lineare:

```
step(glm(data=fumo,   outcome_3_mesi~1,   family=binomial),   direction="forward",   scope   =
 ~Fagerstrom_categorie+genere)
Start:  AIC=159.48
outcome_3_mesi ~ 1
                       Df Deviance    AIC
+ Fagerstrom_categorie  1   149.35 153.35
+ genere                1   154.88 158.88
<none>                      157.49 159.49

Step:  AIC=153.35
outcome_3_mesi ~ Fagerstrom_categorie

         Df Deviance    AIC
+ genere  1   146.11 152.11
<none>       149.35 153.35
Step:  AIC=152.11
outcome_3_mesi ~ Fagerstrom_categorie + genere

Call:  glm(formula = outcome_3_mesi ~ Fagerstrom_categorie + genere,
    family = binomial, data = fumo)
Coefficients:
                 (Intercept)  Fagerstrom_categoriebassa dipendenza   genereM
                      0.1484                               -1.1910   -0.7240

Degrees of Freedom: 125 Total (i.e. Null);  123 Residual
Null Deviance:      157.5
Residual Deviance:  146.1 AIC: 152.1
```

Concludiamo con un modello con **un predittore continuo e uno categoriale**: valutiamo l'effetto congiunto di ansia (X_1, continua) ed età (X_2, categoriale), sullo sviluppo della sindrome cardiaca. Sappiamo che, separatamente, una maggior ansia diminuisce la probabilità di essere sano (Y_1) così come gli anziani (X_{2_0}) hanno minore probabilità di essere sani (Y_1) di giovani adulti (X_{2_1}) e maturi (X_{2_1}) nei due contrasti. Vediamo la model class:

```
nullo<-glm(data=cuore, outcome~1, family = binomial)
ansia<-glm(data=cuore, outcome ~ansia, family = binomial)
eta<-glm(data=cuore, outcome ~eta, family = binomial)
ansia_eta<-glm(data=cuore, outcome~ansia+eta, family = binomial)
model.sel(nullo, ansia, eta,ansia_eta)
Model selection table
          (Intrc)    ansia  eta  df   logLik  AICc   delta   weight
ansia      8.0660  -0.3750        2  -13.754   31.8    0.00   0.569
ansia_eta  5.5400  -0.2921    +   4  -11.716   32.3    0.56   0.431
eta       -1.0990             +   3  -21.401   49.3   17.56   0.000
nullo      0.7538                 1  -31.343   64.8   33.01   0.000
```

Il modello migliore è quello con la sola ansia (all'aumentare della predisposizione all'ansia, la probabilità di essere sani diminuisce, secondo il b_1), che è 1.32 volte più verosimile del modello che comprende anche l'età; quest'ultima, da sola, non è affatto verosimile come modello migliore tra quelli in competizione, pur essendo da preferire al modello con la sola intercetta.

14.1.1 Diagnostiche dei casi e violazione delle assunzioni

Come nel modello lineare, è possibile lavorare sui **residui** con `resid(modello)`: questi possono essere espressi come **residui standardizzati** e usati per rilevare gli outlier: `rstandard(modello)`.

È anche possibile ottenere il vettore dei **valori predetti** dal modello logistico, che sono concettualmente diversi da quelli del modello lineare: essi rappresentano la **probabilità di appartenenza alla categoria Y_1 dell'outcome dati i valori di ciascuno dei predittori X** inseriti nel modello: `fitted(modello)`.

Attenzione: nel modello lineare, abbiamo detto che i valori predetti dal modello potevano essere indifferentemente richiesti da `predict(modello)` e `fitted(modello)`. Nel modello lineare generalizzato **questo non è più vero**, perché in effetti `predict(modello)` restituisce i valori previsti dal modello **prima** che venga applicato l'inverso della link function tra X e Y, mentre `fitted(modello)` li mostra **dopo**[114]. Dato che la link function del modello lineare è una funzione identità, da prima a dopo (ovvero da `predict` a `fitted`) non c'è differenza, ma nella regressione logistica la link function è una funzione logistica, quindi la differenza c'è.

Vediamo come esempio il modello `ansia` che abbiamo appena costruito: creiamo il vettore dei valori predetti `cuore$fitted` (arrotondiamo a tre decimali) e **confrontiamo il valore predetto**, ovvero la probabilità stimata di appartenere alla categoria $Y_{1\ Sano}$, con **l'effettiva appartenenza di ogni soggetto** a una delle due categorie **dell'outcome**. Ordiniamo le categorie per visualizzare meglio:

```
cuore$fitted<-round(fitted(ansia),3)
predetti<-data.frame(outcome=cuore$outcome, fitted=cuore$fitted)
predetti<-predetti[order(predetti$outcome),]
predetti
```

	outcome	fitted	outcome	fitted	outcome	fitted
5	**patologia cardiovascolare**	**0.717**	1 sano	0.838	27 sano	0.956
7	patologia cardiovascolare	0.034	2 sano	0.913	30 sano	0.881
9	patologia cardiovascolare	0.229	3 sano	0.956	31 sano	0.838
10	patologia cardiovascolare	0.017	4 sano	0.984	33 sano	0.913
11	patologia cardiovascolare	0.008	6 sano	0.969	34 sano	0.938
14	patologia cardiovascolare	0.024	8 sano	0.881	35 sano	0.978
15	patologia cardiovascolare	0.229	12 sano	0.978	36 sano	0.984
18	patologia cardiovascolare	0.008	13 sano	0.838	37 sano	0.989
22	patologia cardiovascolare	0.298	16 sano	0.984	38 sano	0.989
28	**patologia cardiovascolare**	**0.938**	17 sano	0.989	**40 sano**	**0.298**
29	patologia cardiovascolare	0.012	19 sano	0.717	41 sano	0.838
32	**patologia cardiovascolare**	**0.639**	20 sano	0.992	42 sano	0.969
39	patologia cardiovascolare	0.092	21 sano	0.838	43 sano	0.978
47	patologia cardiovascolare	0.378	23 sano	0.984	44 sano	0.969
48	patologia cardiovascolare	0.229	**24 sano**	**0.047**	45 sano	0.913
49	patologia cardiovascolare	0.127	25 sano	0.956	46 sano	0.881
					50 sano	0.938

Conoscendo il valore di ansia di tratto X_i, il modello sottovaluta clamorosamente la probabilità di appartenere alla categoria $Y_{1\ Sano}$ di due soggetti effettivamente sani (righe **24 e 40**), e sbaglia all'opposto affermando una notevole probabilità di appartenere alla categoria Y_1 per tre soggetti che, invece, sono malati (righe 5**, 28** e **32**). Si può pensare

[114] In effetti, la funzione `predict` dà diverse opportunità informative in più, a seconda del tipo di modello cui si applica, e si possono ottenere gli stessi valori di `fitted` specificando l'argomento `type= "response"`; per dettagli, `help("predict")`

che i primi due soggetti siano molto più ansiosi dei loro congeneri, e gli ultimi tre meno ansiosi – il che li rende probabili **outlier bivariati**, o comunque soggetti piuttosto anomali. Possiamo cogliere l'occasione per verificarlo:

```
residui<-rstandard(ansia)
residui[c(5,28,32)]
        5        28        32
-1.630068 -2.394812 -1.470047
residui[c(24,40)]
      24       40
2.541670 1.630479
```

Solo i soggetti 28 e 24 superano le due deviazioni standard convenzionali, in valore assoluto, per cui sono effettivamente considerabili outlier bivariati: il paziente della riga 28 è poco meno della metà ansioso rispetto al proprio gruppo, mentre il soggetto sano della riga 24 è due volte più ansioso rispetto ai compagni della sua categoria. **Nessuno** dei due è però un **outlier univariato**, dato che altri hanno punteggi più alti e più bassi dei loro:

```
which(cuore$ansia==max(cuore$ansia); which(cuore$ansia==min(cuore$ansia))
[1] 11 18
[1] 20
cuore$outcome[c(11,18,20)]
[1] patologia cardiovascolare   patologia cardiovascolare    sano
max(cuore$ansia); min(cuore$ansia)
[1] 36
[1] 9
```

Gli ansiosissimi soggetti nelle righe 11 e 18 sono, infatti, pazienti, e il serafico soggetto 20 è sano: per tutti e tre, il modello è praticamente certo rispetto alla loro probabilità di appartenere alla categoria Y_1:

	outcome	fitted		outcome	fitted
11	patologia cardiovascolare	0.008292225	20	sano	0.992261830
18	patologia cardiovascolare	0.008292225			

Ci sono altri outliers bivariati? Dai `fitted` pare di no:

```
which(abs(residui)>= 2)
24 28
24 28
```

Per sapere se gli outlier bivariati sono **casi influenti**, possiamo conoscere il loro valore di leverage con gli **hatvalues** o la distanza di **Cook**: `hatvalues(modello)` o `cooks.distance(modello)`. Per esempio:

```
summary(cooks.distance(solo_ansia))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0000463 0.0002436 0.0014690 0.0263600 0.0044990 0.6355000
```

Oppure:

```
cooks.distance(ansia)[c(24,28)]
       24        28
0.6354717 0.2555228
```

Nessun caso influente, quindi.

Le **assunzioni della regressione logistica** ignorano quelle espressamente richieste dal metodo dei minimi quadrati, ovvero la distribuzione normale degli errori, la loro omoschedasticità, e, ovviamente, la natura metrica di Y. Restano però alcune cose da verificare:

- l'assunzione della linearità della relazione tra $Y \sim X$, la cui impossibilità è proprio il motivo per cui usiamo il $logit$ del dato, si traduce nella presenza di una **relazione lineare tra i predittori X (se continui) e il $logit$ dell'outcome Y**;
- esigenza dell'**indipendenza** delle misure – e quindi degli errori;
- nel caso della regressione logistica multipla, l'assenza di **multicollinearità** tra le X.

La linearità della relazione $X - logit_Y$ si può controllare verificando l'esistenza di una interazione tra il predittore e la sua trasformazione logaritmica (metodo di Hosmer e Lemeshow, 1989): se non è significativa, la linearità della relazione è garantita.

```
cuore$log_ansia<-log(cuore$ansia)
summary(glm(data=cuore, formula=outcome~ansia*log_ansia, family=binomial))
[…]
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)      13.510    133.844   0.101    0.920
ansia            -5.301     28.260  -0.188    0.851
log_ansia         8.995    122.668   0.073    0.942
ansia:log_ansia   1.104      5.519   0.200    0.841
```

La linearità è verificata.

L'**indipendenza degli errori** dovrebbe essere garantita se non ricaviamo i dati dagli stessi soggetti (motivo per cui **non si usa questo tipo di regressione per misure ripetute**). Verifichiamo, comunque, che nel modello `ansia` gli errori siano indipendenti:

```
DurbinWatsonTest(ansia)
       Durbin-Watson test
data:  ansia
DW = 2.0487, p-value = 0.5635
alternative hypothesis: true autocorrelation is greater than 0
```

Se abbiamo più predittori, la **multicollinearità** si verifica nuovamente con la statistica VIF, ovvero il suo inverso $1/VIF$ (tolleranza): nel caso in cui sia inaccettabilmente elevata, le scelte non sono molte. Dato che un modello con questi predittori non può essere altro che inattendibile, uno dei predittori problematici deve essere eliminato – ma la statistica non può essere di grande d'aiuto, dato che ai suoi occhi hanno identica "colpa"; è anche possibile sostituire il predittore prescelto per l'eliminazione con altro predittore ugualmente rilevante, non troppo correlato con gli altri (Bowerman e O'Connelli, 1990). In alternativa, si può eseguire are **un'analisi fattoriale** (procedura di **riduzione**, che semplifica una matrice di correlazione assegnando le variabili a fattori latenti che le sintetizzano) e usare i **punteggi fattoriali** invece delle variabili.

Nel modello `ansia_eta`, verifichiamo con `VIF` di `DescTools`:

```
VIF(ansia_eta)
          GVIF Df GVIF^(1/(2*Df))
ansia 1.044855  1        1.022181
eta   1.044855  2        1.011030
1/VIF(ansia_eta)
           GVIF  Df GVIF^(1/(2*Df))
ansia 0.9570706 1.0       0.9782999
eta   0.9570706 0.5       0.9890904
```

$GVIF$[115] è il fattore di inflazione della varianza generalizzato (Fox e Monette, 1992), ovvero un VIF corretto per i df del predittore secondo la formula: $GVIF = VIF^{1/(2df)}$ riportata nell'output; si interpreta quando X ha almeno $df = 2$.

Un ulteriore aspetto importante per l'affidabilità del risultato è la **dimensione campionaria**: servono molti soggetti, indicazione che si concretizza in un suggerimento di massima che prevede un **rapporto di almeno 10 soggetti che presentano l'outcome meno frequente per ciascun predittore** considerato nell'analisi. Ad esempio, con tre X ("fumo", "calorie", "esercizio fisico"), un outcome Y_1 con probabilità $p_1 = .10$ (ad esempio, "malato") e un outcome Y_0 con probabilità $p_0 = .90$ ("sano"), il campione dovrebbe essere composto da **almeno** $(10 \times 3)/.10 = 300$ soggetti. Naturalmente, se Y_1 fosse più probabile, la numerosità campionaria diminuirebbe, come abbiamo visto succedere molte

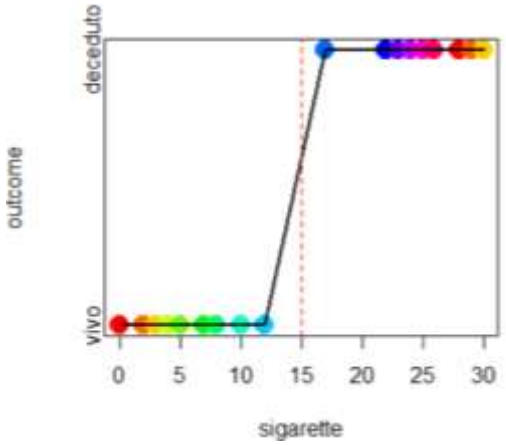
[115] Per i precisini: VIF indica la penalizzazione nel fit del modello quando si aggiunge un ulteriore predittore non ortogonale, mentre $GVIF$ indica la penalizzazione nel fit quando si aggiunge un subset di variabili non ortogonali… ma possiamo fermarci qui.

volte nel rapporto tra potenza-effect size-numerosità campionaria. Se la probabilità di essere "malato" fosse $p_1 = .50$, il campione minimo sarebbe di "soli" $(10 \times 3)/.50 = 60$ soggetti.

Talvolta, le cose nel modello **vanno così male che R non produce proprio un modello**. Questo si verifica per tutte le analisi che usano il metodo della massima verosimiglianza: abbiamo già anticipato che è un **metodo iterativo**, in cui i parametri del modello sono stimati per **approssimazioni successive**, per cui si parte da una prima stima (i "semi" delle stime sono diversi da analisi ad analisi), poi si procede per approssimazioni successive (**iterazioni**) finché le approssimazioni originano coefficienti identici (**la soluzione converge**), o finché si è raggiunto il numero massimo di iterazioni stabilito a priori (varie centinaia, di default). In **due situazioni**, R rallenta estremamente (o si blocca proprio) e alla fine rinuncia, producendo un output "qualsiasi" con un rassegnato messaggio: `Warning message: glm.fit: fitted probabilities numerically 0 or 1 occurred`. Qualsiasi interpretazione dei coefficienti riportati nell'output sarà del tutto inaffidabile.

1) Separazione completa: si verifica, paradossalmente, quando $\boldsymbol{Y}$ **può essere perfettamente predetta da** $\boldsymbol{X}$ (o da una combinazione di X), **senza errori** nel modello. In questo caso, il più grave, si aggiunge un ulteriore warning: `glm.fit: algorithm did not converge`. R non ha trovato una soluzione al modello, non ha raggiunto la convergenza. Per esempio, vediamo un caso estremo della relazione tra numero di sigarette fumate al giorno (X) e outcome ($Y_{0\,Vivo}$, $Y_{1\,Deceduto}$) con cui abbiamo aperto il capitolo:

```
sigarette<-c(0,2,3,4,5,7,8,10,12,17,22,23,24,25,26,28,29,30)
outcome<-c(0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1)
summary(glm(outcome~sigarette, family=binomial))
[…]
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -128.903 168801.052  -0.001    0.999
sigarette       8.891  11517.179   0.001    0.999
…
Null deviance: 2.4953e+01  on 17  degrees of freedom
Residual deviance: 8.8797e-10  on 16  degrees of freedom
AIC: 4
[…]
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1
occurred
```

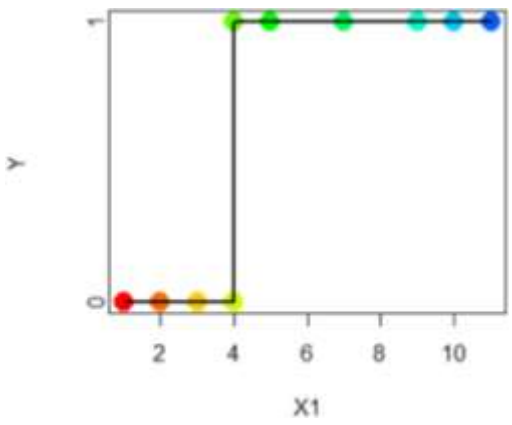



```
plot(sigarette, outcome, pch=19, cex=2, col=rainbow(15), yaxt="n");
lines(sigarette,runmed(outcome, k = 3), lwd=2)
abline(v=15, lty=2, col="red"); mtext(text=c("vivo", "deceduto"), side=2, at=c(0,1))
```

Nessuno tra i vivi ha fumato più di 12 sigarette, nessuno tra i deceduti ha fumato meno di 17 sigarette: conoscendo il numero di sigarette fumate, il modello ha la certezza di predire lo stato Y_0 o Y_1 di ogni caso. L'algoritmo del modello non converge e ci viene spiegato perché – notate, comunque, anche **l'errore standard spaventosamente grande**, che dovrebbe mettere sull'avviso.

2) Separazione quasi completa: è un caso meno estremo, in cui X (o una combinazione di X) consente la perfetta predizione della categoria Y_1 per la **maggior parte dei valori di** $\boldsymbol{X}$ – ma non **tutti**. In questo caso R avvisa solo con `fitted probabilities numerically 0 or 1 occurred`.
Per esempio:

```
Y<-c(0,0,0,0,1,1,1,1,1,1)
X1<-c(1,2,3,4,4,5,7,9,10,11)
summary(glm(Y~X1, family = binomial))

[…]
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -80.18   38637.39  -0.002    0.998
X1             20.04    9659.35   0.002    0.998
…
    Null deviance: 13.4602  on 9  degrees of freedom
Residual deviance:  2.7726  on 8  degrees of freedom
AIC: 6.7726
…
Warning message:
glm.fit:  fitted  probabilities  numerically  0  or  1
occurred
plot(X1, Y, pch=19, col=rainbow(15), cex=2)
lines(X1,runmed(Y, k = 3), lwd=2)
```



Un **ulteriore problema**, che purtroppo non viene segnalato da un *warning*, è l'**informazione incompleta nei predittori**: in questo caso, mancano del tutto le osservazioni in un livello o in una combinazione di livelli, rendendo impossibile predire correttamente la Y per quel livello / combinazione di livelli:

X_1	X_2	X_3		Hai frequentato?	Hai fatto esercizi?	Esame superato?
Sì	No	Sì		Sì	No	Sì
Sì	Sì	Sì	→	Sì	Sì	Sì
No	No	Sì		No	No	Sì
No	Sì	**???**		No	Sì	**???**

Se **non** si raccolgono soggetti che **non hanno frequentato** (X_1) e **hanno fatto esercizi** (X_2), non possiamo sapere se questa combinazione di predittori ha consentito di superare o meno l'esame (X_3). In questo caso, in R la soluzione converge e non veniamo avvisati del problema, ma **l'errore standard** associato ai coefficienti del modello è **enorme**. Non accorgersene, o non vericare questa condizione con una semplice tabella di contingenza dei predittori, porta a interpretazioni del modello molto, molto infelici. Naturalmente, la soluzione potrebbe essere raccogliere più dati, se possibile. Vediamo un esempio:

```
frequentato<-as.factor(c(rep("sì", 12), rep("no", 12)))
esercizi<-as.factor(c(rep("sì",12),rep("no",6), rep("sì", 6)))
esame<-as.factor(c(rep("superato",12), rep("non superato",12)))
table(frequentato, esercizi)
              esercizi
               no   sì
frequentato no   6    6
            sì   0    1

summary(glm(esame~frequentato+esercizi, family=binomial))
[…]
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      19.57    4390.31   0.004    0.996
frequentatosì    19.57    4390.31   0.004    0.996
esercizisì      -39.13    6208.83  -0.006    0.995
```

14.2 Regressione logistica multinomiale

La regressione logistica multinomiale valuta la **predicibilità** di **una** variabile Y **categoriale politomica** (o multinomiale: **più di due categorie**) a partire dai valori di **almeno una** variabile X, indifferentemente **continua o categoriale**.

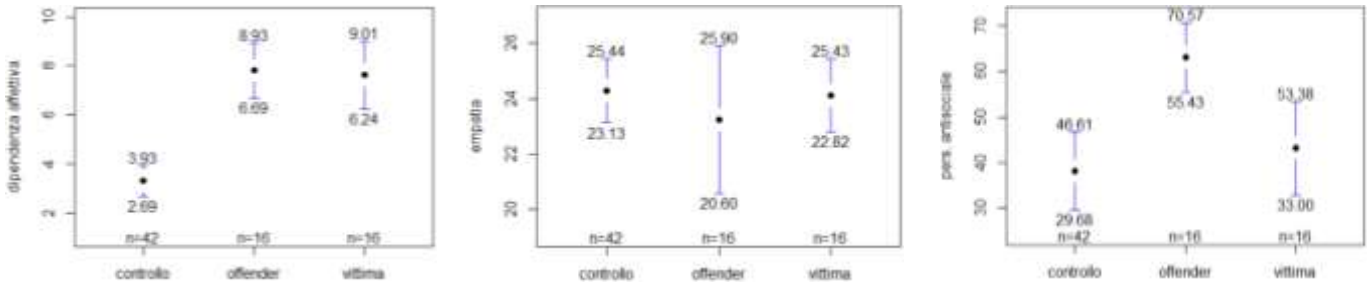
L'ipotesi alternativa è che, conoscendo il valore x_j assunto in X dal soggetto j, sia possibile predire la **probabilità che il soggetto appartenga a una data categoria Y_{j1} rispetto a un'altra categoria Y_{j2}**; H_0 è che, essendo Y e X indipendenti in popolazione, conoscendo il valore x_j assunto in X dal soggetto j, **non** sia possibile predire la **probabilità che il soggetto appartenga a una data categoria Y_{j1} rispetto a un'altra categoria Y_{j2}**.

La logica della regressione logistica multinomiale è identica a quella della regressione logistica binaria, a parte il fatto che è organizzata in una serie di **confronti a coppie** tra la **categoria che viene scelta come riferimento** (**baseline**) e **ciascuna delle altre**.

In R useremo il package **`mlogit`**, ricordandoci di passare prima dalla trasformazione del dataframe.

Vediamolo con un esempio, tornando a utilizzare il dataframe `coppie` (rinominiamolo `cp`): conoscendo la **dipendenza affettiva** dei soggetti (X_1, `$love_addiction_risposte_si`), la loro **empatia** (X_2, `$empatia_risposte_esatte`) e la loro elevazione nella **personalità antisociale** (X_3, `$millon_personalita_antisociale`), **singolarmente** (prima) e in **combinazione** (poi), possiamo **predire la probabilità** che il soggetto appartenga alla categoria di riferimento **Controlli** (**1**), invece che alla categoria **Offender** (**2**) e alla categoria **Vittime** (**3**)? I tre predittori sono ugualmente efficaci (o inefficaci)?

Ricordiamo le distribuzioni nei gruppi:



Le tre X sembrano promettere risultati parecchio differenti.

Il dataframe `offenders` è ricco di variabili: estraiamo solo la Y e le X che ci servono, per facilitare la comprensione della sua trasformazione, creando il dataframe **`multi`**:

```
multi<-data.frame(sogg=cp$soggetto, gruppo=cp$gruppo, empatia = cp$empatia_risposte_esatte,
love_addiction = cp$love_addiction_risposte_si, antisociale = cp$millon_personalita_antisociale)
head(multi)
```

```
  sogg    gruppo empatia love_addiction antisociale
1   S1 controllo      28              4          45
2   S2 controllo      25              4          12
3   S3 controllo      20              2          38
4   S4 controllo      24              6          64
5   S5 controllo      26              7          75
6   S6 controllo      26              5          24
```

Cambiamo il formato con **`mlogit.data(data= dataframe, choice, shape)`**: ogni controllo sarà ripetuto in una riga `TRUE` e due `FALSE` (non-offender e non-vittima), come ogni offender (`TRUE` per offender, `FALSE` per controllo e per vittima) e vittima (`TRUE` per vittima e `FALSE` per controllo e per offender).

```
multi_log<-mlogit.data(data = multi, choice = "gruppo", shape = "wide")

multi_log[1:3,]
```

```
  sogg gruppo empatia love_addiction antisociale chid       alt
1   S1   TRUE      28              4          45    1 controllo
2   S1  FALSE      28              4          45    1  offender
3   S1  FALSE      28              4          45    1   vittima
```

```
multi_log[130:132,]
  sogg gruppo empatia love_addiction antisociale chid      alt
1  S44  FALSE      18              6          67   44 controllo
2  S44   TRUE      18              6          67   44  offender
3  S44  FALSE      18              6          67   44   vittima
multi_log[178:180,]
  sogg gruppo empatia love_addiction antisociale chid      alt
1  S60  FALSE      24              6          60   60 controllo
2  S60  FALSE      24              6          60   60  offender
3  S60   TRUE      24              6          60   60   vittima
```

Cominciamo l'analisi con il solo predittore **dipendenza affettiva**. Applichiamo `mlogit(formula= Y~b0|X1, dataframe)`, ricordando di esprimere l'intercetta (`Y~1`). La categoria di riferimento, per default, è la prima (`=1`, come al solito secondo l'ordine alfanumerico): nell'esempio, `Controlli` va bene per le ipotesi, altrimenti la cambieremmo con l'argomento `reflevel= nuovo livello di riferimento`.

```
dipendenza<-mlogit(formula = gruppo~1|love_addiction, data= multi_log)
```

L'output del `summary` del modello `dipendenza` è lungo: spezziamolo partendo dal fondo, cioè dal modello *overall*:

```
summary(dipendenza)
```

```
[...]
Log-Likelihood: -47.583
McFadden R^2:  0.34635
Likelihood ratio test : chisq = 50.426 (p.value = 1.1226e-11)
```

Fin qui nulla di nuovo: secondo il ***LRT*** la relazione è **significativa**, ovvero la dipendenza affettiva ha un effetto sulla probabilità di appartenere al gruppo Controllo invece che al gruppo Offender o Vittima. L'R_L^2 indica, però, una relazione abbastanza modesta. Possiamo verificare che LRT sia lo stesso di quello eseguito nella regressione logistica binaria sfruttando la LL, che è uno degli elementi della lista del modello creato con `mlogit`: dobbiamo naturalmente trasformarla in $-2LL$ moltiplicandola per -2

```
nullo_dipe<-mlogit(gruppo~1, data = multi_log)
nullo_dipe$logLik*-2
'log Lik.' 145.5917 (df=2)
```

→ $df = 2$ perché il modello nullo contiene **due b_0**: vedi la sezione dei coefficienti

```
print(LRT<-(nullo_dipe$logLik*-2)-(love$logLik*-2))
'log Lik.' 50.42552 (df=2)
```

→ $df_{love} = 4$ [2 b_0, b_1 contrasto I, b_1 contrasto II) - $df_{nullo} = 2$

```
pchisq(LRT, 2, lower.tail = FALSE)
'log Lik.' 1.122634e-11 (df=2)
```

Torniamo alla parte superiore del summary: dopo le **frequenze relative** osservate nei gruppi:

```
 Frequencies of alternatives:
 controllo  offender   vittima
   0.56757   0.21622   0.21622
```

←

```
prop.table(table(offenders$gruppo))
controllo  offender   vittima
0.5675676 0.2162162 0.2162162
```

… troviamo i coefficienti di ogni contrasto. Le intercette sono **due**, una per contrasto: ciascuna b_0, infatti, rappresenta il **valore del logit quando la $X = 0$, per ogni confronto tra livello di riferimento e gli altri**.

```
Coefficients :
                       Estimate Std. Error z-value  Pr(>|z|)
offender:(intercept)    -5.96251    1.29571 -4.6017 4.190e-06
vittima:(intercept)     -5.71431    1.25485 -4.5538 5.269e-06
```

Possiamo comunque ignorare le intercette e concentrarci sui b_1 dei due contrasti.

```
                        Estimate Std. Error z-value  Pr(>|z|)
offender:love_addiction  0.92853    0.21422  4.3345 1.461e-05 → Controlli vs offenders
vittima:love_addiction   0.89638    0.21050  4.2584 2.059e-05 → Controlli vs vittima
```

All'aumentare dei punteggi di dipendenza affettiva, **aumenta** significativamente (+.982) la **probabilità di appartenere alla categoria Offender** invece che alla categoria di riferimento **Controllo**. All'aumentare dei punteggi di dipendenza affettiva, **aumenta significativamente (+.896) la probabilità** di **appartenere alla categoria Vittime**.

Gli ***OR*** ci dicono **di quanto** aumentano le due probabilità:

```
exp(0.92853)
[1] 2.530786
exp(0.89638)
[1] 2.450715
```

All'aumentare della dipendenza affettiva, la probabilità di essere offender o di essere vittima è di circa 2.5 volte maggiore della probabilità di essere un soggetto di controllo.

Se avessimo usato come categoria di riferimento Vittime:

```
dipendenza_2<-mlogit(formula = gruppo~1|love_addiction, data= multi_log, reflevel = 3)
```

avremmo ottenuto un modello *overall* ovviamente identico:

```
Log-Likelihood: -47.583
McFadden R^2:  0.34635
Likelihood ratio test : chisq = 50.426 (p.value = 1.1226e-11)
```

ma contrasti diversi:

```
                        Estimate Std. Error z-value  Pr(>|z|)
controllo:(intercept)    5.714315   1.254847  4.5538 5.269e-06
offender:(intercept)    -0.248193   1.185815 -0.2093    0.8342

controllo:love_addiction -0.896379   0.210497 -4.2584 2.059e-05 → Vittime vs controlli
offender:love_addiction   0.032156   0.146644  0.2193    0.8264 → Vittime vs Offenders
```

All'aumentare dei punteggi di dipendenza affettiva, **diminuisce** (−.896) la **probabilità di appartenere alla categoria Controlli** invece che alla categoria di riferimento **Vittime** (lo sapevamo già). All'aumentare dei punteggi di dipendenza affettiva, **resta immutata (0.03) la probabilità** di **appartenere alla categoria Offenders invece che alla categoria Vittime**.

Prima di passare ai modelli con più predittori, verificate se l'empatia, singolarmente presa, e la personalità antisociale, singolarmente presa, predicono in maniera significativa la probabilità di appartenere al gruppo Controllo invece che al gruppo Offender e al gruppo Vittima.

L'interpretazione dei contrasti è un po' più complicata nei modelli con interazione tra le X. Vediamo il modello `dipe_antisoc` in cui, oltre agli effetti principali di **dipendenza affettiva** e di **personalità antisociale**, verifichiamo l'effetto della loro **interazione**: la probabilità di essere controllo invece che offender e vittima cambia in maniera diversa per chi ha alta o bassa dipendenza affettiva, a seconda della gravità delle loro tendenze antisociali [oppure: cambia in maniera diversa per chi ha una personalità più o meno antisociale, a seconda del loro punteggio in dipendenza affettiva]?

```
dipe_antisoc<-mlogit(formula=gruppo~1|love_addiction*antisociale,data=multi_log, reflevel= 1)
summary(dipe_antisoc)
```

Vediamo prima il **modello overall**:

```
Log-Likelihood: -39.545
McFadden R^2:  0.45677
Likelihood ratio test : chisq = 66.501 (p.value = 2.1285e-12)
```

I due predittori singolarmente e la loro interazione hanno un effetto complessivamente significativo, di intensità discreta sulla probabilità di appartenenza a un gruppo.
La sezione dei parametri del modello è lunga: separiamola per contenuti.

Prima le intercette:

```
Coefficients :
                            Estimate Std. Error z-value Pr(>|z|)
offender:(intercept)      -12.141468   5.930670 -2.0472 0.040635
vittima:(intercept)       -16.334878   6.136718 -2.6618 0.007772
```

Entrambe le b_0 sono significativamente diverse da 0.
Ora gli effetti principali: prima l'effetto principale della dipendenza affettiva:

```
                            Estimate Std. Error z-value Pr(>|z|)
offender:love_addiction     2.117689   1.134047  1.8674 0.061849
vittima:love_addiction      3.228920   1.198680  2.6937 0.007066
```

```
exp(2.117689)
[1] 8.311906
exp(3.228920)
[1] 25.25237
```

La probabilità essere offender (a soglia, ma con un buon OR: 8.3) e quella di essere vittima invece che controllo (grande OR: 25.2) crescono significativamente all'aumentare della dipendenza affettiva, indipendentemente dalla personalità antisociale del soggetto.
Ora l'effetto principale della personalità antisociale:

```
                            Estimate Std. Error z-value Pr(>|z|)
offender:antisociale        0.106935   0.091462  1.1692 0.242331
vittima:antisociale         0.184702   0.094607  1.9523 0.050902
```

```
exp(0.184702)
[1] 1.20286
```

La probabilità di essere offender invece che controllo è indipendente dall'elevazione della personalità antisociale; quella di essere vittima invece che controllo aumenta significativamente (a soglia: infatti, secondo l'OR, aumenta solo di 1.2 volte) al crescere della personalità antisociale, ignorando l'effetto della dipendenza affettiva del soggetto.

Quindi comincia **l'interazione** dipendenza affettiva X personalità antisociale.
Nel **I contrasto**:

```
                                  Estimate Std. Error z-value Pr(>|z|)
offender:love_addiction:antisociale -0.019822   0.016845 -1.1767 0.239315
```

non c'è un'interazione **significativa** nel predire l'appartenenza al gruppo di Controllo invece che al gruppo Offender: quando la personalità antisociale aumenta, in combinazione con una maggiore dipendenza affettiva, la **probabilità di essere Offender invece che Controllo sembra diminuire di più (b_1 negativo)**, ma in realtà l'effetto non è significativo.
Nel **II contrasto**:

```
                                  Estimate Std. Error z-value Pr(>|z|)
vittima:love_addiction:antisociale -0.040088   0.018077 -2.2176 0.026581
```

```
exp(-0.040088)
[1] 0.9607049
```

C'è un'interazione significativa nel predire l'appartenenza al gruppo di Controllo invece che al gruppo **Vittime**: quando la **personalità antisociale aumenta, in combinazione** con una **maggiore dipendenza affettiva**, la **probabilità di essere Vittima invece che Controllo diminuisce** significativamente **di più** (**b_1 negativo**), ma non di molto (OR= .96).

Capitolo 15
Multilevel linear models o linear mixed models

In questo capitolo useremo il dataframe ***idiomi*** *pubblicato su Elly: apritelo e leggetene la descrizione prima di proseguire. Useremo anche* `disforia` *e* `sicurezza`*, ma questi li conoscete già.*

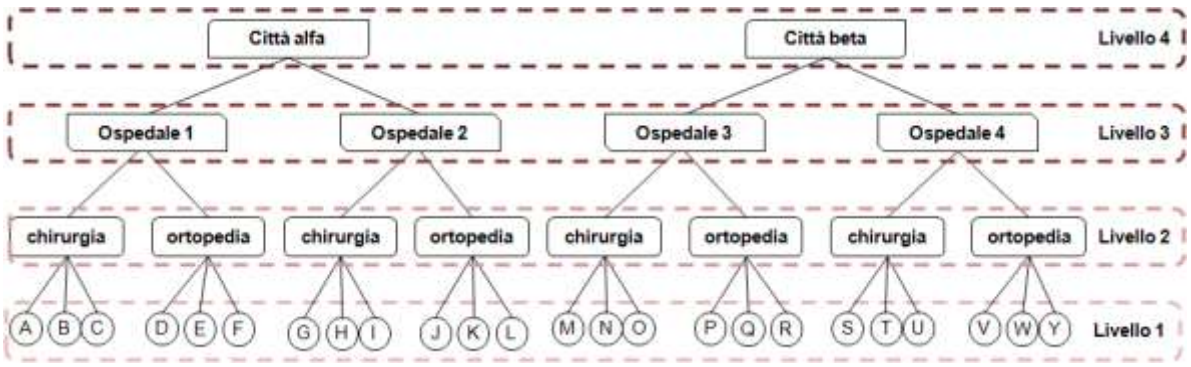
Matricialmente parlando, i multilevel models sono parecchio complessi. In questo capitolo ne assaggiamo solo un po', usando disegni semplici applicabili a un modello ***lineare,*** *che possono essere abbastanza facilmente ampliati a casi più complessi e a modelli* ***non lineari****, per impararne la logica e i processi di base; rimandiamo però a un corso di statistica di livello decisamente più avanzato il trattamento estensivo di questi modelli.*

15.1 ANOVA di I, II e III tipo

L'analisi della varianza fattoriale affrontata nel capitolo 7 ha esplorato casi in cui tutti i livelli di un predittore X_1 incrociano tutti i livelli del predittore X_2 (in generale, di tutti i predittori nel modello): ***crossed designs*** o *crossed experiment*. I *crossed experiment* sono certamente molto diffusi, ma non sono l'unica possibilità di disegno fattoriale, per diversi motivi. Per esempio, i **predittori** nel modello, in molti casi, **non sono indipendenti** tra loro: abbiamo visto in diversi esempi concreti come i predittori Età e Titolo di studio non fossero effettivamente indipendenti, dato che i partecipanti più anziani molto spesso presentavano i livelli di scolarità più bassi. Infine, la verifica delle differenze tra le medie dei livelli non è sempre interessata a confrontare specificamente due o più livelli di X_1, quanto piuttosto all'esistenza di una variabilità fra tutti i livelli di un predittore, di cui quelli previsti nella ricerca costituiscono un campione scelto in maniera casuale. Per esempio, affronteremo tra qualche pagina la **valutazione della comprensione di un brano in bambini di diversa classe** (X_1) appartenenti a **due Istituti scolastici (X_1),** A e B: la ricerca non era interessata alla differenza specifica tra **quei** due Istituti, che costituiscono solo due livelli scelti "casualmente" (beh, all'incirca) della variabile Istituto, i cui livelli sono naturalmente migliaia. In effetti, l'interesse per la variabile Istituto era dovuto solo alla rilevazione della eventuale **variabilità di Y (comprensione del brano)** tra i livelli X_2 (Istituti): per rilevare la variabilità si sono scelti Istituti a caso, e quindi in una successiva ricerca i livelli di $X_{2\,Istituto}$ potrebbero essere differenti, dato che si è interessati a una generica affermazione del tipo: "La comprensione di un testo è significativamente variabile tra gli Istituti di formazione primaria". Questo tipo di disegno vede una strutturazione **gerarchica o nidificata (hierarchical design, nested classification) dei dati**, in cui le variabili sono **raggruppate/ nidificate / nested in altre variabili**, e il **livello** o **cluster** cui appartengono in questa gerarchia ne definisce lo status: il livello superiore influisce su quello minore, che acquista senso quando viene analizzato dentro il primo. Questa strutturazione comporta che i **soggetti appartenenti a un cluster siano più simili tra loro** di quanto siano simili ai soggetti **appartenenti a un altro cluster / livello** della variabile di ordine superiore.

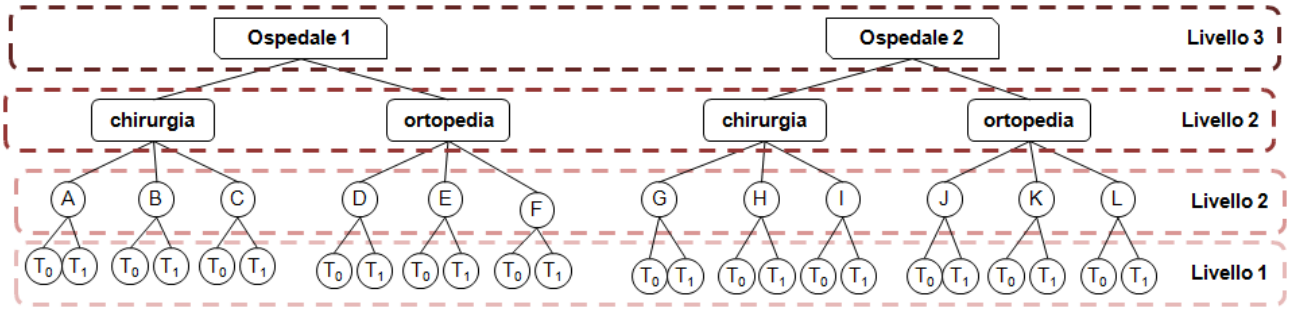
Nel caso dei **disegni between groups**, il livello più basso (**livello 1**) è **rappresentato dal singolo individuo** all'interno del **gruppo**, che è la variabile di **livello 2**, definita **contestuale**; questa, a sua volta, può essere *nested* in una superiore variabile contestuale di **livello 3**… e così via. Un facile esempio, che si trova correntemente nella ricerca psicologica, può essere quello di una ricerca sull'ansia pre-operatoria dei soggetti (livello 1) ricoverati in due diversi reparti (livello 2,

variabile di raggruppamento contestuale), reclutati in diversi istituti ospedalieri (livello 3) di varie città (livello 4) in cinque Regioni diverse (livello 5)....



Si può facilmente ipotizzare che l'ansia dei pazienti che devono sottoporsi a un più leggero intervento nei reparti di ortopedia dell'arto sia minore di quella dei pazienti ricoverati in chirurgia toracica; d'altronde, però, la reputazione complessiva delle équipe chirurgiche degli ospedali può essere diversa; quindi, l'ansia dei pazienti ricoverati nell'ospedale 1, indipendentemente dal loro reparto, può essere minore dell'ansia dei pazienti ricoverati nell'ospedale 2 o 3. Inoltre, le strutture ospedaliera della città alfa fruiscono di servizi di supporto (come lo psicologo di reparto) non presenti nelle strutture della città beta: anche questo elemento può influire sulla misura "ansia".

I modelli multilevel trovano un'applicazione ancora più "naturale" nel caso dei disegni a misure ripetute o misti, in cui non sono gli individui a rappresentare il livello 1, ma le **singole misure (livello 1)**, raggruppate in ogni **individuo**, che ora diventa quindi la **variabile di livello 2**… e che, a sua volta, può naturalmente appartenere a cluster in variabili contestuali di livello superiore. Ad esempio, possiamo misurare la qualità della vita prima e dopo un intervento (T_0-T_1: livello 1) in soggetti (livello 2) ricoverati in due diversi reparti (livello 3) di due ospedali diversi (livello 4), eccetera.



La struttura gerarchica consente di analizzare e confrontare, oltre alle medie marginali di ogni condizione, anche le **varianze** dei vari livelli: l'analisi delle componenti della varianza consente di stimare quanta della variabilità di Y dipenda dalle caratteristiche del livello. Combinandosi con le tecniche di verifica di ipotesi sulle medie marginali, si sono così strutturati **tre diversi tipi di ANOVA**:

1. $\boldsymbol{ANOVA\ I}$ o a effetti fissi (**fixed effect ANOVA**): le ipotesi di interesse riguardano le medie marginali delle condizioni sperimentali: è affine a quella che abbiamo affrontato nel capitolo 7, su disegni *crossed*), ma calata in disegni *nested*. La significatività di ogni predittore è verificata dal rapporto F tra la SS_M di una X e la SS_M del livello immediatamente inferiore.
2. $\boldsymbol{ANOVA\ II}$ o a effetti casuali (**random effects ANOVA**): verifica **ipotesi sulla varianza**, con lo scopo di valutare se sia diversa nei diversi livelli del disegno; i livelli della variabile contestuale sono scelti in maniera casuale tra i molti possibili, e possono essere sostituiti (***repleaceability***) da altri livelli senza che questo incida negativamente sulla generalizzabilità delle conclusioni (***generalization***) tratte dall'analisi. Perché questo avvenga, però, gli effetti

random e gli errori devono essere normalmente distribuiti e indipendenti gli uni dagli altri; come sempre, gli errori devono essere indipendenti fra loro.

3. $\boldsymbol{ANOVA\ III}$ o a **effetti misti** (**mixed model** – può essere **linear** o **generalized - ANOVA**): combina la verifica sulle medie delle condizioni di X (**fixed effects**) con la stima della variabilità di Y nei diversi livelli della variabile di contesto (**random effects**).

È quindi dell'$\boldsymbol{ANOVA\ III}$ **che ci occupiamo in questo capitolo**.

15.2 I parametri del mixed model

Dal punto di vista teorico, il motivo più importante per trattare i dati in analisi come effettivamente gerarchici è il **problema dell'indipendenza delle misure**: le variabili contestuali introducono **dipendenza**, ovvero rendono più probabile che i residui del modello siano correlati. Sappiamo bene che l'indipendenza degli errori è un essenziale requisito dei modelli lineari: quando le entità misurate sono estratte dallo stesso contesto, è più improbabile che il requisito possa essere vero, tanto che è facilmente opinabile che l'ANOVA a misure ripetute "tradizionale" possa essere realmente libera dal *bias* della dipendenza. La dipendenza può essere quantificata dal **coefficiente di correlazione intraclasse** ($\boldsymbol{ICC}$), che **rappresenta la proporzione di variabilità totale della misura attribuita alla variabile contestuale**. Se essa ha un forte effetto, la variabilità sarà piccola e l'ICC ampio; se ha un effetto debole, la variabilità sarà ampia e l'ICC piccolo – cosa che auspichiamo. Uno dei principali vantaggi dei mixed models è proprio che si può **ignorare l'assunzione di indipendenza delle misure**, dato che sono "fatti apposta" per adattare al modello le relazioni tra i casi.
Inoltre, i non richiedono dataset completi, il che è particolarmente utile per disegni a misure ripetute con drop out: nei modelli non multilevel, anche una sola omissione in una delle misure comporterebbe l'esclusione del caso, che invece viene mantenuto nel modello gerarchico. Infine, i modelli lineari assumono che l'effetto di X su Y (b_1) sia lo stesso per tutti i dati di un livello, ma i mixed models possono esplicitamente **inserire nel modello la variabilità dei $\boldsymbol{b_1}$**. Vediamo come.

Nei mixed models distinguiamo i coefficienti fissi dai coefficienti random:

- **fixed coefficient:** il coefficiente di un modello è **fisso** quando si assume che il modello da questi determinato sia **valido per l'intero campione**: per ogni osservazione è possibile ricavare in maniera affidabile un $\hat{y}$ da un'unica b_0 e un unico b_1. I coefficienti dei modelli lineari visti finora sono fissi: una sola b_0, un solo b_1 che esprime un ugual effetto di X per tutti i casi in analisi.
- **random coefficient:** il coefficiente di un modello è **random** quando si assume che il suo valore possa **variare nel campione**: come vedremo, possiamo impostare come random sia b_0 sia b_1.

*Attenzione, quindi, a **non confondere** coefficienti fissi e random con gli **effetti** fissi e random! Ricordiamo:*

- ***fixed effect**: un effetto è detto **fisso** quando **tutte le possibili condizioni** del predittore X **sono presenti** nell'esperimento (ad esempio, l'essere già stati operati o essere ancora in attesa).*
- ***random effect**: un effetto è detto **random** quando nella ricerca si usa un **campione random di tutte le possibili condizioni sperimentali** di X (ad esempio, abbiamo reclutato pazienti in dieci ospedali su una popolazione di varie centinaia di ospedali).*

Facciamo l'esempio di una sola X a misure ripetute: ciò che rende variabile Y è da un lato l'effetto di X, dall'altro le differenze individuali entro i soggetti (SS_R), più le **differenze individuali rilevate tra i soggetti**, quelle la cui somma avevamo definito (capitolo 6) SS tra i soggetti, o SS_B. Nei disegni a misure ripetute, quello che di solito interessa è la variazione **intraindividuale** da una condizione di X all'altra, determinata da X o dall'errore, mentre la variabilità espressa da SS_B non è rilevante per la ricerca, visto che i soggetti dovrebbero essere scelti in maniera casuale dalla popolazione e quindi essere solo casualmente dissimili tra loro. Tuttavia, le differenze tra i soggetti rientrano nell'analisi se consideriamo il fatto che le **misure di uno stesso soggetto sono tra loro più simili di quanto siano simili i punteggi tra soggetti diversi**, ovvero che le misure all'interno di un cluster della variabile di livello 2 – Soggetto sono tra loro più simili rispetto alle misure all'interno di un altro cluster di livello 2 – Soggetto: questo crea dipendenza tra gli errori del modello. Lo stesso discorso si applica, naturalmente, a disegni between groups: soggetti appartenenti alla stessa classe – cluster Livello 2 sono tra loro più simili rispetto a soggetti di un'altra classe – cluster Livello 2. Meglio, allora, costruire un modello che consideri, **quantificandola, anche la variabilità tra i soggetti**, attraverso un coefficiente chiamato u_i: il modello lineare diventa:

più un coefficiente unico per ogni soggetto che esprime la variabilità del soggetto attorno alla $grand\ mean$

Il valore in Y del soggetto i → $y_i = (b_0 + u_0) + b_1 X_{ij} + e_i$ ← *più l'errore del modello per il soggetto i*

*è dato dalla $grand\ mean$ della popolazione cui appartiene, **costante** per tutti i soggetti*

*più l'effetto del predittore X, **costante** per tutti i soggetti*

Naturalmente, se tutti i soggetti mostrassero oscillazioni attorno alla $grand\ mean = 0$ o comunque molto piccole, cioè se fossero tutti molto simili tra loro indipendentemente dall'effetto di X, allora i coefficienti unici u_{0i} sarebbero pari o prossimi a zero, e quindi i modelli senza u_{01} e con u_{0i} sarebbero equivalenti: aggiungere un termine che cattura la variabilità tra i soggetti attorno alla $grand\ mean$ non migliorerebbe il fit del modello. Al contrario, **se le differenze tra i soggetti attorno alla grand mean fossero rilevanti**, la diffe**renza tra il modello senza u_{0i} e con u_{0i} sarebbe altrettanto sensibile:** l'inserimento del coefficiente u_{0i} **migliorerebbe l'adattamento del modello** ai dati, perché rileverebbe la forte variabilità dei soggetti attorno alla $grand\ mean$.

Perché il coefficiente u_0 viene definito **random**? Perché se i soggetti sono selezionati casualmente (random) da una popolazione, allora i coefficienti u_0 associati a ciascuno di loro variano altrettanto casualmente nel campione, cioè **sono un campione casuale dei coefficienti u_0 in popolazione**. Gli altri due termini del modello, b_0 e b_1, sono invece definiti **fissi**, perché uguali per tutti i soggetti.

Continuiamo con il coefficiente angolare. Abbiamo ricordato l'**assunto** dei modelli lineari secondo cui **l'effetto del predittore è lo stesso per tutti i soggetti** (capitolo 3): le differenze tra gli effetti per tutti i soggetti sottoposti a uno stesso livello di X sarebbero prossime a zero (nel caso di un disegno a misure ripetute, questo vuol dire che l'effetto di X è uguale per tutti i soggetti). Tuttavia, come l'indipendenza delle misure anche questo assunto è spesso disatteso nella realtà: l'effetto di X può essere mediato da molte covariate, non tutte prevedibili e controllabili, che rendono l'impatto del predittore più forte in un cluster (i singoli soggetti nei disegni a misure ripetute, i gruppi nei disegni between groups) e più debole in un altro. Allora, è possibile aggiungere al modello un coefficiente che catturi, **quantificandola**, la **variabilità dell'effetto di X**, cioè la variabilità del coefficiente angolare b_1 nei cluster, chiamato u_{1i}: il modello lineare diventa allora:

*più l'effetto del predittore X, **costante** per tutti i soggetti*

Il valore in Y del soggetto i → $y_i = b_0 + (b_1 X_{ij} + u_1 X_{ij}) + e_i$ ← *più l'errore del modello per il soggetto i*

↑ *è dato dalla $grand\ mean$ della popolazione cui appartiene, **costante** per tutti i soggetti*

↑ *più un coefficiente unico per ogni cluster che esprime la variabilità del b_1, cioè dell'effetto di X*

Come nel caso precedente, se l'effetto di X fosse costante nei cluster del disegno (singoli casi o gruppi che siano), allora u_{1i} sarebbe prossimo a zero, e l'inserimento di questo coefficiente nel modello non ne migliorerebbe il fit rispetto al modello con b_1 fisso. Se invece l'effetto di X fosse sensibilmente diverso nei cluster, allora questo modello, che ne quantifica la variabilità, si adatterebbe decisamente meglio ai dati. Di nuovo come nel caso precedente, l'etichetta "random" assegnata a questo termine del modello è riferita al fatto che i coefficienti u_{1i} sono considerati un campione casuale degli u_{1i} riscontrabili in popolazione.

Infine, se si ritiene che nei dati sia presente una forte variabilità tra i soggetti attorno alla grand mean quanto una forte variabilità dell'effetto di X nei cluster, si possono inserire entrambi i termini u_{0i} e u_{1i} nel modello:

più un coefficiente unico per ogni soggetto che esprime la sua variabilità attorno alla $grand\ mean$ ↓ — *più un coefficiente unico per ogni cluster che esprime la variabilità del b_1* ↓

Il valore in Y del soggetto i → $y_i = (b_0 + u_0) + (b_1 X_{ij} + u_1 X_{ij}) + e_i$ ← *più l'errore per il soggetto i*

↑ *è dato dalla $grand\ mean$ della popolazione cui appartiene, **costante** per tutti i soggetti*

↑ *più l'effetto del predittore X, **costante** per tutti i soggetti*

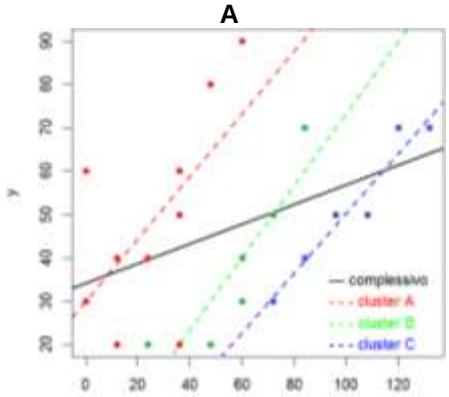
Proviamo a **rappresentare graficamente** le tre possibili situazioni, usando distribuzioni Y e X inventate.

Nel caso A, il modello lineare $Y \sim X$ condotto separatamente nei tre livelli della variabile cluster, mostra slope estremamente simili (l'effetto di X sembra lo stesso nei tre gruppi), ma b_0 molto diverse (i soggetti sono molto variabili rispetto alla $grand\ mean$, indipendentemente da X): un modello con b_0 random e b_1 fisso potrebbe, quindi, meglio adattarsi ai dati rispetto al modello con b_0 e b_1 fissi (in nero).

Nel caso B, i tre modelli condotti nel subset evidenziano b_0 non troppo diverse (i soggetti non sono molto variabili rispetto alla grand mean), b_1 di entità e addirittura direzione diversa: evidentemente, l'effetto di X è molto variabile tra i cluster (nella covariata, avremmo mancato al requisito di equivalenza degli slope). In questa situazione, un modello con b_1 random probabilmente avrebbe un fit migliore del modello con b_1 fisso.

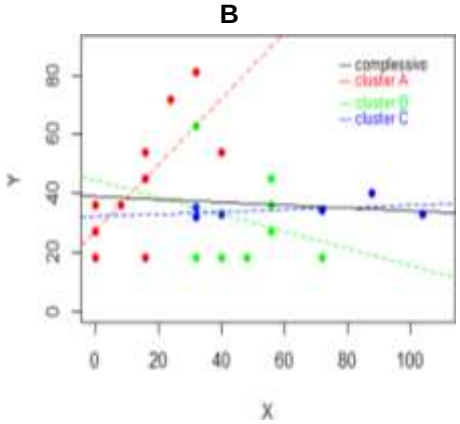
Infine, nel caso C sia gli slope sia le intercette sono molto variabili tra i tre cluster: il modello con soli effetti fissi è molto probabilmente del tutto inadeguato, rispetto a un modello che comprenda anche u_{0i} e u_{1i}.

A



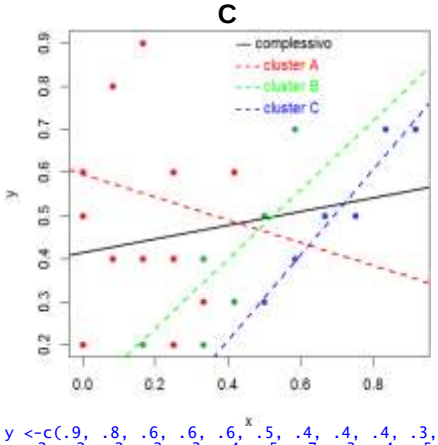

```
y<-c (60, 60, 80, 90, 50, 40, 30, 40, 20, 20, 20, 20, 30, 40, 50, 70, 30, 40, 50, 50, 70, 70)
x <-c(0, 36, 48, 60, 36, 12, 0, 24, 12, 36, 24, 48, 60, 60, 72, 84, 72, 84, 96, 108, 120, 132)
```

B




```
y<-c(81, 72, 54, 54, 45, 36, 36, 27, 18, 18, 18, 18, 18, 18, 27, 36, 45, 63, 33, 35, 32, 34, 40, 33)
x<-c(32, 24, 16, 40, 16, 0, 8, 0, 0, 16, 32, 40, 48, 72, 56, 56, 56, 32, 40, 32, 32, 72, 88, 104)
```

C



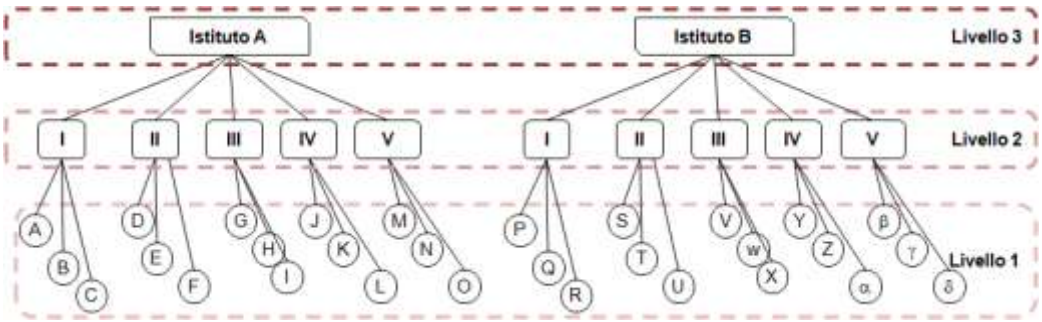

```
y <-c(.9, .8, .6, .6, .6, .5, .4, .4, .4, .3, .2, .2, .2, .2, .3, .4, .5, .7, .3, .4, .5, .5, .7, .7)
x <-c(.17, .08, .0, .25, .42, .0, .08, .17, .25, .33, .0, .25, .17, .33, .42, .33, .50, .58, .50, .58, .67, .75, .83, .92)
```

```
cluster<-c(rep("A", 10), rep("B",6),rep("C",6
  ))
```

```
cluster<-c(rep("A", 10), rep("B",8),
  rep("C",6))
```

```
cluster<-c(rep("A", 12), rep("B",6),
  rep("C",6))
```

Usiamo un semplice esempio con dati veri. Nel dataframe `idiomi` sono raccolte le performance di bambini in età scolare alla prova MT di Comprensione di un testo e le loro risposte a un compito di comprensione e di produzione di espressioni idiomatiche ("rompere il ghiaccio", "far ridere i polli"...), competenze che dovrebbero risentire di un'evidente **curva evolutiva** dalla prima alla quinta classe. Ci concentriamo sulla prova MT di comprensione di un brano: sono stati coinvolti alunni (livello 1) delle cinque classi (variabile di livello 2) di due Istituti (variabile di livello 3).



Il modello che corrisponde all'ipotesi di un'evoluzione nella comprensione del testo è: $\boldsymbol{y_i = b_0 + b_1 classe_i + e_i}$ → il punteggio del bambino i è dato dalla grand mean (intercetta) più l'effetto dell'appartenenza alla classe di i più l'errore del modello per i. Tuttavia, le classi sono gerarchicamente nidificate nella **variabile contestuale** di **livello 3** Istituto (ne indichiamo il livello con $\boldsymbol{j}$): poiché ogni Istituto può seguire un approccio educativo leggermente diverso, vi lavorano insegnanti di formazione differente, eccetera eccetera, è ragionevole attendersi che anche l'Istituto frequentato agisca sulla performance, rendendo i bambini che vi accedono sottilmente più simili tra loro dei bambini che frequentano l'altro plesso, a parità di classe. Quindi, potremmo impostare un **modello che rappresenti questa variazione tra gli istituti dell'effetto della classe** sulla comprensione del brano: o consentiamo a b_0 di variare tra gli istituti, o lo consentiamo a b_1, o a entrambi.

Per **includere nel modello** una $\boldsymbol{b_0}$ **random**, aggiungiamo $\boldsymbol{u_{0i}}$ **che quantifica la variabilità delle** $\boldsymbol{b_0}$**:** $b_{oi} = b_0 + u_{0i}$ → $y_i = (b_0 + u_{0i}) + b_1 classe_i + e_i$.

Se vogliamo **includere** nel modello **un termine random per** b_1**,** ovvero l'effetto della classe sulla comprensione, **aggiungiamo** $\boldsymbol{u_{1i}}$, che quantifica la variabilità dei coefficienti **angolari** tra gli **istituti**: $b_{1j} = b_1 + u_{1j}$ → $y_i = b_0 + (b_1 + u_{1j}) classe_i + e_i$.

Possiamo naturalmente **combinare** $\boldsymbol{b_0}$ **e** $\boldsymbol{b_1}$ **random** in un modello che prevede sia la variabilità delle intercette nei livelli di Istituto (indipendentemente dalla classe frequentata, le performance di comprensione di un testo dei bambini nei due Istituti sono diverse) sia la variabilità dei coefficienti angolari (gli effetti dell'appartenenza a una classe sono variabili nei due istituti): inseriamo un termine $\boldsymbol{u_{0i}}$ e un $\boldsymbol{u_{1i}}$: $y_i = (b_0 + u_{0j}) + (b_1 + u_{1j}) classe_i + e_i$.

Altrettanto naturalmente, è possibile inserire **almeno un altro** predittore al modello (per esempio, potremmo ritenere che il genere incida sulla performance al compito), e anche per questo predittore potremo decidere di usare solo il coefficiente fisso o di aggiungere anche il termine random: $y_i = (b_0 + u_{0j}) + (b_1 + u_{1j}) classe_i + b_2 genere + e_i$, oppure $y_i = (b_0 + u_{0j}) + b_1 classe_i * b_2 genere + e_i$, oppure $y_i = (b_0 + u_{0j}) + (b_1 + u_{1j})\, classe_i + (b_2 + u_{2j}) genere + e_i$, et cetera.

Stiamo quindi facendo una regressione in cui "consentiamo" all'uno o all'altro o a entrambi i parametri di variare tra i cluster, e, per ogni parametro random, **otteniamo anche una stima della variabilità del parametro**, oltre al valore del

parametro stesso. Per ogni predittore aggiunto al modello, possiamo decidere se i suoi parametri di regressione siano fissi o random – ma **useremo un criterio statistico per valutare quale scelta sia la migliore**.

Vediamo allora come costruire i parametri dei mixed linear models (ma lasceremo i calcoli a R, noi vedremo solo il retrobottega teorico) e come quantificare il fit dei modelli.

Attenzione: in questo capitolo ci occupiamo di modelli **lineari**, ma i mixed models possono essere applicati, con opportune modifiche ma con la stessa logica di fondo, anche a **modelli lineari generalizzati**, in cui la relazione tra Y e X non è lineare (come nella regressione logistica del capitolo precedente).

15.3 Fit, struttura e requisiti

I **parametri** di un mixed model sono stimati con il **metodo della massima verosimiglianza - ML**, e il suo **fit** è stimato come in una regressione logistica (capitolo 8), ovvero con l'indice di fit relativo $-2log-likelihood$ o **devianza** del modello, inversamente proporzionale al fit. Come nella regressione logistica, il χ^2 **Likelihood Ratio Test** determina se la differenza nelle devianze di due modelli è significativa. Oltre alla $-2LL$, tornano anche gli indici di fit relativo (capitoli 5 e 8): $AIC = -2LL + 2k$ e $BIC = -2LL + \ln(N) \times k$. Ricordiamo che k sono i **gradi di libertà** del modello, cioè i **parametri liberi** (quelli che devono essere stimati dai dati): in un modello nullo con intercetta fissa sono b_{01} e il termine di errore (la sua varianza, per la precisione), cioè $k = 2$; nel modello nullo con componente random sono b_0, u_{0i} e il termine di errore, cioè $k: df = 3$, eccetera.

Si possono anche usare $AICc$ (§5.1.3): $AICc = -2LL + 2\frac{k*\times(k-1)}{N-k-1}$, o $CAIC$ (**Consistent AIC**, di Bozdogan, 1987): $CAIC = -2LL + k \times (log(N) + 1),$ che nella correzione considerano sia il numero di predittori sia la numerosità campionaria.

Usando un approccio gerarchico, si può procedere impostando un modello "basic", in cui i coefficienti sono fissi, e poi aggiungere i termini random (e le eventuali covariate) modello per modello: in questo modo si possono confrontare i fit dei modelli e valutare l'apporto dei successivi sui precedenti utilizzando il LRT, ovvero verificando la significatività della differenza tra le $-2LL$ dei modelli, come abbiamo visto nella regressione logistica.

Quando si inseriscono termini random nel modello e/o ci sono misure ripetute, deve essere definita la struttura di covarianza dei dati (nel caso di termini random **e** misure ripetute, possono essere specificate strutture di covarianza diverse). La struttura di covarianza specifica la **forma della matrice di varianza – covarianza**, in cui gli elementi della diagonale sono varianze (la **covarianza di un vettore con se stesso corrisponde alla sua varianza, ovviamente**) e quelli fuori dalla diagonale sono covarianze. Tra le varie forme che una matrice di covarianza può assumere, dobbiamo comunicare a R quale noi pensiamo che sia quella corrispondente ai dati; si possono anche costruire più modelli, con diverse strutture di covarianza, e verificare il fit di ciascuno. Questa matrice è usata come punto di partenza per stimare i parametri dei modelli; perciò, si ottengono **diversi risultati a seconda del tipo di matrice scelta**. Se è **troppo semplice, sarà più probabile un errore di I tipo** (definire significativo un parametro che non lo è), e, viceversa, una matrice troppo complessa aumenterà la probabilità di errori di II tipo. Vediamo alcune delle matrici di varianza / covarianza più comuni:

1. **componenti della varianza** o **modello d'indipendenza**: è una struttura molto semplice, in cui tutti **i termini random sono indipendenti** ($covarianza = 0$) e le loro **varianze sono uguali** ($\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \sigma_4^2$; omoschedasticità): la loro somma corrisponde alla varianza totale di Y.

	x_1	x_2	x_3	x_4
x_1	1	0	0	0
x_2	0	1	0	0
x_3	0	0	1	0
x_4	0	0	0	1

2. **matrice diagonale**: come nella precedente matrice, tutti i termini random sono indipendenti ($covarianza = 0$), ma le loro **varianze sono diverse** ($\sigma_1^2 \neq \sigma_2^2 \neq \sigma_3^2 \neq \sigma_4^2$: eteroschedasticità.

	x_1	x_2	x_3	x_4
x_1	σ_1^2	0	0	0
x_2	0	σ_2^2	0	0
x_3	0	0	σ_3^2	0
x_4	0	0	0	σ_4^2

3. struttura **autoregressiva di I ordine (AR_1)**: le relazioni cambiano sistematicamente, per cui la relazione tra misure ripetute è più forte in punti adiacenti e diminuisce nel tempo. Nella struttura rappresentata le varianze sono uguali, ma esiste anche una versione eteroschedastica. È una struttura spesso **usata per dati a misure ripetute prese a distanza di tempo**.

	x_1	x_2	x_3	x_4
x_1	1	ρ	ρ^2	ρ^3
x_2	ρ	1	ρ	ρ^2
x_3	ρ^2	ρ	1	ρ
x_4	ρ^3	ρ^2	ρ	1

4. **non strutturata:** la struttura di covarianza è completamente indeterminata: le covarianze cambiano in modo impredicibile e le varianze non sono omogenee.

	x_1	x_2	x_3	x_4
x_1	σ_1^2	σ_{12}	σ_{13}	σ_{14}
x_2	σ_{21}	σ_2^2	σ_{23}	σ_{24}
x_3	σ_{31}	σ_{32}	σ_3^2	σ_{34}
x_4	σ_{41}	σ_{42}	σ_{43}	σ_4^2

Regole banali per scegliere una matrice di covarianza: se nel modello abbiamo u_{0i}, la matrice di componenti della varianza (di default nella funzione di R che useremo), andrà ragionevolmente bene; quando inseriamo (anche) u_{1i}, probabilmente sarà da assumere una matrice non strutturata; se i dati sono misure ripetute prese a distanza di tempo, probabilmente sarà opportuno assumere una matrice autoregressiva di I ordine.

A parte le "solite" assunzioni di una regressione lineare (ma possiamo dimenticarci della dipendenza delle osservazioni e dei residui, dato che il modello inserisce tra i propri termini la correlazione tra casi creata dalla variabile contestuale), è necessario fare attenzione ai coefficienti random: **sia u_{0i} sia u_{1i} devono essere normalmente distribuiti nel modello complessivo**, ovvero nei diversi livelli della variabile di contesto,
Attenzione anche alla **multicollinearità** per modelli con più b_1. In questo può essere d'aiuto la cautela di **centrare i predittori** (l'abbiamo visto nel Capitolo 5), cioè traslare i punteggi in scarti attorno alla $grand\ mean$ (la media complessiva della distribuzione X, come sappiamo), o alla media di gruppo: di solito si centrano le X di livello 1 sulla $grand\ mean$, ma è possibile anche centrare una variabile di livello 1 attorno alle **medie di una variabile di livello 2**. Nei modelli multilevel, se si centra attorno alla $grand\ mean$, il modello ottenuto dai punteggi grezzi e quello ottenuto dai punteggi centrati sono perfettamente equivalenti; il modello centrato sulla media di gruppo, invece, non è equivalente a quello grezzo, né per le componenti fisse né per quelle casuali, tranne quando solo b_0 è random e le medie di gruppo sono inserite nel modello come variabili di livello 2.

15.4 Qualche esempio di analisi multilevel

Vediamo tre esempi di analisi per mixed linear models: due per gruppi indipendenti, uno per misure ripetute. Useremo i package **nlme** e **MuMIn** per le analisi.

15.4.1 Disegni between groups

Cominciamo dal modello multivel più semplice per impratichirci: una X **a due livelli**, gruppi **indipendenti**, una variabile **contestuale** a due **livelli**.

Rispolveriamo il dataframe `disforia` usato nel capitolo 2: sappiamo che la **disforia** è una dimensione più facilmente riscontrabile tra le donne, per cui è una buona idea verificare che il test costruito per misurarla (`$daq`) sia davvero sensibile alla **differenza tra generi**. I soggetti sono stati raccolti in **diversi setting** che avrebbero dovuto corrispondere a diverse popolazioni per la dimensione in oggetto (`$gruppo_categorie`: CSM, SerT, ambulatorio ginecologico per i gruppi clinici; per la popolazione non clinica, soprattutto aule universitarie): usiamo la variabile `$gruppo_categorie` come variabile di possibile clusterizzazione, dato che potrebbe rendere i soggetti che appartengono a ciascun cluster più simili tra loro rispetto a soggetti appartenenti a cluster differenti. Eliminiamo però dall'analisi i soggetti appartenenti al livello `premestruale` (sono solo donne) e `sert` (sono solo uomini), e concentriamoci su pazienti CSM e popolazione non clinica. Vediamo il modello lineare con **soli parametri fissi**:

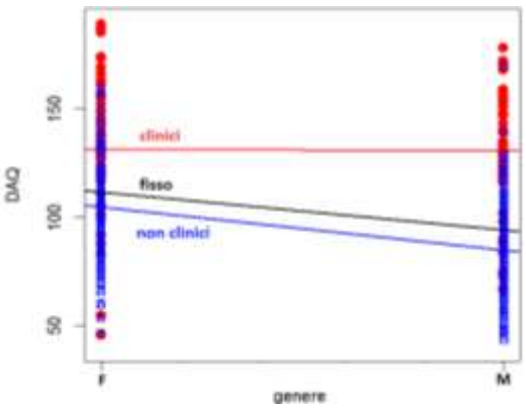
```
disfo<-subset(disforia, disforia$gruppo_categorie == "CSM"| disforia$gruppo_categorie == "non clinici")
summary(genere<-lm(disfo$daq~disfo$genere))
[…]
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        111.694      1.844  60.565  < 2e-16
disfo$generemaschio -17.496      2.521  -6.939 1.32e-11

Residual standard error: 27.29 on 469 degrees of freedom
Multiple R-squared:  0.09311,    Adjusted R-squared:  0.09118
F-statistic: 48.15 on 1 and 469 DF,  p-value: 1.317e-11
```

Nel test sulla disforia, gli uomini hanno in media 17.5 punti in meno: la differenza è significativa, e il genere spiega un po' più del 9% della varianza della disforia nel campione (altrettanto in popolazione).

Questo modello fisso, ricordiamo, **assume** che la **differenza uomini – donne (b_1) sia la medesima nei due cluster $gruppo, così come che la disforia per $X = 0$ (b_0) sia la medesima nei due cluster**.

Vediamo però come si comporta la relazione disforia~genere i due cluster:

```
csm<-subset(disfo, disfo$gruppo_categorie=="CSM")
noclinici<-subset(disfo, disfo$gruppo_categorie=="non clinici")

mod_csm<-lm(csm$daq~csm$genere)
mod_noclinici<-lm(noclinici$daq~noclinici$genere)

modcsm$coefficients
 (Intercept)   csm$generemaschio
       131.6                -0.9
modnocl$coefficients
(Intercept)   noclinici$generemaschio
  105.01829                -19.85493
```



Beh, **non si direbbe** che gli assunti del modello fisso tengano un granché: i soggetti dei cluster sono sensibilmente diversi nella loro variazione dalla *grand mean*, così come i b_1 nei due cluster sembrano descrivere un'assenza di relazione nel cluster CSM e una piuttosto

evidente relazione, nella direzione preannunciata dal modello fisso, nel cluster del campione non clinico.

Vediamo, allora, se **aggiungere le componenti u_{0i} e u_{1i} migliora il fit** del modello e la sua capacità di leggere la realtà.

Cominciamo dal modello nullo con la sola intercetta fissa e confrontiamolo il suo fit con quello un modello nullo in cui inseriamo **anche la componente u_{0j}** che cattura la componente random dell'intercetta. Se il modello con intercetta random ha un fit migliore (AIC o BIC più basso), significa che il valore di Y per $X = 0$ (b_0) è molto variabile tra i cluster, e che il modello che cattura questa variabilità si adatta ai dati meglio del modello con il solo coefficiente parametro b_0 fisso.

Per procedere all'analisi, dovremo usare il metodo della Massima Verosimiglianza: quindi, per confrontare il fit dei modelli è necessario che siano tutti costruiti con questo metodo, anche il modello nullo con intercetta fissa che finora abbiamo creato, utilizzando il metodo dei minimi quadrati, con `lm(Y~1)`. Sostituiamo `lm` con `glsErrore. Il segnalibro non è definito.(formula= Y~1, data=, method="ML")` del package `nlme`: `gls` (*generalized least squared*) costruisce il modello **lineare** usando il metodo della Massima verosimiglianza (`method="ML"`[116]), quindi il modello creato potrà essere confrontato con i successivi, che conterranno le componenti random dei parametri.

Il modello con b_0 e b_1 fissi costruito con questo metodo conferma la significatività dell'effetto del genere ottenuta nel modello costruito con il metodo OLS:

```
fissi_ML<-gls(model= daq~genere, data= disfo, method="ML")
summary(fissi_ML)
Generalized least squares fit by maximum likelihood
  Model: daq ~ genere
  Data: d
       AIC      BIC    logLik
  4455.437 4467.901 -2224.718

Coefficients:
                Value Std.Error  t-value p-value
(Intercept)   111.69406  1.844202 60.56497       0
generemaschio -17.49565  2.521267 -6.93923       0
```

Il modello nullo con **intercetta fissa** assume che il **punteggio medio di disforia per $X = 0$ non è significativamente variabile tra i cluster**.

```
nullo_fisso<-gls(daq~1, data= disfo, method = "ML")
summary(nullo_fisso)
Generalized least squares fit by maximum likelihood
  Model: daq ~ 1
  Data: disfo
      AIC     BIC    logLik
  4499.47 4507.78 -2247.735

Coefficients:
             Value Std.Error  t-value p-value
(Intercept) 102.3333  1.319109 77.57761       0


Standardized residuals:
        Min          Q1         Med          Q3         Max
-2.03979778 -0.67604726 -0.04662395  0.61776733  3.03055670

Residual standard error: 28.59761
Degrees of freedom: 471 total; 470 residual
```

Notate che i residui proposti da questo `summary` sono **standardizzati**.

[116] Il metodo della **massima verosimiglianza ristretta** (`method="REML"`, di default), l'altra possibile opzione per il calcolo dei parametri, non è adatto per confrontare modelli. In generale, ML produce stime più precise dei parametri fissi, cosa cui di solito si è più interessati, REML è migliore nella stima della variabilità dei parametri random, ma perlopiù le differenze non sono rilevanti (Twisk, 2006).

Ora costruiamo il modello che comprende la **variazione random** dell'intercetta: questo modello assume che il **punteggio medio di disforia per $X = 0$ è significativamente variabile tra i cluster**. Usiamo `lmeErrore. Il segnalibro non è definito.(fixed= Y~1, data=, random= ~parametro|variabile di contesto, method="ML")` di `nlme` (`lme` sta per *linear model estimate*), che permette di gestire l'aggiunta della componente random u_{0i} o u_{1i} nell'argomento `random=`, in cui sono indicati i parametri random (`~1` se intercetta, `~X`, se b_1) all'interno della variabile di contesto (`|variabile di contesto`). Indichiamo anche l'argomento `control=`, che gestisce una lista (`list`) di opzioni di default che possono essere cambiate, se il modello non converge, ovvero non raggiunge un risultato stabile nella stima dei parametri. Per esempio, si può **aumentare il numero di iterazioni** massimo della ML, che di default è $= 50$, con `control= list(maxlterm = numero di iterazioni)`. Spesso, più utile è la definizione dell'ottimizzatore, che di default è `nlminb` e può essere cambiato in **`optim:`** `control= list(opt="optim")`; se ci sono dati mancanti, va aggiunto `na.action=na.exclude`.

Avremo quindi:

```
nullo_random<-lme(fixed= daq~1, random = ~1|gruppo_categorie, data= disfo, method = "ML", contro
l=list(opt="optim"))
```

Spezziamo il commento dell'output del `summary(nodello)` nelle sue parti:

```
summary(nullo_random)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: disfo
      AIC      BIC    logLik
  4350.52 4362.985 -2172.26
```

Vediamo il **fit** e confrontiamolo con il precedente: tutti gli indicatori si sono concordemente abbassati: **l'aggiunta di u_{0i} migliora la capacità del modello** di descrivere i dati.

```
Random effects:
 Formula: ~1 | gruppo_categorie
        (Intercept) Residual
StdDev:    18.45905 24.11865
```

Questa è la sezione in cui in cui è riportata la **variabilità** (in forma di deviazione standard) di $\boldsymbol{b_0}$, unico parametro per cui abbiamo aggiunto una componente random, **e del termine di errore**.

```
Fixed effects: daq ~ 1
               Value Std.Error  DF  t-value p-value
(Intercept) 112.5093  13.13435 469 8.566035       0
```

Qui sono riportati il valore del termine fisso del modello (b_0), il suo SE, il test t e relativo $p - value$ per $H_0: b_0 = 0$.

```
Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.51897209 -0.58650740  0.03541798  0.65734336  3.14504488
```

Questi sono i residui (standardizzati) del modello: qualche problema ci sarebbe...

Il fit del modello è migliorato, ma è un miglioramento significativo? Per valutare se l'aggiunta della componente random u_{0i} ha migliorato significativamente il fit possiamo ricorrere ad `anova` in una sua ulteriore incarnazione, che fornisce AIC e BIC dei modelli, la loro LL e la significatività del LRT (differenza tra le devianze dei due modelli), cioè tutto quello che ci serve:

```
anova(nullo_fisso, nullo_random)
```

	Model	df		AIC	BIC	logLik	Test	L.Ratio	p-value
nullo_fisso	1	2	← b_0, e_i	4499.47	4507.780	-2247.735			
nullo_random	2	3	← b_0, u_{0i}, e_i	4350.52	4362.985	-2172.260	1 vs 2	150.9504	<.0001

I df dei modelli corrispondono al **numero di parametri liberi dei modelli**, cioè i parametri non fissati a priori, ma derivati dai dati: due nel primo (b_0, e_i) e tre nel secondo (b_0, u_{0i}, e_i).

Ricordate?

```
(AIC_random<-(-2*(-2172.260))+(2*3))
[1] 4350.52
(BIC_random<-(-2*(-2172.260))+(log(471)*3))
[1] 4362.985
```

La variazione del fit è significativa: aggiungere la variazione dell'intercetta all'interno dei gruppi migliora la capacità predittiva del modello. Evidentemente, come avevamo supposto dal grafico, il punteggio medio della disforia, indipendentemente dal genere, è variabile all'interno dei gruppi, e "catturare" questa variabilità nel modello nullo è un vantaggio.

Ripartiamo da questo modello nullo e verifichiamo se **l'aggiunta del predittore Genere migliora il fit** oppure no. Aggiungiamo prima solo la componente fissa del coefficiente angolare: restando costante il fatto che l'intercetta varia significativamente tra i gruppi, il modello afferma che l'effetto del genere sulla disforia, se significativo, è lo stesso in tutti i gruppi (beh, sappiamo già che in realtà non è proprio così. Nell'argomento `fixed=`, sostituiamo `~1` con `~X`[117]:

```
genere_fisso<-lme(data= disfo, fixed= daq~genere, random = ~1|gruppo_categorie, method="ML", na.
 action=na.exclude, control=list(opt="optim"))

summary(genere_fisso)
Linear mixed-effects model fit by maximum likelihood
 Data: disfo
       AIC      BIC    logLik
  4300.826 4317.446 -2146.413
```

Il fit è migliorato rispetto al modello nullo random: quindi, anticipiamo che il predittore dovrebbe essere significativo, come leggeremo nella sezione `fixed effects`:

```
Random effects:
 Formula: ~1 | gruppo_categorie
        (Intercept) Residual
StdDev:    17.87489 22.82858

Fixed effects: daq ~ genere
                  Value Std.Error  DF   t-value p-value
(Intercept)   120.54670 12.775830 468  9.435528       0
generemaschio -15.61993  2.117635 468 -7.376119       0
```

La disforia dei maschi è significativamente inferiore a quella delle donne; la differenza media è stimata in -15.6 punti.

Invece del contrasto semplice (o in aggiunta ad esso), possiamo ottenere la tradizionale tabella ANOVA, che riporta solo la significatività degli effetti fissi con F e $p - value$ con `anova.lme(modello con effetti random)` di `nlme`, ma anche la solita `anova(modello)` delle funzioni di base. È inutile con una X a due livelli, ma sarà preziosa per una X a più di due livelli o per più predittori. Vediamola, comunque;

```
anova.lme(genere_fisso)
            numDF denDF  F-value p-value
(Intercept)     1   468 78.12715  <.0001
genere          1   468 54.40713  <.0001 ← (-7.376119)^2: 54.40713
```

Possiamo proseguire con l'output:

```
Correlation:
              (Intr)
generemaschio -0.085
```

[117] Possiamo evitare di scrivere `fixed=daq~1+X`, anche se R accetta questa modalità, perché R di default considera l'intercetta presente nel modello; per escluderla, dovremmo scrivere `fixed= Y~0+X`.

Dovreste ricordare (capitolo 3) che nel modello "tradizionale" b_0 **e** b_1 **sono assunti come indipendenti**; nei modelli multilevel, questo assunto è messo alla prova: se la correlazione tra b_0 e b_1 (e tra i b_1, nei modelli che prevedono più b_1) dovesse essere effettivamente assente, sarebbe verificato; in caso contrario, avremmo costruito un modello migliore, mettendo a modello la loro reale covarianza (che, in questo modello, è praticamente assente).

```
Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-4.04463194 -0.64656716  0.01050399  0.65092620  3.62964876
```

Possiamo confermare:

```
anova(nullo_fisso, nullo_random, genere_fisso)
             Model df        AIC      BIC    logLik   Test    L.Ratio  p-value
nullo_fisso      1  2   4499.470 4507.780 -2247.735
nullo_random     2  3   4350.520 4362.985 -2172.260  1 vs 2  150.95041  <.0001
genere_fisso     3  4   4300.826 4317.446 -2146.413  2 vs 3   51.69347  <.0001
```

Abbiamo aggiunto un df, cioè un parametro libero al modello: b_0, u_{0i}, b_1, e_i

```
anova(nullo_fisso, nullo_random, genere_fisso)
             Model df      AIC      BIC    logLik   Test    L.Ratio  p-value
nullo_fisso      1  2 4499.470 4507.780 -2247.735
nullo_random     2  3 4350.520 4362.985 -2172.260  1 vs 2  150.95041  <.0001
genere_fisso     3  4 4300.826 4317.446 -2146.413  2 vs 3   51.69347  <.0001
```

Il genere ha un effetto significativo sulla disforia: la sua aggiunta al modello ne aumenta la capacità esplicativa secondo AIC, BIC e LRT.

Proseguiamo, allora, inserendo nel modello la componente random del b_1 u_{1i}: questo modello afferma che **l'effetto del Genere sulla disforia del brano, oltre che significativo, è significativamente variabile tra i cluster** (la differenza tra uomini e donne nel cluster CSM è diversa dalla differenza tra uomini e donne nel cluster popolazione non clinica. Aggiungiamo nell'argomento `random=` la variazione del b_1 specificando: `~genere`. Di **default**, `lme` assume che il modello deve comprendere anche u_{0i}, quindi non serve specificarlo[118]; vedremo nel §15.3 come escludere dal modello la componente random dell'intercetta, se non migliora significativamente il modello rispetto alla sola intercetta fissa.

```
genere_random<-lme(fixed=daq~genere, random = ~genere|gruppo_categorie, data= disfo,
 method="ML", na.action=na.exclude, control=list(opt="optim"))
summary(genere_random)
Linear mixed-effects model fit by maximum likelihood
 Data: disfo
       AIC      BIC    logLik
  4290.863 4315.793 -2139.432
```

Il fit è migliorato; questo modello descrive meglio la realtà di quello precedente.

```
Random effects:
 Formula: ~genere | gruppo_categorie
 Structure:General positive-definite,Log-Cholesky parametrization → matrice di varianza e covarianze
                 StdDev    Corr
(Intercept)   13.223786   (Intr)
generemaschio  9.457891   0.996
Residual      22.485210
```

È stata aggiunta la **stima della variabilità del b_1 nei cluster**; La prima colonna (`Stdev`) rappresenta la (radice quadrata della) varianza di ogni termine, mentre la colonna `Corr` esprime la covarianza dell'intercetta e del coefficiente angolare, decisamente rilevante.

[118] La formulazione completa dell'argomento sarebbe `random= ~1+ X|cluster`

```
Fixed effects: daq ~ genere
                 Value Std.Error  DF   t-value p-value
(Intercept)   118.2467  9.504630 468 12.440953  0.0000
generemaschio -10.4368  7.037488 468 -1.483029  0.1387
```

Oplà: la differenza tra uomini e donne mantiene la stessa direzione, ma **non è più significativa**: evidentemente, parte della variabilità di Y è stata "catturata" dalla variabilità di b_1 e di b_0, per cui la porzione assegnabile a X (SS_M) si è ridotta al punto da non esser più significativamente diversa dalla variabilità attribuita all'errore.

Ciò comporta che le **conclusioni tratte dal modello fisso, in realtà, non sono generalizzabili** a una popolazione indefinita: la differenza tra uomini e donne è rilevabile solo nella popolazione normativa, mentre in quella clinica l'effetto della psicopatologia, probabilmente, è più forte di quello del genere, nel livellare la disforia di uomini e donne.

```
Correlation:
              (Intr)
generemaschio 0.903

Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-3.798584319 -0.629855490 -0.003618033  0.656275188  3.769432617
```

Verifichiamo la variazione del fit:

```
anova(nullo_fisso, nullo_random, genere_fisso, genere_random)
               Model df      AIC      BIC    logLik   Test   L.Ratio p-value
nullo_fisso        1  2 4499.470 4507.780 -2247.735
nullo_random       2  3 4350.520 4362.985 -2172.260 1 vs 2 150.95041  <.0001
genere_fisso       3  4 4300.826 4317.446 -2146.413 2 vs 3  51.69347  <.0001
genere_random      4  6 4290.863 4315.793 -2139.432 3 vs 4  13.96311   9e-04
```

Sì, la variazione è significativa.

Se nel report (articolo, tesi…) si desidera riportare il rapporto F dell'effetto fisso, si può usare per l'ennesima volta `anova(modello)`, che riporta semplicemente:

```
anova(mod_b1_random)
            numDF denDF   F-value p-value
(Intercept)     1   468 1030.2577  <.0001
genere          1   468    2.1994  0.1387
```

Possiamo anche usare `model.sel(model class)` di `MuMIn`, che è anche più informativa e non richiede solo modelli nidificati:

```
model.sel(fissi_ML, nullo_fisso, nullo_random, genere_fisso, genere_random)
Model selection table
             (Intrc) gener             family class  na.action     control  random df
genere_random   118.2     + gaussian(identity)   lme na.exclude list(optim) gn|gr_c  6
genere_fisso    120.5     + gaussian(identity)   lme na.exclude list(optim)    gr_c  4
nullo_random    112.5       gaussian(identity)   lme                           gr_c  3
fissi_ML        111.7     + gaussian(identity)   gls                                 3
nullo_fisso     102.3       gaussian(identity)   gls                                 2
                logLik   AICc  delta weight
genere_random -2139.432 4291.0   0.00  0.993
genere_fisso  -2146.413 4300.9   9.87  0.007
nullo_random  -2172.260 4350.6  59.53  0.000
fissi_ML      -2224.718 4455.5 164.44  0.000
nullo_fisso   -2247.735 4499.5 208.45  0.000
Models ranked by AICc(x)
Random terms:
gn|gr_c = 'genere | gruppo_categorie'
gr_c = '1 | gruppo_categorie'
```

Attenzione ai df del modello `genere_random`: **per ogni b_1**, oltre ad aggiungere b_{1i} (stima della varianza dei b_1 tra i livelli del cluster), aggiungiamo anche un termine di stima della **covarianza tra b_0 e b_1** (che indica quanto b_0 e b_1 dipendono l'uno dall'altro), nonché un termine della **covarianza tra i b_1**.

In questo esempio abbiamo un solo b_1, quindi niente covarianza tra b_1, ma solo tra b_0 e b_1, per cui: $b_0 + u_{01} + e_i + b_1 + u_{1i} + cov_{b_0 b_1} \rightarrow parametri = 6$.

Vogliamo sapere altro? Non sempre, ma in alcuni casi può essere utile sapere se la variabilità tra le b_0 o tra i b_1 sia significativamente $\neq 0$, nonché quale sia il suo CI: si può usare **`intervals(modello random)`** del package **`nlme`**, che fornisce anche i CI degli effetti fissi:

```
intervals(genere_random)
Approximate 95% confidence intervals
 Fixed effects:
                  lower     est.      upper
(Intercept)    99.60932 118.2467 136.883998
generemaschio -24.23640 -10.4368   3.362792
```

Il CI del predittore Genere è ampio, e comprende 0, come sapevamo dal `summary` precedente.

```
 Random Effects:
  Level: gruppo_categorie
                       lower       est.      upper
sd((Intercept))    4.8026558 13.2237865 36.4107976
sd(generemaschio)  3.1277345  9.4578909 28.5995184
```

il range di variabilità dell'intercetta in popolazione è ampio, e non comprende 0, così come quello del coefficiente angolare: quindi, la **variabilità di b_0 e b_1 tra i gruppi è significativamente $\neq 0$,** confermando che abbiamo fatto bene a usare un modello con termini random.

Ora che abbiamo visto passo passo un metodo (gerarchico) per la costruzione dei modelli, possiamo ripassare anche la **modalità non nidificata** che abbiamo già usato nella regressione multipla per modelli lineari e generalizzati: sappiamo che `model.sel(model class)` di `MuMIn` restituisce una tabella in cui **i modelli sono ordinati dal migliore al peggiore** secondo $AICc$ e LL. Ricordiamo anche che, poiché questi indicatori non sono affatto esenti da difetti, i modelli dovrebbero operazionalizzare sensate relazioni tra costrutti, bisogna stare **attenti alle interpretazioni post hoc basate sul puro numero**.

```
model.sel(nullo_fisso, nullo_random, genere_fisso, genere_random)
Model selection table
             (Intrc) gener class na.action  control  random df      logLik     AICc  delta weight
genere_random   118.2     +   lme   na.excl lst(opt) gn|gr_c  6   -2139.432   4291.0   0.00  0.993
genere_fisso    120.5     +   lme   na.excl lst(opt)    gr_c  4   -2146.413   4300.9   9.87  0.007
nullo_random    112.5         lme                       gr_c  3   -2172.260   4350.6  59.53  0.000
nullo_fisso     102.3         gls                             2   -2247.735   4499.5 208.45  0.000
```

Il modello migliore è `genere_random`, e lo sapevamo.

Ricordiamo che nella prima parte della tabella troviamo le informazioni sui parametri (b_0 - `Intrc` e b_1, ciascuno indicato con il nome di X e "+" nei modello che lo prevedono). Rispetto alla tabella prodotta per i modelli lineari, troviamo anche indicazioni sul **tipo** di modello (nell'ordine: la sua classe, la gestione degli NA, il tipo di ottimizzazione, gli effetti random, i df). **Da `logLik` in poi, ritroviamo gli indicatori di fit,** i più interessanti: $logLik$, $AICc$, $delta$ (ricordiamo: la **differenza tra l'$AICc$ del modello migliore** e **ciascuno degli altri**). Dalla verosimiglianza relativa (*relative likelihood*), cioè l'esponziale di $-.5 \times delta$ $[exp(-.5 * \Delta)]$ del modello si ricava l'**Akaike's weight w_i** (tornate al capitolo 5, per ulteriori dettagli sul calcolo), interpretato come la **probabilità che il modello in oggetto sia il migliore, alla luce dei dati e dei modelli con cui è confrontato**. Nel nostro esempio, la probabilità che il modello con genere random sia migliore del modello con genere fisso è $.993/.007 = 141.9$ volte più grande – si direbbe che potremmo fidarci.

Sappiamo che grazie ai w_i si può determinare il ***confidence set*** della model class, inserendovi i modelli il cui w_i sia $\geq 10\%$ dei w_i più grande. Nell'esempio, questo includerebbe nel confidence set, oltre al migliore, tutti i modelli con una plausibilità maggiore del 10% di .993 $\rightarrow w_i > 0.99$... ma non ne esistono.

Come alternativa a `lme`, possiamo usare la funzione `lmer(Y~X+(componente random))` di `lme4`, la cui struttura è solo leggermente differente: i coefficienti random si aggiungono dopo la definizione degli effetti fissi; anche in questo caso si possono cambiare gli *optimizer* di default (`"nloptwrap"`). Di default, la funzione usa la correzione dei df dell'effetto fisso secondo Satterthwaite[119], che protegge dall'errore di I tipo in caso di violazione dell'omoschedasticità.

```
genere_fisso<-lmer(daq~genere+(genere|gruppo_categorie), data=d, REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

summary(genere_fisso)
Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's method [lmerModLmerTest]
Formula: daq ~ genere + (genere | gruppo_categorie)
Data: d
Control: lmerControl(optimizer = "bobyqa")

     AIC      BIC   logLik deviance df.resid
  4290.8   4315.7  -2139.4   4278.8      465
Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.7994 -0.6298 -0.0033  0.6561  3.7693

Random effects:
Groups           Name          Variance Std.Dev. Corr
gruppo_categorie (Intercept)   175.2    13.237
                 generemaschio  88.8     9.423   1.00
Residual                       505.6    22.485
Number of obs: 471, groups:  gruppo_categorie, 2

Fixed effects:
             Estimate Std. Error      df t value Pr(>|t|)
(Intercept)   118.253      9.493   1.977  12.457  0.00667
generemaschio -10.447      6.996   1.895  -1.493  0.28045
```

Vediamo un modello appena un po' più complesso, con **un predittore a più di due livelli**, verificando l'ipotesi usata per l'esempio a inizio capitolo: la **comprensione del brano** (Y: `idiomi$MT_comprensione_grezzo`) è **significativamente diversa per classe** frequentata (X `$idiomi$classe`), mettendo a modello la **variabilità di b_0 e b_0 tra gli Istituti** (cluster: `$idiomi$Istituti`)?

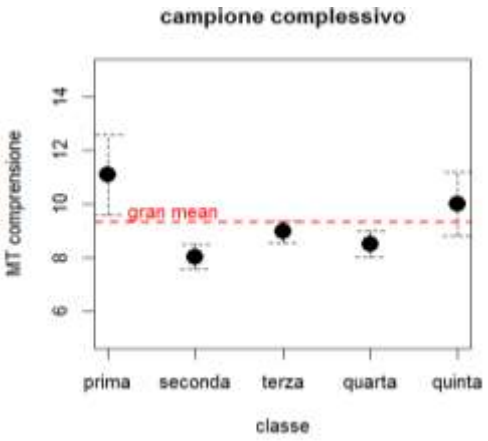
Qualche operazione preliminare: le etichette delle classi non seguono l'ordine naturale (quarta e quinta precedono seconda e terza, in ordine alfabetico). Ordiniamo il fattore `$classe`:

```
idiomi$classe<-ordered(idiomi$classe, levels=c("prima", "seconda","terza", "quarta", "quinta"))
```

Ora `$classe` è un fattore ordinato, quindi i contrasti di default sono polinomiali. Nulla di male, ma ci interessano i contrasti semplici:

[119] La ricordiamo: $d' = \frac{\left(\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}\right)}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1-1}+\frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2-1}}$

```
contrasts(idiomi$classe)<-contr.treatment(5,1)
contrasts(idiomi$classe)
        2 3 4 5
prima   0 0 0 0
seconda 1 0 0 0
terza   0 1 0 0
quarta  0 0 1 0
quinta  0 0 0 1

tapply(idiomi$MT_comprensione_grezzo, idiomi$classe, mean)
    prima    quarta    quinta   seconda     terza
11.093750  8.500000 10.000000  8.025000  8.958333
tapply(idiomi$MT_comprensione_grezzo, idiomi$classe, sd)
    prima    quarta    quinta   seconda     terza
4.1686957 1.1795356 3.1847853 1.4409310 0.9078961
```



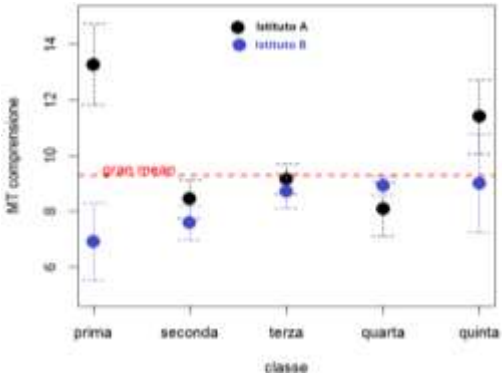
La presentazione dei dati nel campione complessivo, **ignorando l'informazione derivante dall'istituto**, sembra descrivere un vantaggio per la prima classe rispetto a tutte le altre – a parte, di poco, la quinta -, tra loro molto simili, con un'ampia variabilità soprattutto in prima, ma anche in quinta. Il modello "classico", con i soli parametri fissi e le stime dei parametri secondo il metodo dei minimi quadrati, ci direbbe:

```
summary(lm(idiomi$MT_comprensione_grezzo~idiomi$classe))
[...]
Residuals:
    Min      1Q  Median      3Q     Max
-8.0938 -1.0250  0.0417  1.5000  4.0000

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     11.0938     0.4552  24.373  < 2e-16
idiomi$classe2  -3.0688     0.6107  -5.025 1.47e-06
idiomi$classe3  -2.1354     0.6953  -3.071 0.002548
idiomi$classe4  -2.5938     0.6953  -3.731 0.000274
idiomi$classe5  -1.0938     0.6601  -1.657 0.099725

Residual standard error: 2.575 on 144 degrees of freedom
Multiple R-squared:  0.1735,    Adjusted R-squared:  0.1505
F-statistic: 7.555 on 4 and 144 DF,  p-value: 1.489e-05

summary(gls(MT_comprensione_grezzo~classe, data = idiomi, method = "ML"))
Generalized least squares fit by maximum likelihood
  Model: MT_comprensione_grezzo ~ classe
  Data: idiomi
       AIC      BIC    logLik
  711.5964 729.6201 -349.7982

Coefficients:
                  Value Std.Error   t-value p-value
(Intercept)   11.093750 0.4551623 24.373172  0.0000
classequarta  -2.593750 0.6952720 -3.730555  0.0003
classequinta  -1.093750 0.6601342 -1.656860  0.0997
classeseconda -3.068750 0.6106644 -5.025265  0.0000
classeterza   -2.135417 0.6952720 -3.071340  0.0025
```

L'effetto della classe è significativo, e spiega nel complesso il 17.4% di varianza nella comprensione nel campione (15% in popolazione: un contrasto non è significativo). Le significatività dei contrasti le avevamo già predette dal grafico: con la parziale eccezione del confronto I-V, in cui la variazione non è pienamente individuata, i punteggi si abbassano passando dalla prima alle classi superiori.

Rappresentiamo la situazione **nella variabile di clusterizzazione Istituto**: i soggetti sono ugualmente variabili attorno alla $grand\ mean$ (u_{0i})? E i coefficienti angolari dei contrasti sono variabili nei due livelli, o sono coerenti (u_{1i})?

```
tapply(idiomi$MT_comprensione_grezzo,list(idiomi$Istituto,
  idiomi$classe), mean)
      prima seconda    terza   quarta   quinta
A 13.285714    8.45 9.153846 8.083333 11.41667
B  6.909091    7.60 8.727273 8.916667  9.00000

tapply(idiomi$MT_comprensione_grezzo,list(idiomi$Istituto,
  idiomi$classe), sd)
     prima  seconda    terza    quarta   quinta
A 3.180296 1.468081 0.898717 1.5050420 2.108784
B 2.071451 1.313893 0.904534 0.5149287 3.482097
```



Beh, no. Seconde, terze e quarte sono simili (le seconde meno), ma le quinte e soprattutto le prime sono assolutamente diverse non solo tra loro, ma anche nel comportamento dei rispettivi contrasti: nell'Istituto A, passando dalla I alle altre classi i punteggi si abbassano (b_1 negativi), nell'istituto B i punteggi si alzano (b_1 positivi). Il modello che abbiamo costruito ignorando l'informazione sul cluster Istituto si perde parecchie informazioni importanti, e non si adatta affatto bene ai dati, in realtà. I presupposti per usare un approccio multilevel sembrano piuttosto fondati.

Possiamo anche quantificare i b_1 costruendo i subset `A` e `B` (o usando l'argomento `subset` di `lm`) e facendo i due modelli, anche se è ormai ridondante:

```
summary(lm(A$MT_comprensione_grezzo~A$classe))
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.2857     0.4617  28.777  < 2e-16
A$classe2    -4.8357     0.6610  -7.315 2.70e-10
A$classe3    -4.1319     0.7466  -5.534 4.65e-07
A$classe4    -5.2024     0.7656  -6.795 2.49e-09
A$classe5    -1.8690     0.7656  -2.441   0.0171

Residual standard error: 2.116 on 73 df
Multiple R-squared:0.5147,Adjusted R-squared: 0.4881
F-statistic: 19.36 on 4 and 73 DF,p-value: 6.841e-11
```

```
summary(lm(B$MT_comprensione_grezzo~B$classe))
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.9091     0.6219  11.109   <2e-16
B$classe2     0.6909     0.7743   0.892   0.3755
B$classe3     1.8182     0.8795   2.067   0.0426
B$classe4     2.0076     0.8610   2.332   0.0228
B$classe5     2.0909     0.7982   2.620   0.0109

Residual standard error: 2.063 on 66 f
Multiple R-squared:0.1397,AdjustedR-squared: 0.08753
F-statistic: 2.679 on 4 and 66 DF,  p-value: 0.03916
```

Dato che abbiamo giù visto un procedimento passo passo, costruiamo i possibili modelli e andiamo direttamente alla verifica della variazione del loro fit (se volete, potete creare i `summary` di ciascuno ed esercitarvi nella loro interpretazione):

```
modA<-lm(idiomi$MT_comprensione_grezzo~idiomi$classe2, subset = idiomi$Istituto=="A")
modB<-lm(idiomi$MT_comprensione_grezzo~idiomi$classe2, subset = idiomi$Istituto=="B")
round(modA$coefficients,3); round(modB$coefficients,3)
    (Intercept) idiomi$classe2 idiomi$classe3 idiomi$classe4 idiomi$classe5
         13.286         -4.836         -4.132         -5.202         -1.869
    (Intercept) idiomi$classe2 idiomi$classe3 idiomi$classe4 idiomi$class25
          6.909          0.691          1.818          2.008          2.091
nullo_fisso<-gls(MT_comprensione_grezzo~1, data= idiomi, method = "ML")
nullo_random <- lme(fixed= MT_comprensione_grezzo~1, random = ~1|Istituto, control = list(opt =
  "optim"), data = idiomi, method = "ML")
classe_fisso<-lme(fixed = MT_comprensione_grezzo~classe, random = ~1|Istituto, data= idiomi, met
  hod ="ML", control = list(opt="optim"))
classe_random<- lme(fixed= MT_comprensione_grezzo~classe, random = ~classe|Istituto, data= idiom
  i, method="ML", control=list(opt="optim"))
```

Ora verifichiamo se la variazione è andata nella direzione di un miglioramento:

```
anova(nullo_fisso, nullo_random, classe_fisso, classe_random)
              Model df       AIC       BIC      logLik     Test    L.Ratio   p-value
nullo_fisso       1  2  731.9830  737.9909 -363.9915
nullo_random      2  3  719.3582  728.3700 -356.6791   1 vs 2   14.62489     1e-04
classe_fisso      3  7  698.4617  719.4894 -342.2309   2 vs 3   28.89640    <.0001
classe_random     4 21  683.9437  747.0266 -320.9719   3 vs 4   42.51801     1e-04
```

Attenti ai df del modello `classe_random`: abbiamo più b_1, ciascuno corrispondente a un contrasto; quindi, oltre alla covarianza tra b_0e b_1 abbiamo anche le ***covarianze tra i*** b_1*:* $b_0 + u_{0i} + e_i + 4b_1 + 4u_{1i} + 4cov_{b_0 b_1} + 6\, cov_{b_1}$*, per un totale di parametri, cioè gradi di libertà, pari a* $df = 21$

anova ci pone il problema evidenziato nel capitolo 5: secondo AIC, LRT e LL, il modello **classe_random** migliora il fit, mentre **secondo BIC il fit peggiora drasticamente**, addirittura ottenendo un risultato peggiore del modello nullo con intercetta fissa. Non dovrebbe essere sorprendente, dato che **BIC penalizza il fit tanto più drasticamente quanti più parametri il modello contiene** – e il quarto modello triplica i suoi df, rispetto al terzo. Stante l'evidenza della descrizione dei dati nei due Istituti, prendo la responsabilità (e ovviamente potrei sbagliare...) di dar ragione ad AIC, attribuendo il peggioramento di BIC a un eccesso di penalizzazione dovuto ai molti df. Verifichiamo con l'informatore $AICc$ fornito da model.sel:

```
model.sel(nullo_fisso, nullo_random, classe_fisso, classe_random)
Model selection table
              (Intrc) class             family class.1     control
classe_random  10.120     + gaussian(identity)     lme list(optim)
classe_fisso   10.820     + gaussian(identity)     lme list(optim)
nullo_random    9.251       gaussian(identity)     lme list(optim)
nullo_fisso     9.295       gaussian(identity)     gls
                       random df   logLik  AICc delta weight
classe_random    c|I 21 -320.972 691.2  0.00  0.982
classe_fisso       I  7 -342.231 699.3  8.04  0.018
nullo_random       I  3 -356.679 719.5 28.30  0.000
nullo_fisso           2 -363.992 732.1 40.85  0.000
```

Infatti, $AICc$ conferma il deciso miglioramento introdotto da u_{1i}. Consideriamo quindi come **modello migliore** **classe_random** e commentiamone il summary:

```
summary(classe_random)
Linear mixed-effects model fit by maximum likelihood
 Data: idiomi
       AIC      BIC    logLik
  683.9437 747.0266 -320.9719
Random effects:
 Formula: ~classe | Istituto

 Structure: General positive-definite, Log-Cholesky parametrization
             StdDev   Corr
(Intercept)  3.149943 (Intr) classe2 classe3 classe4
Classe2      2.727946 -1.000
Classe3      2.937642 -1.000 1.000
Classe4      3.562840 -1.000 1.000  1.000
Classe5      1.960138 -1.000 1.000  0.999   0.999
Residual     2.033013
```

Le prime due colonne le abbiamo già viste: radice quadrata della varianza di ogni termine e correlazione (covarianza) tra intercetta e b_1; le successive sono le correlazioni (covarianze) tra i b_1.

```
Fixed effects: MT_comprensione_grezzo ~ classe
                  Value Std.Error  DF   t-value p-value
(Intercept)   10.121066  2.297595 143  4.405070  0.0000
classe2       -2.096066  2.025480 143 -1.034849  0.3025
classe3       -1.180257  2.188170 143 -0.539381  0.5905
classe4       -1.621066  2.625090 143 -0.617528  0.5379
classe5        0.081873  1.512066 143  0.054147  0.9569
```

La differenza nelle significatività dei contrasti, rispetto al modello con soli termini fissi, è eclatante. A parità di df, **nessuna differenza tra la I classe e le altre resta significativa**. Evidentemente, buona parte della variabilità di Y è stata "catturata" dalla variabilità dei $b1$ e di b_0 per cui la porzione assegnabile a X (SS_M) e ripartita in ciascun contrasto si è ridotta al punto da non esser più significativamente diversa dalla variabilità attribuita all'errore.

Possiamo applicare l'ubiqua funzione anova al modello multilevel per avere il rapporto F *overall*:

```
anova(classe_random)
            numDF denDF   F-value p-value
(Intercept)     1   143 1553.6450  <.0001
classe          4   143    3.3204  0.0124
```

L'effetto complessivo della classe è ancora significativo, ma il rapporto F è più che dimezzato rispetto al modello con soli effetti fissi – e quindi la significatività si è ridotta; notate che i df di errore sono diminuiti (da 144 a 143), avendo aumentato il numero di parametri nel modello.

Costruite i CI dei parametri del modello migliore.

E se, in presenza di effetti principali significativi, volessimo fare test post hoc? Potremmo usare la già nota funzione `glht(modello, linfct=mcp(predittore="contrasti"))` del package **multcomp**, che abbiamo visto nel capitolo 7 e si applica anche a multilevel model. Ricordiamo che in `linfct=mcp()` si specifica il tipo di contrasti o la matrice dei confronti, e che possiamo correggere il family-wise error rate in `summary(oggetto glht, test=adjusted(type="none/bonferroni/BH/holm…"))`.

```
posthoc_classe<-glht(classe_random, linfct=mcp(classe="Tukey"))


summary(posthoc_classe,test= adjusted(type = "BH"))
         Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: Tukey Contrasts
Fit: lme.formula(fixed = MT_comprensione_grezzo ~ classe, data = idiomi,
    random = ~classe | Istituto, method = "ML", control = list(opt = "optim"))
Linear Hypotheses:
                      Estimate Std. Error z value Pr(>|z|)
quarta - prima == 0   -1.62116    2.57921  -0.629   0.6487
quinta - prima == 0    0.08232    1.48413   0.055   0.9558
seconda - prima == 0  -2.09616    1.99270  -1.052   0.5857
terza - prima == 0    -1.18005    2.15435  -0.548   0.6487
quinta - quarta == 0   1.70348    1.26618   1.345   0.4463
seconda - quarta == 0 -0.47500    0.78882  -0.602   0.6487
terza - quarta == 0    0.44111    0.73319   0.602   0.6487
seconda - quinta == 0 -2.17848    0.73924  -2.947   0.0321
terza - quinta == 0   -1.26237    0.89565  -1.409   0.4463
terza - seconda == 0   0.91611    0.54786   1.672   0.4463
---
 (Adjusted p values reported -- BH method)
```

Solo un confronto è significativo.

```
classe_random<- lme(fixed= MT_comprensione_grezzo~classe, random = ~classe|Istituto, data=
  idiomi, method="ML", control=list(opt="optim"))
genere_fisso<-lme(fixed = MT_comprensione_grezzo~genere, random = ~1|Istituto, data= idiomi,
  method ="ML", control = list(opt="optim"))
genere_random<-lme(fixed = MT_comprensione_grezzo~genere, random = ~genere|Istituto, data=
  idiomi, method ="ML", control = list(opt="optim"))
classe_random_genere<-lme(fixed = MT_comprensione_grezzo~classe*genere, random =
  ~classe|Istituto, data= idiomi, method ="ML", control = list(opt="optim"))
classe_genere_random<-lme(fixed = MT_comprensione_grezzo~classe*genere, random =
  ~genere|Istituto, data= idiomi, method ="ML", control = list(opt="optim"))

model.sel(classe_random, genere_fisso, genere_random, classe_random_genere,classe_genere_random)
Model selection table
                     (Int) cls gnr cls:gnr             family random df   logLik
classe_random        10.120   +             gaussian(identity)    c|I 21 -320.972
classe_random_genere  9.386   +   +       + gaussian(identity)    c|I 26 -318.420
classe_genere_random 10.490   +   +       + gaussian(identity)    g|I 14 -339.337
genere_fisso          9.097       +         gaussian(identity)      I  4 -356.434
genere_random         9.097       +         gaussian(identity)    g|I  6 -355.577
                     AICc delta weight
classe_random        691.2  0.00   0.99
classe_random_genere 700.3  9.13   0.01
classe_genere_random 709.8 18.59   0.00
genere_fisso         721.1 29.93   0.00
genere_random        723.7 32.53   0.00
Models ranked by AICc(x)
Random terms:
```

```
c|I = 'classe | Istituto'
g|I = 'genere | Istituto'
I = '1 | Istituto'
```

```
anova(classe_random_genere)
               numDF denDF   F-value p-value
(Intercept)        1   138 1531.8804  <.0001
classe             4   138    3.2712  0.0135
genere             1   138    1.2928  0.2575
classe:genere      4   138    0.9130  0.4583
```

```
anova(classe_genere_random)
               numDF denDF  F-value p-value
(Intercept)        1   138 405.6286  <.0001
classe             4   138   8.2356  <.0001
genere             1   138   0.0942  0.7593
classe:genere      4   138   0.3180  0.8655
```

15.4.2 Disegni within subjects

Come detto, nei disegni a misure ripetute la **variabile contestuale usuale è il singolo soggetto**, entro il quale sono clusterizzate le multiple misure prese nella ricerca.

La gestione in R prevede la trasformazione del dataframe da *wide* a *long*, come sempre per le misure ripetute. Inoltre, dovremo specificare una matrice di covarianza adeguata, che potrebbe essere (ma non è una ricetta valida per tutti le ricerche, ovviamente) una **matrice autoregressiva di primo ordine** (§9.1), se le misure ripetute sono prese a distanza di apprezzabile tempo. Potremo usare ancora `lme` per costruire il singolo modello, e usare un approccio gerarchico costruendo dal modello nullo in su, oppure `model.sel` per concentrarci sul modello migliore. Infine, il fattore **within subject deve essere inserito nel modello come `numeric`**, per cui, dopo aver trasformato il dataframe in long format con **`melt`**, dovremo **ricordarci di trasformarlo da `factor` a `numeric`** prima di procedere con i modelli.

Usiamo dati noti (dataframe `sicurezza`) per procedere con più leggerezza: rivalutiamo il cambiamento nel tempo (T_0-T_2) della gravità dei **comportamenti** (Y) a rischio nei lavoratori che hanno frequentato il corso di formazione (X), questa volta **mettendo a modello anche la variazione dell'intercetta tra i soggetti** (cluster: `$codice`), perché riteniamo che, indipendentemente dal corso, i lavoratori si comportino in maniera differente nei confronti del rischio, nonché la **variazione del coefficiente angolare tra i soggetti**, perché riteniamo che il corso abbia avuto un impatto molto variabile da soggetto a soggetto.

Selezioniamo **solo i partecipanti al corso** e creiamo il dataframe **`cono`**, in formato *long*, con le sole variabili che ci servono: Y, X within e il cluster **`$codice`**, cioè l'identificativo dei soggetti:

```
s<-subset(sicurezza, sicurezza$gruppo!="controllo")
```

```
table(s$gruppo)
    controllo formazione non obbligatoria formazione obbligatoria
            0                          35                        56
```

```
s$gruppo<-droplevels(s$gruppo)
```

```
cono<-melt(data = s, id.vars = c("codice"), measure.vars = c("comportamenti_t0",
  "comportamenti_t1", "comportamenti_t2"))
```

```
names(cono)<-c("sogg","tempo","comportamenti")
```

```
class(cono$tempo)
[1] "factor"
```

```
levels(cono$tempo)<-c("T0", "T1", "T2")
```

Trasformiamo il fattore within in `numeric`: $T_0 \rightarrow 0$, $T_1 \rightarrow 1$, $T_2 \rightarrow 2$;

```
cono$tempo_num <- ifelse(cono$tempo == "T0", 0, ifelse(cono$tempo == "T1", 1 ,2))
```

```
class(cono$tempo_num)
[1] "numeric"
```

Avreste già dovuto verificare (esercizio nel capitolo 6) che nel modello con **soli termini fissi** la variazione è significativa tra T_0 e T_1, e resta sostanzialmente stabile a T_2.

Procediamo a costruire i modelli:

```
nullo_fisso<-gls(comportamenti~1, data=cono, method = "ML")
nullo_random<-lme(comportamenti~1, data=cono, random= ~1|sogg, method = "ML",
  control=list(opt="optim"))
AIC(nullo_fisso); AIC(nullo_random)
[1] 726.9219
[1] 727.2711

anova(nullo_fisso, nullo_random, tempo_fisso, tempo_random)
             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
nullo_fisso      1  2 726.9219 734.1409 -361.4610
nullo_random     2  3 727.2711 738.0995 -360.6355 1 vs 2  1.65083  0.1988
```

Non c'è un vantaggio nell'inserire u_{0i}: il ft resta sostanzialmente identico, e semmai peggiora. La variabilità tra i soggetti attorno alla grand mean, quindi, non è apprezzabile.

Torniamo al modello nullo senza u_{0i} e aggiorniamolo con il coefficiente b_1 fisso: dobbiamo usare ancora `gls`, per applicare la ML:

```
tempo_fisso<-gls(comportamenti~temponumerico, data=cono, method = "ML")
```

il modello fisso con il metodo dei minimi quadrati ci aveva detto che il predittore tempo è significativo, quindi certamente il fit del modello con predittore è migliore del modello nullo, ma, volendone testimonianza documentale, facciamo:

```
anova(nullo_fisso, tempo_fisso, tempo_random)
             Model df      AIC      BIC    logLik   Test  L.Ratio
nullo_fisso      1  2 726.9219 734.1409 -361.4610
tempo_fisso      2  3 665.3457 676.1741 -329.6728 1 vs 2 63.57625
```

Quod erat demonstrandum.

Ora, inseriamo u_{1i}: dobbiamo anche indicare la matrice di varianza – covarianza che riteniamo meglio si adatti ai dati: `correlation= corAR1(0, ~X)` ipotizza una struttura autoregressiva di primo ordine. Dobbiamo anche **escludere dal modello la componente u_{0i}**, che sarebbe inserita di default: quindi, esplicitiamo nell'argomento `random = ~0` *[escludi* u_{0i}*]* `+ tempo_num | sogg` *[*u_{1i} *entro soggetti]*

```
tempo_random <- lme(comportamenti ~ tempo_num, correlation= corAR1(0, ~tempo_num), data=cono, ra
  ndom= ~0 + tempo_num|sogg, method = "ML", control=list(opt="optim"))
```

Vediamo il modello migliore:

```
anova(nullo_fisso, tempo_fisso, tempo_random)
           Model df                                AIC      BIC    logLik    Test   L.Ratio p-value
nullo_fisso    1  2 ← b0, ei                   726.9219 734.1409 -361.4610
tempo_fisso    2  3 ← b0, b1, ei            665.3457 676.1741 -329.6728   1 vs 2  63.57625 <.0001
tempo_random   3  5 ← b0, b1, cov_b0b1, ei   656.3255 674.3729 -323.1628   2 vs 3  13.02013 0.0015

model.sel(nullo_fisso, tempo_fisso, tempo_random)
Model selection table
             (Int) tmp_num             family class          correlation
tempo_random 2.295 -0.5077 gaussian(identity)   lme corAR1(0,~tempo_num)
tempo_fisso  2.273 -0.5077 gaussian(identity)   gls
nullo_fisso  1.765         gaussian(identity)   gls

                 control  random df   logLik  AICc delta weight
tempo_random list(optim) 0+t_n|s  5 -323.163 656.6  0.00  0.988
tempo_fisso                       3 -329.673 665.4  8.88  0.012
nullo_fisso                       2 -361.461 727.0 70.42  0.000
Models ranked by AICc(x)
Random terms:
```

```
0+t_n|s = '0 + tempo_num | sogg'
```

La variazione random dell'effetto del tempo migliora il fit: evidentemente, il passare del tempo ha un effetto sulla gravità dei comportamenti a rischio apprezzabilmente variabile tra i soggetti.

```
summary(tempo_random)
Linear mixed-effects model fit by maximum likelihood
 Data: cono
       AIC      BIC    logLik
  656.3255 674.3729 -323.1628
Random effects:
 Formula: ~0 + tempo_num | sogg
            tempo_num  Residual
StdDev:   0.003473437 0.8093043

Correlation Structure: AR(1)
 Formula:  ~tempo_num | sogg
 Parameter estimate(s):
      Phi
0.2615757

Fixed effects: comportamenti ~ tempo_num
                 Value  Std.Error  DF   t-value p-value
(Intercept)    2.2954104 0.08196087 181 28.006174       0
tempo_num     -0.5076923 0.05811538 181 -8.735938       0
```

Il **tempo mantiene il suo effetto significativo** (passando da T_0 a T_2, la gravità dei comportamenti a rischio si abbassa, di circa mezzo punto, in media, per rilevazione).

```
Correlation:
              (Intr)
temponumerico -0.709

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.8362762 -0.7262824 -0.2224126  0.3952664  3.8361216
```

Perché nel modello OLS che avrete costruito come esercizio nel capitolo 13, con queste stesse variabili, compaiono due b_1 e qui uno solo?

Concludiamo l'analisi del modello con i CI dei parametri, fissi e random:

```
intervals(tempo_random)
Approximate 95% confidence intervals
 Fixed effects:
                  lower     est.      upper
(Intercept)    2.1342822 2.2954104 2.4565386
temponumerico -0.6219423 -0.5076923   -0.3934424
```

Il CI del predittore Tempo è piuttosto ristretto e non contiene 0, come sapevamo dal `summary` precedente.

```
 Random Effects:
  Level: sogg
                           lower        est.     upper
sd((temponumerico)) 6.271522e-13 0.003473437  19237384
```

Con la procedura `model.sel`, avremmo visto:

```
model.sel(nullo_fisso, nullo_random, tempo_fisso, tempo_random)
Model selection table
             (Intrc)   tmpnm class  control    correlation random df   logLik  AICc delta weight
tempo_random   2.295 -0.5077   lme lst(opt) crAR1(0,tmpnm)  0+t|s  5 -323.163 656.6  0.00  0.988
tempo_fisso    2.273 -0.5077   gls                                3 -329.673 665.4  8.88  0.012
nullo_fisso    1.765           gls                                2 -361.461 727.0 70.42  0.000
nullo_random   1.765           lme lst(opt)                     s  3 -360.636 727.4 70.81  0.000

Abbreviations:
control: lst(opt) = 'list(optim)'
```

```
correlation: crAR1(0,tmpnm) = 'corAR1(0,~temponumerico)'
Models ranked by AICc(x)
Random terms:
0+t|s = '0 + temponumerico | sogg'
s = '1 | sogg'
```

Appendice I

Calcolo combinatorio

Per stabilire la probabilità di un evento è utile, talvolta, ricorrere al *calcolo combinatorio*: dato un insieme di oggetti, il calcolo combinatorio fornisce i criteri per configurare i raggruppamenti che si possono formare con tali oggetti.

Diciamo che *a*, *b*, *c*, e *d* sono quattro oggetti (persone, automobili, ecc.), e immaginiamo di dover formare **gruppi di due elementi**: prima di tutto bisogna stabilire un criterio in base al quale si possa dire se due gruppi sono uguali oppure diversi, in base all'ordine con cui si succedono gli elementi nel gruppo.

	Gruppi possibili											
Se l'ordine conta, il gruppo ab è diverso dal gruppo ba	*ab*	***ba***	*ac*	***ca***	*ad*	***da***	*bc*	***cb***	*bd*	*db*	***cd***	*dc*
Se l'ordine conta, il gruppo ab è uguale al gruppo ba	*ab*	*ac*	*ad*	*bc*	*bd*	*cd*						

Possiamo quindi creare **disposizioni**, **permutazioni** o **combinazioni** degli oggetti:

a. Disposizioni

Dati *N* oggetti, si chiamano **disposizioni semplici** a *r* a *r* (o di classe *r*) i sottogruppi di oggetti (con *r*<*N*) che si possono formare con gli *N* oggetti dati, seguendo il criterio di **considerare distinti** due gruppi se differiscono **per almeno un elemento o se differiscono per l'ordine con cui gli elementi** si presentano.

Le disposizioni di classe 2 che si possono creare con i quattro oggetti a,b,c,d sono quindi quelle formate nella prima riga dell'esempio:

ab	***ba***	*ac*	***ca***	*ad*	***da***	*bc*	***cb***	*bd*	*db*	***cd***	*dc*

Il numero di disposizioni semplici per N oggetti a *r* a *r* si calcola:

$$\boldsymbol{D_{N,r} = N(N-1)\ldots(N-r+1) = \frac{N!}{(N-r)!}}$$

N! è N fattoriale, ovvero il prodotto **dei primi numeri N interi positivi** (0!= 1). Nel nostro esempio:

$$D_{4,2} = \frac{4!}{(4-2)!} = \frac{1x2x3x4}{1x2} = \frac{24}{2} = 12$$

Se volete usare R, il fattoriale di un numero positivo si richiede con la funzione `factorial(numero)`. Avremo quindi che le disposizioni semplici sono:

```
> factorial(4)/factorial(4-2)
[1] 12
```

Se, nel singolo gruppo, lo stesso elemento può comparire più di una volta, siamo invece nel caso di ***disposizioni con ripetizione***. In questo caso, le disposizioni si calcolano:

$$\boldsymbol{D_{N,r}^{rip} = N^r}$$

Per esempio, le disposizioni di classe 3 dei 2 oggetti *a* e *b* sono$D_{2,3} = 2^3 = 8$:

aaa	*abb*	*bab*	*bba*	*aba*	*baa*	*aab*	*bbb*

Cioè:

```
> 2^3
[1] 8
```

b. Permutazioni

Le **permutazioni semplici** di N oggetti sono un caso particolare delle disposizioni: precisamente, sono le **disposizioni di N oggetti di classe N**, cioè il numero di **sequenze ordinate** che si possono comporre con gli N oggetti. Per la precisione, una permutazione è una funzione (biiettiva), cioè l'operazione che consente di passare da una sequenza ordinata a un'altra, e non la sequenza… ma possiamo passarci sopra. Ad esempio, le permutazioni semplici delle tre lettere *a*, *b* e *c* sono:

abc	*acb*	*bac*	*bca*	*cab*	*cba*

Le permutazioni si calcolano con:

$$\boldsymbol{P_N = N!}$$

Cioè, nel nostro caso: $\mathbf{P_3 = 1x2x3 = 6.}$

```
> factorial(3)
[1] 6
```

`Permn(x= N)` di `DescTools` richiede di indicare in x un vettore di elementi e restituisce le permutazioni disponibili, non il loro numero, che corrisponde alle righe della matrice risultante (potremmo chiederlo con `nrow(Permn(x= N)`, per essere ridondanti):

```
> Permn(x = (1:3))
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    1    3
[3,]    2    3    1
[4,]    1    3    2
[5,]    3    1    2
[6,]    3    2    1
> nrow((Permn(x = (1:3))))
[1] 6
```

Nel caso in cui gli *n* oggetti **non** siano tutti diversi tra loro, ad esempio le tre lettere *a*, *a*, *b*, siamo nel campo delle permutazioni con ripetizioni

aaa	*baa*	*aba*

$$\boldsymbol{P_N^{(N_1,N_2,..,N_k)} = \frac{N!}{N_1!\,xN_2!\,x\,\ldots xN_k!} =}$$

Naturalmente con: $N_1+N_2+\ldots+N_k$. Infatti, se n_1 oggetti sono uguali tra loro: $P_N = N_1!\,xP_N^{N_1} = \frac{P_N}{N_1!}$

Nell'esempio: $P = \frac{3!}{2!} = \frac{1x2x3}{1x2} = 3$

Le permutazioni non sono disponibili tra le statistiche di base in R. Naturalmente, se ricordate che le permutazioni semplici sono N!, è facile calcolarne il numero; per avere l'elenco delle permutazioni, o per le permutazioni con ripetizione, però, sono diversi i package disponibili: per esempio, potete scaricare il package **`gtools`** e usare la funzione **`permutations(n=N, r=r, , v=vettore; repeats.allowed= TRUE/FALSE)`**; l'argomento logico `repeats.allowed=` consente di specificare se le permutazioni sono con o senza ripetizione. Ad esempio:

```
>x<-c("a,"b,"c)
```

```
>permutations(n=3,r=2,v=x, repeats.allowed=TRUE)
>     [,1] [,2]
>[1,] "c" "c"
>[1,] "c" "b"
>[1,] "c" "a"
>[1,] "b" "c"
>[1,] "b" "b"
>[1,] "b" "a"
>[1,] "a" "c"
>[1,] "a" "b"
>[1,] "a" "a"
```

Per conoscere solo il numero:

```
>nrow(permutations(n=3,r=2,v=x, repeats.allowed=TRUE)
[1] 9
```

Potete provare anche il package **combinat**.

c. Combinazioni

Le **combinazioni semplici** di N oggetti di classe *r* sono tutti i sottogruppi di *r* distinti elementi che si possono formare estraendoli dall'insieme di partenza: si considerano **distinti due sottogruppi solo se differiscono per almeno un oggetto**, **non ha importanza l'ordine** con cui sono estratti i loro elementi e **non si può ripetere lo stesso elemento più volte**. Le combinazioni derivano dal rapporto tra Disposizioni e Permutazioni.

Ad esempio, con *a*, *b*, *c*, *d* le combinazioni semplici di classe 2 sono:

ab	*ac*	*ad*	*bc*	*bd*	*cd*

Il numero di combinazioni semplici per N oggetti a *r* a *r* si calcola:

$$C_{N,r} = \frac{N!}{r!(N-r)!}$$

Nel nostro esempio: $C_{4,2} = \frac{4!}{2!(4-2)!} = \frac{1x2x3x4}{1x2(1x2)} = \frac{24}{4} = 6$

Le combinazioni si indicano anche con $\binom{n}{k}$, notazione che prende il nome di coefficiente binomiale, in cui n=N e k= r.

In R si usa la funzione **combn(N, r)**, che fa parte delle statistiche di base, o la funzione **choose(N,k)**:

```
> combn(4,2)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    2    2    3
[2,]    2    3    4    3    4    4

> choose(n = 4,k = 2)
[1] 6
```

CombN(x= N, k= r) di DescTools richiede di indicare in x un vettore di elementi:

```
> CombN(x = c(1:4), m = 2)
[1] 6
```

Invece di N si può specificare anche un vettore per ottenere

```
> combn(c("a","b","c","d"),2)
[,1] [,2] [,3] [,4] [,5] [,6]
[1,] "a"  "a"  "a"  "b"  "b"  "c"
[2,] "b"  "c"  "d"  "c"  "d"  "d"
```

Le ***combinazioni con ripetizione*** considerano i raggruppamenti che si possono formare con N oggetti, rimuovendo la condizione che il singolo oggetto si presenti più di una volta in ciascun gruppo. Ad esempio, date le due lettere *a* e *b*, le combinazioni con ripetizione di classe 4 che si possono formare sono:

aaaa	*aaab*	*aabb*	*abbb*	*bbbb*

Il numero di combinazioni con ripetizione per N oggetti a *r* a *r* si calcola:

$$\boldsymbol{C_{N,r}^{rip} = \frac{(r+N-1)!}{N!\,(r-1)!}}$$

E quindi: $C_{4,2}^{rip} = \frac{(2+4-1)!}{(1x2x3x4)(2-1)!} = \frac{5!}{24} = \frac{120}{24} = 5$

Appendice II
Un esempio di power analysis con R

Nel §6.6.4 è stata brevemente delineata l'analisi di potenza: in questa appendice vediamo come farla usando il package **pwr**, creato secondo le indicazioni di Cohen. Nel package sono contenute **più funzioni** dedicate alla power analysis, ciascuna dedicata un diverso test statistico; per tutte, gli argomenti sono quelli indicati nel §6.6.4: **n=** numerosità campionaria, **d=** effect size nella forma del coefficiente di Cohen, **sig.level**= alfa, **power=** 1-beta. Indicati tre argomenti, la funzione calcola il quarto.

	Test	Funzione
Proporzioni	un campione	pwr.p.test
	2 campioni, N uguale	pwr.2p.test
	2 campioni, N diverso	pwr.2pn.test
Analisi della varianza		pwr.anova.test
Chi quadrato		pwr.chisq.test
Modello lineare generale		pwr.f2.test
Correlazione		pwr.r.test
T-test	Un campione; due campioni; appaiati	pwr.t.test
	Due campioni, N diverso	pwr.t2n.test

Ad esempio, se si effettua un confronto tra le medie di due campioni di uguale numerosità, per un effect size previsto forte, con α= .05 e β= .80, la numerosità minima di ogni campione per rilevare una differenza non casuale tra le medie dei campioni, ammesso che tale differenza esista tra le due popolazioni, è pari a 26:

```
>pwr.t.test(d = .80, sig.level = .05, power = .80, type = "two.sample")
     Two-sample t test power calculation
n = 25.52457
              d = 0.8
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
NOTE: n is number in *each* group
```

Se invece **l'effect size stimato è debole**, a parità di α e β, la numerosità minima di ciascuno dei due campioni **sale** drammaticamente:

```
>pwr.t.test(d = .30, sig.level = .05, power = .80, type = "two.sample")
     Two-sample t test power calculation
n = 175.3847
              d = 0.3
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
NOTE: n is number in *each* group
```

Per stimare se un campione è rappresentativo della popolazione (one sample), per un effect size previsto forte, con α= .05 e β= .80, la numerosità minima del campione è pari a 14:

```
>pwr.t.test(d = .80, sig.level = .05, power = .80, type = "one.sample")
     One-sample t test power calculation
              n = 14.30278
              d = 0.8
      sig.level = 0.05
          power = 0.8
alternative = two.sided
```

Ma se l'effect size è debole, N sale:

```
>pwr.t.test(d = .30, sig.level = .05, power = .80, type = "one.sample")
     One-sample t test power calculation
              n = 89.14936
              d = 0.3
      sig.level = 0.05
          power = 0.8
alternative = two.sided
```

Attenzione: tra le statistiche di base sono disponibili le funzioni `power.t.test`, `power.anova.test` e `power.prop.test`, che consentono di fare la power analysis rispettivamente per i t-test, l'analisi della varianza e la differenza tra proporzioni. Gli elementi sono analoghi a quelli delle funzioni `pwr`, che, come abbiamo visto, hanno comunque un range di applicazione più ampio.

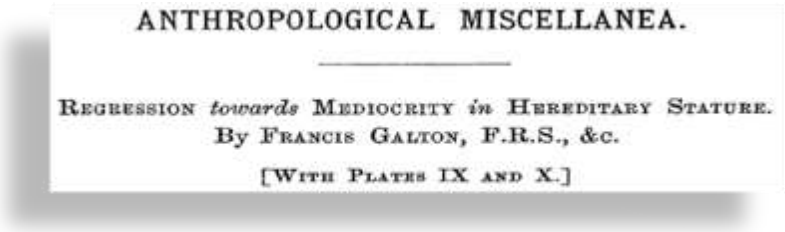
Per vedere un esempio, replichiamo il calcolo della numerosità campionaria per un t-test a campione unico, con effect size forte, usando la funzione di base: in questo caso, **delta corrisponde alla differenza tra le medie**, non a un coefficiente di effect size.

```
> power.t.test(delta = .80,sig.level = .05,power = .80,type = "one.sample")
     One-sample t test power calculation
n = 14.3028
           delta = 0.8
              sd = 1
      sig.level = 0.05
power = 0.8
alternative = two.sided
```

Per analisi più complesse, è disponibile il package **`pwr2`**.

Appendice III
L'origine della regressione lineare

L'applicazione dell'analisi di regressione all'ambito delle scienze sociali nasce in questo articolo:



ANTHROPOLOGICAL MISCELLANEA.

REGRESSION *towards* MEDIOCRITY *in* HEREDITARY STATURE.
By FRANCIS GALTON, F.R.S., &c.
[WITH PLATES IX AND X.]

Notate che si parla di regressione "toward mediocrity", ovvero verso la mediocrità – nel senso letterale del termine: vediamo perché. I dati di Galton, tratti dalla tabella I dell'articolo, sono inseriti nel dataframe `galton1886`, su Elly.

'Il modello lineare che calcola la retta con il metodo least squares appare ufficialmente nel 1806 (Legendre la utilizza per prevedere le orbite delle comete), anche se Gauss rivendica (1809) di aver applicato il metodo già nel 1795 per prevedere orbite di asteroidi. Tuttavia, l'applicazione del metodo alla statistica per le scienze sociali si deve, quasi un secolo dopo, a Galton e successivamente a Pearson (allievo di Galton) e Fisher. Galton, cugino di Darwin, nato in una famiglia di banchieri, è stato uno studioso decisamente eclettico: esploratore, meteorologo, scopritore e sostenitore delle impronte digitali come mezzo di identificazione, statistico (applica la distribuzione normale all'intelligenza ed introduce l'uso dei **percentili** nelle distribuzioni normali), psicologo e sostenitore entusiasta della quantificazione dell'intelligenza, e soprattutto **genetista** ante – Mendel. Influenzato da Darwin e dal darwinismo sociale di Spencer, fonda **l'eugenetica** o ***behavioral genetics*** (cui peraltro aderiscono sia Pearson sia Fisher): classi sociali e "razze" inferiori potrebbero "migliorarsi" solo incrociandosi con elementi delle classi superiori, non con l'educazione; apre il dibattito ambiente – natura (anche i "caratteri morali" sarebbero ereditari ed innati: "teoria del sangue blu") e dà il via agli studi sui gemelli per l'analisi dei caratteri ereditari. In effetti, tuttavia, Galton è stato evidentemente piuttosto selettivo nell'usare il lavoro del cugino per supportare il proprio di vista del meccanismo dell'ereditarietà. Secondo Cowan (1977), Galton non avrebbe mai realmente compreso il tema dell'evoluzione per selezione naturale, né era interessato al problema della creazione di nuove specie.

A ogni modo, l'interesse fondamentale di Galton era la trasmissione intergenerazionale dei caratteri: nell'uomo esistono fattori ereditari fisici e psicologici che possono essere espressi in una legge matematica? Per esempio:**conoscendo la statura dei genitori, è possibile prevedere la statura dei figli?** Questa è la domanda cui Galton risponde nell'articolo citato, dopo aver sperimentato per anni sulla trasmissione ereditaria dei caratteri nei piselli dolci: "The experiments showed further that **mean filial regression towards mediocrity** was directly proportional to the parental deviation from it. This **curios result** was based on so many plantings [...] during one, two or even three generations of the plants, that I could entertain no doubt of the truth of my conclusions."

Per passare dai vegetali agli umani, reclutò 930 figli/e adulti e 205 **coppie parentali** (quindi il dato dei genitori è l'altezza **media della coppia**); consapevole delle differenze di genere, "pondera" l'altezza delle femmine con fattore di correzione di 1.08. Nel dataframe **`galton1886`** sono disponibili solo le coppie per cui sono indicati entrambi i valori: quindi N é leggermente inferiore all'originale (N=892 verso 930), ma i risultati sono del tutto analoghi.

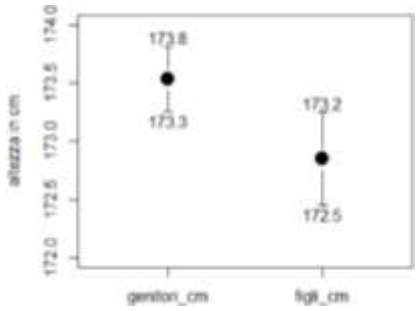
TABLE I.

NUMBER OF ADULT CHILDREN OF VARIOUS STATURES BORN OF 205 MID-PARENTS OF VARIOUS STA
(All Female heights have been multiplied by 1·08).

Heights of the Mid-parents in inches.	Heights of the Adult Children.													Total Number of	
	Below	62·2	63·2	64·2	65·2	66·2	67·2	68·2	69·2	70·2	71·2	72·2	73·2	Above	Adult Children.	Mid-parents.
Above ..	..	..	..	..	..	..	..	..	..	..	..	1	3	..	4	5
72·5	..	..	..	..	..	..	..	1	2	1	2	7	2	4	19	6
71·5	..	..	..	..	1	3	4	3	5	10	4	9	2	2	43	11
70·5	1	..	1	..	1	1	3	12	18	14	7	4	3	3	68	22
69·5	..	..	1	16	4	17	27	20	33	25	20	11	4	5	183	41
68·5	1	..	7	11	16	25	31	34	48	21	18	4	3	..	219	40
67·5	..	3	5	14	15	36	38	28	38	19	11	4	..	..	211	33
66·5	..	3	3	5	2	17	17	14	13	4	..	..	..	..	78	20
65·5	1	..	9	5	7	11	11	7	7	5	2	1	..	..	66	12
64·5	1	1	4	4	1	5	5	..	2	..	..	..	..	..	23	5
Below ..	1	..	2	4	1	2	2	1	1	..	..	..	..	..	14	1

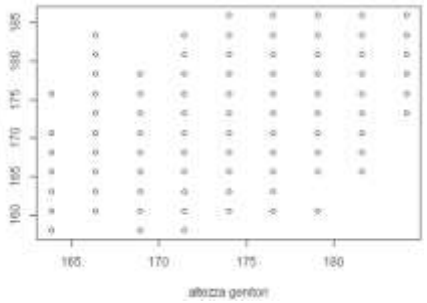
Curiosiamo con i mezzi del XXI secolo in questi gruppetti familiari del XIX, rinominando il dataframe come g. I dati di Galton sono in **pollici** (*inches*: 1 *inch*= 2.54 cm); per nostro agio, trasformiamoli in cm:

```
> g$genitori_cm<-g$genitori*2.54
> g$figli_cm<-g$figli*2.54
> summary(g$genitori_cm); summary(g$figli_cm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  163.8   171.4   174.0   173.5   176.5   184.2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  158.0   168.1   173.2   172.8   175.8   185.9

> cohen.d(g$genitori, g$figli, paired = TRUE)
Cohen's d
d estimate: 0.1285116 (negligible)
95 percent confidence interval:
     lower      upper
0.05661913 0.20040409
```



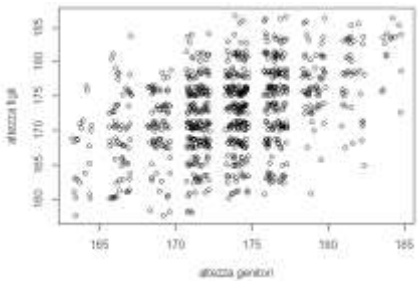
Le altezze medie delle due generazioni sono molto prossime, ma, ahimè per Galton, i figli tendono a essere mediamente più piccoli dei genitori: l'effetto è trascurabile, ma certamente non in linea con le previsioni dell'eugenetica. La coppia genitoriale più piccola è un po' più alta del figlio più piccolo (163.8 *vs* 158.5); la coppia genitoriale più alta è un po' più piccola del figlio più alto (184.2 *vs* 185.9). Il range interquartilico dei genitori è più ridotto di quello dei figli, ma ricordiamoci che l'altezza dei genitori è un dato medio tra madre e padre, e per sua natura la media attenua la variabilità.

Dopo aver visto le due distribuzioni univariate, vediamo quello che davvero ci interessa, ovvero la **distribuzione bivariata genitori – figli** (con relativo baricentro in rosso): in ascissa X-genitori, in ordinata Y-figli



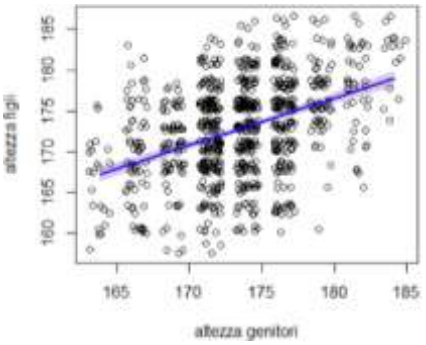
In realtà, moltissime osservazioni sono sovrapposte (Galton ha approssimato gli *inches* a un solo decimale: .2 per le altezze dei figli, .5 per quelle genitoriali). Aggiungiamo una sorta di "secondo decimale" random usando la funzione `jitter(variabile, noise=)`, che attribuisce una piccola quantità casuale (***noise***) alle osservazioni.

```
>plot(jitter(g$genitori_cm, factor = 1.5), jitter(g$figli_cm,fac
tor = 1.5), xlab="altezza genitori", ylab="altezza figli")
```



La natura della relazione adesso sembra un po' più chiara: si direbbe lineare e positiva.

Allora, **applichiamo il miglior modello lineare (retta) per stimare la relazione di predizione dell'altezza genitoriale sull'altezza dei figli** (Galton l'ha fatto a mano). Dato che i caratteri si trasmettono "all'ingiù" tra le generazioni, X sarà naturalmente l'altezza dei genitori e Y sarà l'altezza dei figli.




```
>print(galton<-lm(figli_cm~genitori_cm, data=g))

Call:
lm(formula = figli_cm ~ genitori_cm)

Coefficients:
(Intercept)  genitori_cm
    72.6844       0.5772
```

Ed ecco il risultato ha turbato Galton e gli eugenetici: per ogni cm in più di altezza tra i genitori, la variazione in altezza dei figli **è di solo 0.6 cm in più**: **L'incremento in Y (figli) è minore dell'incremento in X (genitori)**.Da qui la sconfortata definizione di ***regression toward mediocrity***, in tutto il senso negativo del termine (oggi si dice ***toward mean***, che è più politically – and **statistically**– correct): è vero che i figli di genitori più alti tendono a essere più alti dei figli di genitori più bassi, **ma non quanto** ci si potrebbe aspettare in base all'altezza del genitori.
Non solo: se calcoliamo il coefficiente di determinazione R^2:

```
>cor(g$genitori_cm, g$figli_cm)^2
[1] 0.1645501
```

Scopriamo che l'altezza dei genitori spiega solo il 16.5% della variabilità dell'altezza dei figli; il restante 83.5% deve evidentemente essere determinato da fattori che non sono la trasmissione ereditaria del carattere "altezza" [ammesso che una tale cosa esista – e ora sappiamo che non esiste], altro dato sconfortante da una prospettiva eugenetica.
Affrontiamo ora quello che è stato il colpo di grazia per Galton: **vediamo qual è l'altezza media dei figli all'interno dei singoli valori (classi) dell'altezza genitoriale**. Usiamo `tapply` chiedendo la media della distribuzione `figli_cm` (in nero) per ogni valore della distribuzione `genitori_cm` (in rosso); arrotondiamo a 1 decimale per semplificare:

```
>round(tapply(g$figli_cm, g$genitori_cm, mean),1)
 163.83  166.37  168.91  171.45  173.99  176.53  179.07  181.61  184.15
  166.5   169.6   170.3   171.7   172.8   174.2   176.6   177.6   181.4
```

I figli dei genitori più bassi (da 163.8 a 171.5) sono mediamente e sensibilmente più alti di loro; invece, **i figli dei genitori più alti** (da 176.5 a 1874.1) sono **mediamente e sensibilmente più piccoli di loro,** ovvero **regrediscono verso la media** della propria distribuzione (che ricordiamo essere = 172.8).
Galton, un po' mestamente, spiega il risultato ereditariamente (anche perché il dato umano conferma lo stesso meccanismo, cioè il "curious result", che aveva scoperto nei semi): "The explanation of item as follows. The child inherits partly from his parents, partly from his ancestry. Speaking generally, **the further his genealogy goes back, the more numerous and varied will his ancestry become**, until they cease to differ from any equally numerous samples taken at haphazard from the race at large. Their mean stature will then be the same as that of the race; in other words, it will

be mediocre". Insomma, sarebbe colpa di nonni, bisnonni, trisnonni e giù giù tra gli antenati, che non sarebbero stati abbastanza attenti a evitare *mesalliances* in tempo utile per preservare i discendenti.
Però, Galton avrebbe trovato sollievo a spiegare il risultato statisticamente. Proviamo a invertire figli e genitori nei ruoli di X e Y: Galton non l'avrebbe mai fatto, perché è un non senso genetico postulare l'altezza dei figli come predittore di quella genitoriale, ma noi sì:

```
>print(inversione<-lm(g$genitori_cm~g$figli_cm))
Call:
lm(formula = genitori_cm ~ figli_cm)
Coefficients:
(Intercept)     figli_cm
   124.2564       0.2851
```

Il coefficiente angolare dice che per ogni incremento unitario in X (altezza dei figli: 1 cm), l'altezza dei genitori Y **aumenta di meno di un terzo di centimetro: l'incremento in Y (genitori) è minore dell'incremento in X (figli)**. Oh: allora aveva ragione Galton? Vediamo l'altezza media dei genitori per ogni valore (classe) dell'altezza dei figli:

```
>round(tapply(genitori_cm, figli_cm, mean),1)
157.988 160.528 163.068 165.608 168.148 170.688 173.228 175.768 178.308 180.848 183.388 185.928
  169.3   169.7   172.2   172.1   171.9   172.5   173.6   174.0   175.1   175.5   178.2   178.3
```

Effettivamente, se i genitori dei figli più bassi sono più alti di loro, i genitori dei figli più alti sono più bassi dei propri figli. Ma non ha ragione l'eugenetica: il motivo è tutto in un "trucco" nel modo in cui è calcolato b_1, cioè nel metodo dei minimi quadrati.

Abbiamo detto che il coefficiente angolare è: $b_1 = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sum(x_i-\bar{x})^2}$, che equivale a: $\boldsymbol{b_1 = r\frac{s_Y}{s_X}}$

Infatti:

```
>galton$coefficients
(Intercept) genitori_cm
 72.6843516   0.5772145
```

È uguale a:

```
>cor(g$genitori_cm, g$figli_cm)*(sd(g$figli_cm)/sd(g$genitori_cm))
[1] 0.5772145
```

Dato che quasi sempre $r \neq 1$, quando X cambia di una unità allora Y cambia di una unità **moltiplicata per r**, cioè **cambia di *r* deviazioni standard**: e se **r<1**, allora la **variazione unitaria di Y è inferiore all'unità**.
Inoltre, se s_y e s_x sono uguali, allora b_1 sarà identico al coefficiente di correlazione *r*, e quindi < |1|; a maggior ragione, se le due deviazioni standard non sono uguali, allora avremo b_1<|r| < 1: **solo se $s_y > s_x$ potremo avere b_1>|1|**, ma questo dipenderà sia da *r* sia dal rapporto s_y/s_x.
Le deviazioni standard dei dati di Galton sono tutt'altro che uguali: quella di **Y è maggiore di quella di X**, ma la correlazione è **debole:**

```
>sd(g$figli_cm); sd(g$genitori_cm);cor(g$genitori_cm, g$figli_cm)
[1] 6.022513
[1] 4.232427
[1] 0.4056477
```

Perché il coefficiente angolare della regressione di Galton fosse b_1> 1, e quindi perché la variazione in altezza dei figli fosse superiore alla variazione unitaria nei genitori, la correlazione tra X e Y avrebbe dovuto essere **superiore a .70**:

```
> rapporto<-sd(figli_cm)/sd(genitori_cm)
 >.4*rapporto      >.5*rapporto      >.6*rapporto      >.7*rapporto      >.75*rapporto
 [1] 0.5691781     [1] 0.7114727     [1] 0.8537672     [1] 0.9960617     [1] 1.067209
```

Ovvero, con buona pace di Galton, l'ambiente dovrebbe avere molta meno influenza di quanta evidentemente ne ha nella trasmissione del "carattere" altezza.

Per concludere con Forrest (1974): "non è vero che la discendenza sia stata spinta verso la mediocrità dalla pressione dei suoi remoti, mediocri antenati, ma come una conseguenza della correlazione meno che perfetta tra genitori e figli. Restringendo la sua analisi ai figli di un campione selezionato di genitori e tentando di capire la loro deviazione dalla media, Galton non riesce a dar conto della devianza di tutti i figli [...] La conclusione di Galton è che la regressione è perpetua, e che il solo modo in cui può avvenire un cambiamento evolutivo è la comparsa di "mutazioni", di esemplari che differiscono in maniera rilevante dal prototipo della loro specie".

Appendice V
Model selection e averaged parameter

Nel capitolo 11 abbiamo usato la funzione `model.sel` del package `MuMin` per definire il modello migliore in una *model class* e il *confidence set* di modelli alternativamente plausibili.
Usiamo qui il package **`AICcmodavg`** per il calcolo degli *averaged parameters* (e dei loro SE), con lo scopo di ottenere una stima del più prossima al vero coefficiente angolare in popolazione, che tenga conto anche della verosimiglianza relativa dei modelli in cui è calcolato.
Usiamo i modelli usati nel §5.2.3 per la relazione depressione ~ burden; si inizia costruendo la lista dei candidate models, cioè la ***model class***, che **deve essere un oggetto di classe `list`**:

```
> model_class<-list()
```

Riempiamo la lista, indicando come suoi argomenti i modelli proposti per la selezione; poiché nel capitolo 5 avevamo già creato i modelli per la *model selection*, per brevità usiamo il loro nome, invece della formula completa:

```
> model_class[[1]]<-nullo
> model_class[[2]]<-restrizione
> model_class[[3]]<-fisico
> model_class[[4]]<-restrizione_fisico
> model_class[[5]]<-restrizione_per_fisico
```

Se non specifichiamo diversamente i nomi degli oggetti della lista, negli output R li identificherà di default come `Mod1`, `Mod2`, ecc. Per chiarezza, potremo anche specificarli con `names`:

```
>names(model_class) <- c("nullo", "restrizione", "fisico", "restrizione_fisico",
"restrizione_per_fisico")
```

Sistemata la lista, possiamo usare la funzione **`aictab(cand.set= model class, sort = TRUE)`** per avere una tabella di fit basata sull'AICc, con modelli in ordine decrescente di fit, simile a quella di `model.sel`: sono presenti solo i quantificatori di fit, non i parametri, e in un ordine un po' diverso:

```
> aictab(cand.set=model_class, sort= TRUE)
Model selection based on AICC:
                        k      AICc   Delta_AICc   AICcWt   Cum.Wt        LL
 restrizione_fisico     4     278.8         0.00     0.57     0.57  -134.837
 restrizione_per_fisico 5     280.5         1.70     0.24     0.82  -134.378
 fisico                 3     281.1         2.29     0.18     1.00  -137.220
 nullo                  2     298.5        19.73    0.000    1.000  -147.111
 restrizione            3     300.4        21.62    0.000    1.000  -146.886
```

Come facilmente intuibile, `k` sono i df del modello (parametri); `AICc`, `Delta_AICc` e `LL` sono esattamente corrispondenti alla tabella vista nel §5.2.3, come `AICcWt` è il *weight* (w_i) del modello. Unica novità, `Cum.Wt` è il *weight* cumulato dei modelli.
Per calcolare gli *averaged parameters*, usiamo la funzione **`modavg(model class, parm= "averaged parameter da stimare")`**; nell'argomento `parm` deve essere indicato il nome del predittore esattamente come appare nei modelli. Inoltre, useremo l'argomento **`exclude=`** se il predittore di cui si chiede l'*averaged parameter* è coinvolto in una interazione (o in un polinomio di secondo ordine) in uno o più dei modelli candidati. L'*averaged parameter* del predittore **`fisico`** è uguale a:

```
> modavg(model_class, parm= "a$CBI_burden_fisico", exclude=
"a$CBI_burden_fisico*$CBI_burden_restrizione")

Multimodel inference on "a$CBI_burden_fisico" based on AICc

AICc table used to obtain model-averaged estimate:

                         k      AICc   Delta_AICc   AICcWt    Estimate        SE
 fisico                  3    281.11         2.29     0.18        1.45      0.29
 restrizione_fisico      4    278.82         0.00     0.57        1.82      0.33

Model-averaged estimate: 1.73
Unconditional SE: 0.36
95% Unconditional confidence interval: 1.03, 2.43
```

La stima di b_1 per X_{fisico} nei due modelli è = 1.45 e = 1.82. La semplice media aritmetica dei due, ignorando l'informazione sulla significatività, darebbe un b_1 stimato uguale a:

```
> (1.45+1.82)/2
[1] 1.635
```

Invece, **l'*averaged parameter* stimato per il predittore X_{fisico} è = 1.73**: siccome il modello `restrizione_fisico` è più plausibile del modello `fisico` (w_i= .57 *versus* w_i= .18), la stima ponderata per la plausibilità del modello fa "pendere" il b_1 verso un valore più prossimo a quello di `restrizione_fisico`.

Lo SE e il CI sono definiti **unconditional** (rispetto al modello) proprio in quanto trascendono lo specifico modello, mentre SE e CI calcolati modello per modello sono *conditional* rispetto ad esso. L'*unconditional* SE è calcolato dalla *Model Selection Variance* (**MSV**), calcolata per ogni parametro di ogni modello come differenza al quadrato tra l'*averaged parameter* e il parametro "grezzo": $MSV = (averaged\ parameter - parametro\ del\ modello)^2$. La somma dello SE^2 (varianza condizionale) e della MSV, sotto radice quadrata e moltiplicata per il w_i del modello, restituisce lo SE pesato, cioè lo SE *unconditional* dell'output.

L'*unconditional* CI per ogni *averaged parameter* è calcolato come: $averaged\ parameter \pm t_{N-1} \times SE_{conditional}$

Notate l'uso del quantile *t*, per df = N-1, invece del quantile *z*.

L'*averaged parameter* di $X_{Restrizione}$, invece, è uguale a:

```
> modavg(model_class, parm= "a$CBI_burden_restrizione_tempo",exclude=
"a$CBI_burden_fisico*a$CBI_burden_restrizione")

Multimodel inference on "a$CBI_burden_restrizione_tempo" based on AICc

AICc table used to obtain model-averaged estimate:

                         k      AICc   Delta_AICc   AICcWt    Estimate        SE
 restrizione             3    300.44        21.62        0        0.17      0.27
 restrizione_fisico      4    278.82         0.00        1       -0.51      0.23

Model-averaged estimate: -0.51
Unconditional SE: 0.23
95% Unconditional confidence interval: -0.97, -0.05
```

Il modello con la sola restrizione è il peggiore tra i modelli del set, ha una plausibilità irrilevante rispetto a quella del modello additivo `restrizione_fisico`: inevitabilmente, la stima del b_1 (e del suo SE) coincide con quella del coefficiente angolare nel modello `restrizione_fisico`.

Appendice VI

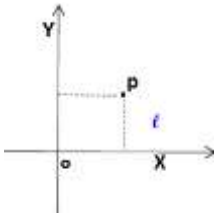
Ripasso elementare su elementi di trigonometria e logaritmi per non perdersi nelle correlazioni e nella regressione logistica

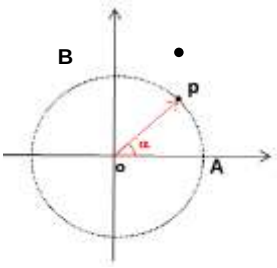
Nel capitolo 8 abbiamo usato alcuni elementi di trigonometria per descrivere i coefficienti di correlazione, nel capitolo 13 siamo incappati nei logaritmi. Nel caso siano passati diversi anni e molto oblio da quando li avete studiati, li ri-definiamo qui. Non saranno **mai** argomento di esame, naturalmente.

Un minimo di trigonometria…

Si parte da un sistema di assi X e Y nello spazio e da un problema: definire la posizione di un punto **P** rispetto agli assi.

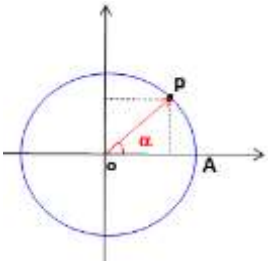
Possiamo risolverlo usando **due** parametri, ovvero le **coordinate di P in X e Y** P(X,Y):





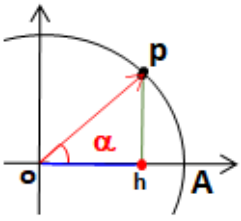
oppure usando **un solo** parametro, cioè un **angolo α**.

Si traccia una **circonferenza, centrata** nell'origine degli assi O di raggio $r_{\overline{OA}}$ **= 1**, e si considera **l'angolo tra il raggio della circonferenza e l'asse X** (o meglio il **semi-asse X**, dato che si considera solo il "pezzetto" di X del quadrante in cui appare P). Come abbiamo visto nel capitolo 8, l'ampiezza dell'angolo può essere espressa in gradi (sessagesimali) o radianti (l/r), ma il suo significato è lo stesso: **posizione di P rispetto a XY**.



Ora, sovrapponiamo le coordinate XY e l'angolo α: l'**ascissa** del punto P è il **coseno dell'angolo α**; l'**ordinata in Y del punto P è il seno dell'angolo α.**

Allarghiamo il quadrante per vedere meglio:



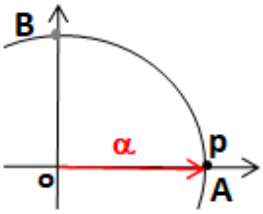
Quindi, $cos_\alpha = \overline{OH}$ e $sin_\alpha = \overline{PH}$, o, meglio, usando il valore assoluto: $|cos_\alpha| = \overline{OH}$ e $|sin_\alpha| = \overline{PH}$

Ricordandoci che il **raggio OA è = 1**, possiamo trarre "a occhio" alcune conclusioni su coseno e seno:

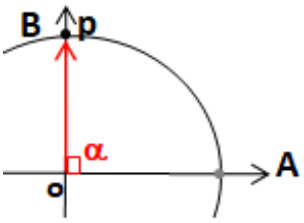
- Se il **punto P coincide con A**, l'angolo α sarà =0°, il suo coseno OH coincide con il raggio OA e quindi $cos_\alpha = 1$; il suo seno PH coincide con O, quindi è $sin_\alpha = 0$.
- Se il **punto P coincide con B**, l'angolo α sarà =90°, il suo coseno OH coincide con O e quindi $cos_\alpha = 0$ e il suo seno PH coincide con il raggio OB, quindi è $sin_\alpha = 1$.

Possiamo verificarlo con R, ricordando di convertire in radianti i gradi, moltiplicandoli per π/180:

```
> cos(0*pi/180)
[1] 1
> sin(0*pi/180)
[1] 0
```

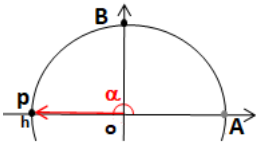



```
> round(cos(90*pi/180),1)
[1] 0
> sin(90*pi/180)
[1] 1
```

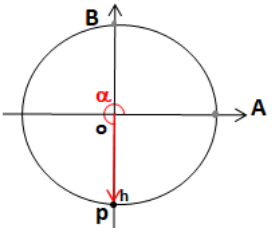


Nei quadranti **negativi** (X<0, Y<0), naturalmente, il **segno** di **seno** e **coseno**, quando sono **≠ 0**, è **negativo**:

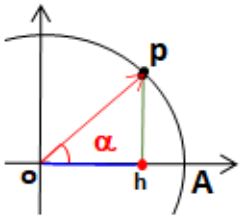
```
round(cos(180*pi/180),1)
[1] -1
round(sin(180*pi/180))
[1] 0
```




```
round(cos(270*pi/180),1)
[1] 0
round(sin(270*pi/180),1)
[1] -1
```



Per tutte le altre posizioni P, comunque, ci aiuta il teorema di Pitagora: $\overline{OP} = \sqrt{\overline{OH}^2 + \overline{PH}^2}$



Logaritmi

Il logaritmo in base *a* di un numero positivo *b* è **l'esponente che dobbiamo dare ad *a* (base) per ottenere *b* (argomento)**: $\log_a = b$. Per calcolarli con R, la funzione di base è `log(x=argomento, base= base)`, che di default è in base naturale *e*. La base naturale (numero di Nepero) è = 2.718282…

Per esempio: qual è l'esponente che dobbiamo dare a 2 (base) per ottenere 16 (argomento), cioè il logaritmo in base 2 di 16?

```
> log(x = 16,base = 2)
[1] 4
```

Qual è l'esponente che dobbiamo dare a 3 (base) per ottenere 27, cioè il logaritmo in base 3 di 27?

```
> log(x = 27,base = 3)
[1] 3
```

Qual è l'esponente che dobbiamo dare a 10 (base) per ottenere 100, cioè il logaritmo in base 10 di 100?

```
> log(x = 100,base = 10)
[1] 2
```

E così via.

Ricordiamo che sia l'argomento sia la base devono essere numeri positivi (dato che qualsiasi elemento elevato a potenza non può essere negativo).

```
> log(x = -16,base = 2)
[1] NaN
> log(x = 16,base = -2)
```

```
[1] NaN
```

… e che la base deve essere ≠1

```
> log(x = 16,base = 1)
[1] Inf
```

… dato che ciascun numero reale positivo (argomento) si può scrivere univocamente come potenza di un altro numero positivo (base), diverso da 1.

I logaritmi, essendo potenze di numeri, godono delle proprietà delle potenze; ne abbiamo usate nella regressione logistica:

1) Il prodotto di due potenze con la stessa base equivale alla somma delle potenze: $a^x * a^y = a^{x+y}$ → Il logaritmo di un prodotto (x*y) equivale alla somma dei logaritmi di x e di y: $\log_a(x*y) = \log_a x + \log_a y$

```
> (5^3)*(5^2); 5^(3+2)
[1] 3125
[1] 3125
> log(6*4); log(6)+log(4)
[1] 3.178054
[1] 3.178054
```

2) Il quoziente di due potenze con la stessa base equivale alla differenza delle potenze: $a^x / a^y = a^{x-y}$ → Il logaritmo di un rapporto x/y equivale alla differenza tra i logaritmi di x e di y: $\log_a(x/y) = \log_a(x) - \log_a(y)$.

```
> (5^3)/(5^2); 5^(3-2)
[1] 5
[1] 5
> log(36/6); log(36)-log(6)
[1] 1.791759
[1] 1.791759
```

3) La potenza di una potenza equivale al prodotto delle potenze della base: $a(x)^y = a^{x*y}$ → il log di x^y è uguale all'esponente y per il $\log_a$ di x, ovvero $\log_{a(x^y)} = y*\log_a(x)$

```
> (5^3)^2; 5^(3*2)
[1] 15625
[1] 15625
> log(36^2); 2*log(36)
[1] 7.167038
[1] 7.167038
```

Per completezza, ricordiamo le altre due proprietà delle potenze:

4) La radice ennesima di un numero a equivale al numero elevato all'inverso della radice ennesima: $\sqrt[n]{x} = x^{1/n}$.

```
> sqrt(25); 25^(1/2)
[1] 5
[1] 5
```

5) Il rapporto di due basi diverse elevate alla stessa potenza *n* equivale al rapporto delle basi elevato alla potenza *n*: $\frac{x^n}{y^n} = \left(\frac{x}{y}\right)^n$

```
> (5^2)/(4^2);(5/4)^2
[1] 1.5625
[1] 1.5625
```

La funzione che associa a ogni numero positivo il corrispondente logaritmo è la **funzione logaritmica**. Il suo inverso è la **funzione esponenziale**, che **associa a ogni logaritmo la corrispondente base**: $y = a^x$. La base della funzione esponenziale *a* deve essere >0 e ≠1 (se fosse = 1, avremmo $y = 1^x$, ma dato che 1 elevato a qualsiasi potenza è sempre = 1, avremmo costruito una funzione **costante**).

Script per i primi esercizi

Script centrocampo, Capitolo 2

```
centrocampisti<-c("hakimi","perisic", "vidal", "barella", "brozovic")
centrocampo_goal<-c(7,4,1,3,2)
centrocampo_presenze<-c(37,32,23,36,33)
centrocampo_minuti<-c(2672,1800,1142,2900,2584)
centrocampo_assist<-c(8,4,1,7,6)
centrocampo_maglia<-c(2,14,22,23,77)
centrocampo_nascita <- as.Date(c("1998/11/04", "1998/02/02", "1987/05/22", "1997/02/07",
"1992/11/16"))
centrocampo<-data.frame(campione=centrocampisti, goal= centrocampo_goal,
presenze=centrocampo_presenze,minuti=centrocampo_minuti,
assist=centrocampo_assist,maglia=centrocampo_maglia,nascita=centrocampo_nascita)
centrocampo$maglia<-as.character(centrocampo$maglia)
str(centrocampo)
```

Script empatia per i gatti, Capitolo 3

```
# Punto 1
table(gatti$riconosce_miao_cibo)
table(gatti$riconosce_miao_isolamento)
table(gatti$riconosce_miao_spazzolamento)
round(prop.table(table(gatti$riconosce_miao_cibo)),2)
round(prop.table(table(gatti$riconosce_miao_isolamento)),2)
round(prop.table(table(gatti$riconosce_miao_spazzolamento)),2)
round(prop.table(table(gatti$riconosce_miao_cibo)),3)*100
round(prop.table(table(gatti$riconosce_miao_isolamento)),3)*100
round(prop.table(table(gatti$riconosce_miao_spazzolamento)),3)*100
#Punto 2
gatti$istruzione2[gatti$istruzione=="diploma superiore"]<-"diploma superiore"
gatti$istruzione2[gatti$istruzione=="laurea"|gatti$istruzione=="specializzazione post
lauream"]<-"laurea o specializzazione"
#oppure
gatti$istruzione3<-ifelse(test= gatti$istruzione=="diploma superiore", yes = "diploma
superiore", no="laurea o specializzazione")
class(gatti$istruzione2)
table(gatti$istruzione2)
gatti$istruzione2<-factor(gatti$istruzione2)
class(gatti$istruzione2)
levels(gatti$istruzione2)
table(gatti$istruzione2)
# è un fattore ordinato; per fortuna, i livelli in ordine alfabetico creati da R sono coerenti
con l'ordine naturale della variabile, per cui non è necessario modificarli
# Punto 3
vive_con_gatto<-subset(gatti,gatti$vive_con_gatto=="si")
table(vive_con_gatto$riconosce_miao_cibo)
table(vive_con_gatto$riconosce_miao_isolamento)
table(vive_con_gatto$riconosce_miao_spazzolamento)
round(prop.table(table(vive_con_gatto$riconosce_miao_cibo)),2)
round(prop.table(table(vive_con_gatto$riconosce_miao_isolamento)),2)
round(prop.table(table(vive_con_gatto$riconosce_miao_spazzolamento)),2)
round(prop.table(table(vive_con_gatto$riconosce_miao_cibo)),3)*100
round(prop.table(table(vive_con_gatto$riconosce_miao_isolamento)),3)*100
round(prop.table(table(vive_con_gatto$riconosce_miao_spazzolamento)),3)*100
```

```
#Punto 4
#a
table(gatti$empatia_gatti)
#b
empatia_gatti_categorie<-c(rep(1,21), rep(2,38), rep(3,20))
empatici<-gatti[order(gatti$empatia_gatti),]
head(empatici$empatia_gatti)
tail(empatici$empatia_gatti)
empatici$empatia_gatti_categorie <-factor(empatia_gatti_categorie,level=c(1:3), labels =
c("bassa","media","travolgente"))
head(empatici$empatia_gatti_categorie)
tail(empatici$empatia_gatti_categorie)
View (empatici)
numeratore<-table(empatici$empatia_gatti_categorie)
denominatore<-c(8.5-2.5, 18.5-8.5, 27.5-18.5)
densita<-numeratore/denominatore
densita
```

Script relazione con i gatti, Capitolo 3#1

```
round(prop.table(gatti$autovalutazione_relazione_con_gatti),3)*100
#no, non è stata una buona idea: per variabili continue con distribuzioni così ampie, i modelli
descrittori che funzionano per variabili discrete sono cattivi modelli: non rappresentano
sinteticamente e affidabilmente il dato empirico
#2
summary(gatti$autovalutazione_relazione_con_gatti)
#3 uso i quartili appena calcolati - ma si possono usare anche altri quantili, ad esempio il 20°
e l'80° percentile
# possiamo farlo ordinando tutto il dataframe
ordinato_relazione<-gatti[order(gatti$autovalutazione_relazione_con_gatti),]
str(ordinato_relazione)
table(ordinato_relazione$autovalutazione_relazione_con_gatti)
summary(gatti$autovalutazione_relazione_con_gatti)
autovalutazione<-c(rep(1,21),rep(2,35), rep(3,23))
ordinato_relazione$amiconi<-factor(autovalutazione,levels = c(1:3), labels=c("scarsa relazione",
"media relazione", "buona relazione"))
head(ordinato_relazione$amiconi)
tail(ordinato_relazione$amiconi)
# oppure ordinando solo la nuova variabile
gatti$amiconi[gatti$autovalutazione_relazione_con_gatti<=4]<-"scarsa relazione"
gatti$amiconi[gatti$autovalutazione_relazione_con_gatti>=5&gatti$autovalutazione_relazione_con_g
atti<12]<-"media relazione"
gatti$amiconi[gatti$autovalutazione_relazione_con_gatti>=12]<-"buona relazione"
class(gatti$amiconi)
table(gatti$amiconi)
gatti$amiconi<-factor(gatti$amiconi)
levels(gatti$amiconi)
gatti$amiconi <- ordered(gatti$amiconi,  levels= c("scarsa relazione", "media relazione", "buona
relazione"))
levels(gatti$amiconi)
table(gatti$amiconi)
```

Script AES, Capitolo 3

```
#1
table(gatti$AES_empatia_animali)
```

```
# la distribuzione non è unimodale
median(gatti$AES_empatia_animali); mean(gatti$AES_empatia_animali)
# o più rapidamente:
summary(gatti$AES_empatia_animali)
# o anche:
Desc(gatti$AES_empatia_animali)

#2.
#a
# varianza, devianza e deviazione standard:
var(gatti$AES_empatia_animali);var(gatti$AES_empatia_animali)*78;sd(gatti$AES_empatia_animali)
#b
# il primo quartile l'abbiamo visto nel summary, ma possiamo anche fare:
quantile(gatti$AES_empatia_animali,.25)
gatti$empatia_animali_categoria[gatti$AES_empatia_animali<=137.5]<-"antropocentrici"
gatti$empatia_animali_categoria[gatti$AES_empatia_animali>137.5]<-"non antropocentrici"
prop.table(table(gatti$empatia_animali_categoria))
#3
tapply(gatti$AES_empatia_animali,gatti$cresciuto_animali_domestici,mean)
#verifichiamo se le due distribuzioni hanno un diverso N
table(gatti$cresciuto_animali_domestici)
tapply(gatti$AES_empatia_animali,gatti$cresciuto_animali_domestici,var)
---------------
Script distribuzione normale, Capitolo 5
#1.a
1-.682
.318/2
round(qnorm(p = .159, mean = 100, sd = 15, lower.tail = TRUE))
round(qnorm(p = .159, mean = 100, sd = 15, lower.tail = FALSE))
#1.b
pnorm(q = 80, mean = 100, sd = 15, lower.tail = TRUE)
#1.c
pnorm(q = 110, mean = 100, sd = 15, lower.tail = TRUE)
#1.d
pnorm(q = 95, mean = 100, sd = 15, lower.tail = TRUE)
pnorm(q = 115, mean = 100, sd = 15, lower.tail = TRUE)
.8413447-.3694413
#1.e
pnorm(q = 70, mean = 100, sd = 15, lower.tail = TRUE)
pnorm(q = 85, mean = 100, sd = 15, lower.tail = TRUE)
.1586553-.002275013
#1.f
qnorm(p = .1, mean = 100, sd = 15, lower.tail = FALSE)
#1.g
qnorm(p = .1, mean = 100, sd = 15, lower.tail = TRUE)
#2.a
round(pnorm(q = 22, mean = 19, sd = 3, lower.tail = TRUE),2)
#2.b
round(pnorm(q = 15, mean = 19, sd = 3, lower.tail = FALSE),2)
#2.c
round(pnorm(q = 20, mean = 19, sd = 3, lower.tail = FALSE),2)
#2.d
round(pnorm(q = 25, mean = 19, sd = 3, lower.tail = TRUE),2)
round(pnorm(q = 21, mean = 19, sd = 3, lower.tail = TRUE),2)
.98-.75
```

Script probabilità , Capitolo 6

```
#1.a
(4.50-4.60)/.4
pnorm(q = -.25, mean = 0, sd = 1, lower.tail = TRUE)
#1.b
SE_20<-.40/sqrt(20)
(4.50-4.60)/SE_20
pnorm(q = -1.12, mean = 0, sd = 1, lower.tail = TRUE)
#1.c
SE_50<-.40/sqrt(50)
(4.50-4.60)/SE_50
pnorm(q = -1.77, mean = 0, sd = 1, lower.tail = TRUE)
#2.
SE_30<-.50/sqrt(30)
 (4.50-5.10)/SE_30
pnorm(q = -6.57, mean = 0, sd = 1, lower.tail = TRUE)
#È molto, molto probabile che i lavoratori abbiano ragione.
```

Script fumo e genere, Capitolo 7

```
#proporzioni entro righe - genere
round(prop.table(table(fumo$genere,fumo$outcome_12_mesi),1),3)*100
# oppure proporzioni entro colonne – status paziente
round(prop.table(table(fumo$genere,fumo$outcome_12_mesi),2),3)*100
# e sul totale complessivo
round(prop.table(table(fumo$genere,fumo$outcome_12_mesi)),3)*100
# i maschi astinenti dopo un anno sono ancora più numerosi, ma, purtroppo, entrambi i generi vedono crescere i   fumatori rispetto al breve termine
```

Script fumo e outcome a lungo termine, Capitolo 7

```
table(fumo$Fagerstrom_categorie, fumo$outcome_12_mesi)
odds_alta_dipendenza<-33/32
odds_bassa_dipendenza<-40/21
odds_alta_dipendenza;odds_bassa_dipendenza
print(odds_ratio<-odds_alta_dipendenza/odds_bassa_dipendenza)
#oppure
print(odds_ratio<-(33*21)/(32*40))
#e per la significatività
Desc(table(fumo$Fagerstrom_categorie,fumo$outcome_12_mesi))
#purtroppo, a distanza di un anno l'OR non è più significativo (la signifività è stata verificata nel §7.2.4): la probabilità di smettere di fumare nei due gruppi con diversa dipendenza è la stessa: il vantaggio dei pazienti poco dipendenti diminuisce. Questo potrebbe significare che la bassa dipendenza è un fattore protettivo a breve termine, ma sul lungo termine altri fattori prevalgono e facilitano la ripresa dell'abitudine. Quali suggerimenti avreste per lo psicologo del Centro antifumo?
```

Script fumo e depressione, Capitolo 7

```
#rendiamo dicotomico il fattore Depressione e verifichiamo le frequenze osservate ottenute
table(fumo$Zung_categorie)
levels(fumo$Zung_categorie)[1:2]<-"depresso"
table(fumo$Zung_categorie, fumo$outcome_3_mesi)
# calcoliamo gli odds a tre mesi nei due gruppi
```

```
odds_depresso<-14/11
odds_non_depresso<-72/29
odds_depresso;odds_non_depresso
print(odds_ratio<-odds_depresso/odds_non_depresso)
#oppure
print(odds_ratio<-(14*29)/(11*72))
#e per la significatività:
Desc(table(fumo$Zung_categorie,fumo$outcome_3_mesi))
# a breve termine, la depressione in baseline non ha cambiato significativamente la probabilità
di riuscire a smettere di fumare (ma il campione è fortemente sbilanciato!)
# calcoliamo gli odds a un anno nei due gruppi
table(fumo$Zung_categorie, fumo$outcome_12_mesi)
odds_depresso<-12/13
odds_non_depresso<-61/40
odds_depresso;odds_non_depresso
print(odds_ratio<-odds_depresso/odds_non_depresso)
#velocizziamo usando solo Desc:
Desc(table(fumo$Zung_categorie,fumo$outcome_12_mesi))
#anche a lungo termine, essere o  non essere depresso a inizio trattamento non cambia la probabilità
di riuscire a smettere di fumare.
```

Script convivenza e miagolii, Capitolo 7

```
#prima il miagolio per il cibo
prop.table(table(gatti$vive_con_gatto,gatti$riconosce_miao_cibo),1)*100
mosaicplot(table(gatti$vive_con_gatto,gatti$riconosce_miao_cibo),  col=rainbow(15),  main="cibo",
xlab="vive con gatto", ylab="riconosce miagolio")
cibo<-chisq.test(gatti$vive_con_gatto,gatti$riconosce_miao_cibo)
cibo
cibo$stdres
print(cibo_phi<-sqrt(cibo$statistic/79))
# l'associazione è significativa: secondo i residui standardizzati corretti, chi non vive con un
gatto dà più risposte sbagliate
#di quelle attese in base al caso, e chi vive con un gatto dà più risposte corrette di quelle
#attese in base al caso.
#L'associazione è piuttosto debole.
#poi il miagolio per l'isolamento
prop.table(table(gatti$vive_con_gatto,gatti$riconosce_miao_isolamento),1)*100
mosaicplot(table(gatti$vive_con_gatto,gatti$riconosce_miao_isolamento),          col=rainbow(15),
main="isolamento", xlab="vive con gatto", ylab="riconosce miagolio")
isolamento<-chisq.test(gatti$vive_con_gatto,gatti$riconosce_miao_isolamento)
isolamento
isolamento$stdres
print(isolamento_phi<-sqrt(isolamento$statistic/79))
# l'associazione è significativa: secondo i residui standardizzati corretti, chi non vive con un
#gatto dà più risposte sbagliate
#di quelle attese in base al caso, e chi vive con un gatto dà più risposte corrette di quelle
#attese in base al caso.
#la forza dell'associazione è di media entità.
#poi il miagolio per lo spazzolamento
prop.table(table(gatti$vive_con_gatto,gatti$riconosce_miao_spazzolamento),1)*100
mosaicplot(table(gatti$vive_con_gatto,gatti$riconosce_miao_spazzolamento),       col=rainbow(15),
main="spazzolamento", xlab="vive con gatto", ylab="riconosce miagolio")
spazzola<-chisq.test(gatti$vive_con_gatto,gatti$riconosce_miao_spazzolamento)
```

```
spazzola
spazzola$stdres
print(spazzola_phi<-sqrt(spazzola$statistic/79))
# l'associazione non è significativa: secondo i residui standardizzati corretti, chi non vive #con un gatto dà lo stesso numero di risposte
#giuste e sbagliate di chi vive con un gatto: questa situazione è di difficile riconoscimento #per tutti.
#Coerentemente, la forza dell'associazione è nulla
```

Script gatti e luoghi comuni, Capitolo 7

```
#le zitelle sono donne ($genere== “F”) single ($stato_civile==”single”). Creiamo una nuova #variabile che identifichi correttamente le zitelle distinguendole dalle altre categorie:
table(gatti$genere, gatti$stato_civile)
gatti$genere_stato[gatti$genere=="F"& gatti$stato_civile=="single"]<-"zitella"
gatti$genere_stato[gatti$genere=="M"& gatti$stato_civile=="single"]<-"celibe"
gatti$genere_stato[gatti$stato_civile=="coniugato"]<-"coniugato"
gatti$genere_stato[gatti$stato_civile=="divorziato"]<-"divorziato"
#oppure
gatti$genere_stato2<-ifelse(gatti$genere=="F"& gatti$stato_civile=="single", yes = "zitella",
no=ifelse(gatti$genere=="M"& gatti$stato_civile=="single", yes="celibe",
no=ifelse(gatti$stato_civile=="coniugato", yes = "coniugato", no="divorziato")))

round(prop.table(table(gatti$genere_stato, gatti$vive_con_gatto),1)*100,2)
#effettivamente la maggioranza delle zitelle vive con un gatto, mentre nelle altre categorie i #conviventi con gatti sono una minoranza. Vediamo se la prevalenza è solo casuale.
# zitelle<-chisq.test(gatti$genere_stato, gatti$vive_con_gatto)
zitelle
zitelle$expected
zitelle$stdres
#il test  overall non è significativo, anche se il residuo standardizzato delle zitelle è #prossimo alla significatività, però ci sono troppe celle con valore atteso <5, quindi il #test non è affidabile. Il problema potrebbero essere i divorziati: eliminiamoli.
>gatti2<-subset(gatti, gatti$genere_stato!="divorziato")
nodiv<-chisq.test(gatti2$genere_stato, gatti2$vive_con_gatto)
nodiv
nodiv$expected
nodiv$stdres
#il problema dei requisiti è risolto e il test è ancora non significativo: il luogo comune è, almeno per ora, #smentito.
```

Script depressione, ansia di stato e ansia di tratto, Capitolo 8

```
cor.test(attaccamento$BDI_II_depressione,attaccamento$STAI_stato)
cor.test(attaccamento$STAI_tratto,attaccamento$STAI_stato)
```

Script burden fisico ed emotivo, Capitolo 8

```
#nel campione completo la relazione è positiva, ma non significativa
cor.test(attaccamento$CBI_burden_fisico,attaccamento$CBI_burden_emotivo)
#togliamo prima il soggetto 2, poi anche il soggetto 22
```

```
menodue<-attaccamento[-2,]
cor.test(menodue$CBI_burden_fisico,menodue$CBI_burden_emotivo)
meno_2_22<-attaccamento[c(-2,-22),]
cor.test(meno_2_22$CBI_burden_fisico,meno_2_22$CBI_burden_emotivo)
#il modello è migliorato: la relazione è di intensità più forte ed è significativa, anche se la
varianza condivisa è sempre abbastanza scarsa.
---------------
```

Script empatia e relazione con i gatti, Capitolo 9

```
#prima l'empatia per i pets in generale
empatia_animali<-lm(gatti$autovalutazione_relazione_con_gatti~gatti$AES_empatia_animali)
plot(gatti$AES_empatia_animali,gatti$autovalutazione_relazione_con_gatti,       col=rainbow(15),
xlab="empatia animali", ylab="relazione con gatti", pch=19)
abline(empatia_animali, col="red", lwd=2)
summary(empatia_animali)
confint(empatia_animali)
#la relazione è positiva: per un punto in più di empatia verso gli animali, l'autovalutazione
#della relazione con i gatti cresce
#di .07 punti. la varianza spiegata dal punteggio all'AES è abbastanza ridotta (11.4%; 10.3 in
popolazione)
#In popolazione, la variazione unitaria attesa nell'autovalutazione per ogni punto in più #nell'AES
sta tra .02 punti e .13 punti.
#Il CI non contiene il valore previsto da H0 (il modello è significativo), e la sua ampiezza è
#piuttosto limitata.
#poi l'empatia per i gatti
empatia_gatti<-lm(gatti$autovalutazione_relazione_con_gatti~gatti$empatia_gatti)
plot(gatti$empatia_gatti,gatti$autovalutazione_relazione_con_gatti,             col=rainbow(15),
xlab="empatia verso i gatti", ylab="relazione con gatti", pch=19)
abline(empatia_gatti, col="red", lwd=2)
summary(empatia_gatti)
confint(empatia_gatti)
#la relazione è positiva: per un punto in più di empatia verso i gatti, l'autovalutazione della
#relazione con i gatti cresce
#di circa mezzo punto. la varianza spiegata dal punteggio di empatia è molto alta: 52% nel
#campione e sostanzialmente immutata in
# in popolazione).In popolazione, la variazione unitaria attesa nell'autovalutazione per ogni
#punto in più nell'empatia è compresa tra .38 e .60 punti
#Il CI non contiene il valore previsto da H0 (il modello è significativo), e la sua ampiezza è
limitata; il fit del modello sembra buono.
##Valutiamo la presenza di influential cases; cominciamo con gli outlier bivariati:
summary(rstandard(empatia_gatti))
summary(rstandard(empatia_gatti))
which(rstandard(empatia_gatti)>2)
which(rstandard(empatia_gatti)< (-2))
plot(rstandard(empatia_gatti), col=rainbow(15), pch=19)
abline(h=c(-2,2), lwd=2)
identify(rstandard(empatia_gatti))
gatti[c(14,36,48,66),c(10,9)]
#per quattro soggetti il modello compie errori rilevanti; il soggetto 18 ha molta empatia e ma
#cattiva relazione con i gatti,
#mentre il soggetto 66 ha pochissima empatia e una discreta relazione con i gatti;i soggetti 36
#e 48 hanno una relazione
#con i gatti molto migliore di quella attesa in base alla loro (media) empatia
summary(cooks.distance(empatia_gatti))
```

```
# per fortuna nessuno dei casi è un influential case.
#vediamo i test di specificazione:
plot(empatia_gatti)
shapiro.test(rstandard(empatia_gatti));
t.test(rstandard(empatia_gatti))
#installate lmtest
bptest(empatia_gatti)
dwtest(empatia_gatti)
#la linearità della relazione e l'omoschedasticità dei residui sembrano buone. la distribuzione
#degli errori
#è normale e la loro media non è significativamente diversa da zero. L'autocorrelazione è #assente
```

Script empatia e riconoscimento miagolii, Capitolo 9

```
#prima il campione complessivo:
summary(lm(gatti$totale_riconoscimenti_corretti~gatti$empatia_gatti))
confint(lm(gatti$totale_riconoscimenti_corretti~gatti$empatia_gatti))
#il modello è significativo (F=5.29, p .02), ma la variabilità nel riconsocimento del miagolio
#spiegata dall'empatia è molto scarsa, circa il 6% nel campione e il 5% in popolazione. La
#relazione è positiva, ma per ogni punto in più nell'autovalutazione dell'empatia, il numero di
#riconoscimenti corretti aumenta solo di .04 punti. Il CI non contiene il valore previsto da H0,
#ma ci va davvero vicinissimo
#poi i due subset; si possono creare prima i due sub campioni con la funzione subset, oppure
#usare l'argomento subset di lm():
no<- (lm(gatti$totale_riconoscimenti_corretti~gatti$empatia_gatti, subset =
gatti$vive_con_gatto=="no"))
summary(no)
confint(no)
si<-(lm(gatti$totale_riconoscimenti_corretti~gatti$empatia_gatti, subset =
gatti$vive_con_gatto=="si"))
summary(si)
confint(si)
#in entrambi i casi i modelli non sono significativi: il segno della relazione è positivo, ma la
#variazione unitaria in Y è praticamente zero e i CI contengono entrambi zero. Il fit dei #modelli
nei due campioni è cattivo (solo il 6.7% di varianza spiegata nel primo caso e solo #l'1% nel
secondo).
#perchè nel campione complessivo abbiamo ottenuto una significatività della relazione, #nonostante
il fit di quel modello fosse addirittura peggiore del fit del modello "no" (6.7% #versus 6.4% di
varianza spiegata), che non è significativo?
#Suggerimento: guardate i df dei due modelli...
```

INDICE DELLE FUNZIONI E DEI PACKAGE USATI NELLA DISPENSA

In blu sono indicate le **funzioni**, in rosso i **package**. È riportato il numero di pagina in cui la funzione compare per la **prima volta**, o delle pagine in cui compare in **contesti differenti**. A fianco delle funzioni, **solo** per quelle **non** disponibili nel package base, è riportato il package che le contiene.