Università degli Studi di Trento

Dipartimento di Psicologia e Scienze Cognitive

Corso di Laurea in Scienze e Tecniche di Psicologia Cognitiva

Corso di

Analisi dei dati con applicazioni informatiche

Statistica descrittiva

Dispensa – Statistica Descrittiva

Ottavia M. Epifania

ottavia.epifania@unitn.it

Nota sul materiale

Parte del materiale di questa dispensa è riadattato dal corso *Test per le organizzazioni* dell'Università degli Studi di Padova, tenuto insieme alla Dott.ssa Margherita Calderan (https://ottaviae.github.io/test-organizzioni-fisppa/).

Per l'organizzazione del materiale e per la generazione dei grafici e dei dati utilizzati nelle esercitazioni è stato utilizzato Claude (Anthropic), un assistente basato su intelligenza artificiale. Tutti i contenuti sono stati rivisti e verificati dall'autrice, che ne è l'unica responsabile.

Licenza

Quest'opera è distribuita con licenza Creative Commons Attribuzione 4.0 Internazionale (CC BY 4.0). Puoi copiare, ridistribuire e modificare il materiale con qualsiasi mezzo e per qualsiasi fine, a condizione di citare l'autrice e di indicare se sono state apportate modifiche.

Per citare questa dispensa:
Epifania, O. M. *Statistica descrittiva. Dispensa del corso Analisi dei dati con applicazioni informatiche.* Università degli Studi di Trento.

Indice

1 Modulo 1 – Introduzione a `R` 4
- 1.1 RStudio . . . 4
- 1.2 Working directories (`wd`) . . . 5
- 1.3 Script e reporting . . . 7
- 1.4 Reporting con quarto . . . 8
- 1.5 Elementi base . . . 11
- 1.6 Pacchetti e Funzioni . . . 14
- 1.7 Operatori . . . 17
- 1.8 Tavole di verità . . . 18
- 1.9 R ed errori . . . 18
- 1.10 Vettori . . . 20
- 1.11 Matrici . . . 24
- 1.12 Dataframe . . . 27
- 1.13 Liste . . . 30
- 1.14 Importare dati esterni . . . 31
- 1.15 Sintesi delle strutture tabellari . . . 35

2 Modulo 2 – Scale di misura e statistica descrittiva univariata 38
- 2.1 Tipi di variabili . . . 38
- 2.2 Scale di misura (Stevens, 1946) . . . 38
- 2.3 Frequenze . . . 44
- 2.4 Indici di posizione . . . 46
- 2.5 Indici di variabilità . . . 51
- 2.6 Perché i grafici . . . 53
- 2.7 Grafici . . . 54
- 2.8 Esempi . . . 60

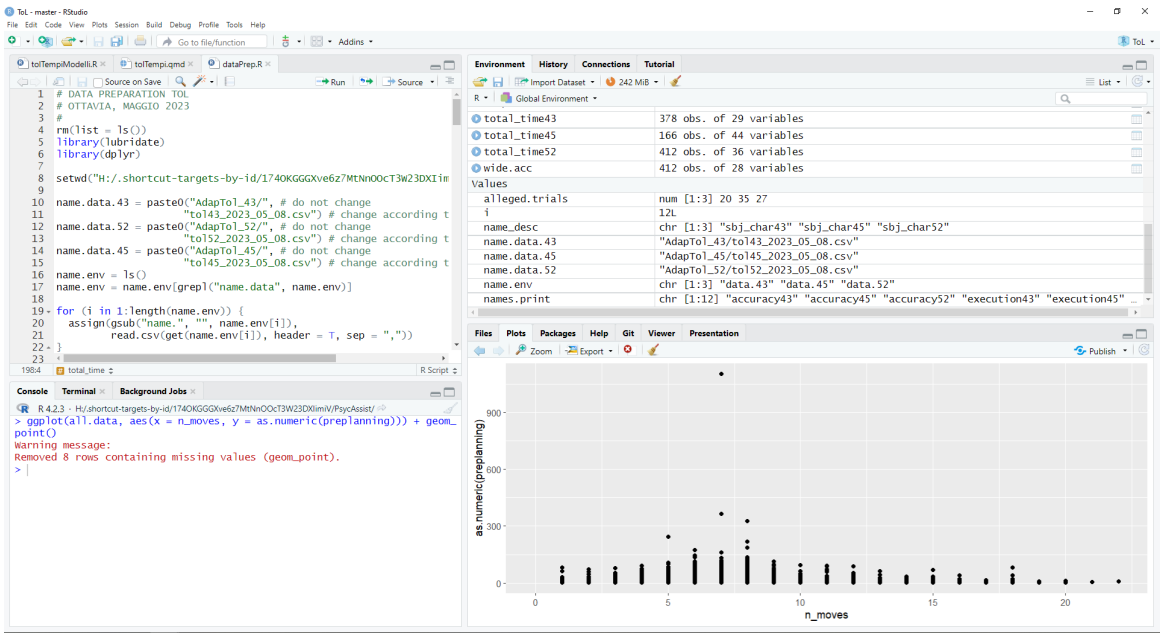
3 Modulo 3 – Statistica descrittiva bivariata 64
- 3.1 Variabile statistica doppia . . . 64
- 3.2 Connessione (Variabili qualitative) . . . 71
- 3.3 Indipendenza stocastica e massima dipendenza . . . 74
- 3.4 Associazione (variabili quantitative) . . . 76
- 3.5 Previsione variabili quantitative . . . 86
- 3.6 Materiale supplementare . . . 99

4 Funzioni riassuntive dei moduli 103
- 4.1 Operatori . . . 103
- 4.2 Valori di verità . . . 104

4.3 Operatori e funzioni utili . . . 104
4.4 Convertitori . . . 104
4.5 Strumenti . . . 105
4.6 Funzioni matematiche . . . 105
4.7 Matrici . . . 106
4.8 Liste e Data Frame . . . 106
4.9 Distribuzioni di frequenza . . . 107
4.10 Funzioni grafiche . . . 107
4.11 Associazione variabili categoriali . . . 107
4.12 Regressione lineare . . . 107

1 Modulo 1 – Introduzione a R

1.1 RStudio



Ambiente/Environment (Pannello in alto a dx). Contiene tutti gli oggetti che sono stati creati fino a quel momento nell'ambiente di lavoro. Viene automaticamente suddiviso in oggetti di tipo vettoriale o scalare (`values`) e oggetti più complessi come dataframe e liste.

Script. (Pannello in alto a sinistra). File di tipo testuale con estensione `.R` dove salvare il codice usato per generare gli oggetti presenti nell'ambiente. Si possono combinare codice e commenti (righe che non vengono eseguite e interpretate come codice, sono le righe precedute dal simbolo `#`, vedi Listing 1.1). Per eseguire i comandi negli script bisogna usare la combinazione di tasti `ctrl + invio` (per il mac, `cmd + invio`).

Console. (Pannello in basso a sinistra). Pannello di lavoro dove i comandi vengono eseguiti immediatamente quando si preme invio. Le righe di comando non vengono salvate ma sono sempre accessibili premendo la freccina in su della tastiera. Nella console appaiono anche i risultati dei codici eseguiti nello script.

Plot/Files. (Pannello in basso a dx). Pannello composto da diversi pannelli. Nel pannello `Plots` compaiono i grafici risultanti dalle analisi statistiche svolte negli script. Nel pannello

`Files` appaiono tutti i file e le cartelle disponibili in una data working directory (Vedi Section 1.2 e Section 1.2.2).

Listing 1.1 Codice con commento

```
# codice per assegnare il numero 25 alla variabile x
x <- 25
```

Console

I comandi nella console vengono eseguiti con click su Invio
Il codice non viene salvato e scompare sequenzialmente.
L'output appare immediatamente nella console.
Per fare lo switch alla console → `ctrl + 2`

Script

Si possono salvare gli script di codice
Per far "correre" qualsiasi comando nello script → Ctrl + Invio (o `cmd + Enter`)
L'output appare nella console.
Per fare lo switch allo script → `ctrl + 1`

1.2 Working directories (`wd`)

Working directory. La cartella dentro cui sta lavorando R in quel momento. Diventa di fondamentale importanza perché è dove R si aspetta che ci siano i file (e.g., i dati) con cui volete lavorare

1.2.1 Hard way

Listing 1.2 Funzione per ottenere la wd, ovvero la cartella dentro cui sta lavorando R

```
# per sapere dove sta lavorando R
getwd()
```

```
[1] "C:/Users/Ottavia/Documents/GitHub/stpc-analisi-dati"
```

La `wd` si può cambiare con il comando `setwd()`:

Listing 1.3 Funzione per cambiare la wd dentro cui sta lavorando R. Dentro la funzione, va messo il percorso completo entro cui si vuole spostare la cartella di lavoro tra virgolette, come nell'esempio sottostante.

```
setwd("C:/Users/Ottavia/Documents/altro-percorso")
```

1.2.2 Easy Way: R Project

Crea automaticamente una working directory (ovvero una cartella) in cui potete trovare tutto il materiale necessario per delle analisi. Questo vi evita di dover pensare alla gestione delle working directories.

Struttura tipica

```
my-project/
|
+-- index.qmd
+-- laboratorioTemplate.qmd
+-- myScript.R
+-- my-project.Rproj
+-- data/
|   +-- mydata.csv
```

Nell'esempio precedente, la cartella di ordine superiore è la cartella `my-project`, ovvero la cartella che contiene tutto il vostro progetto. Al suo interno si trovano i seguenti file, a diversi livelli:

1. Primo Livello: `index.qmd`, `laboratorioTemplate.qmd`, `myScript.R` (file di quarto con estensione `qmd`, file R con estensione `R`) e una cartella `data`. Il file `my-project.Rproj` è il file del vostro progetto R con estensione `Rproj`.
2. Secondo Livello: all'interno della cartella `data` si trova il file `mydata.csv`. Per accedere a questo file di dati da un file contenuto nella cartella di ordine superiore, bisogna semplicemente comunicare a R che si trova nella cartella data, `"data/mydata.csv"`.

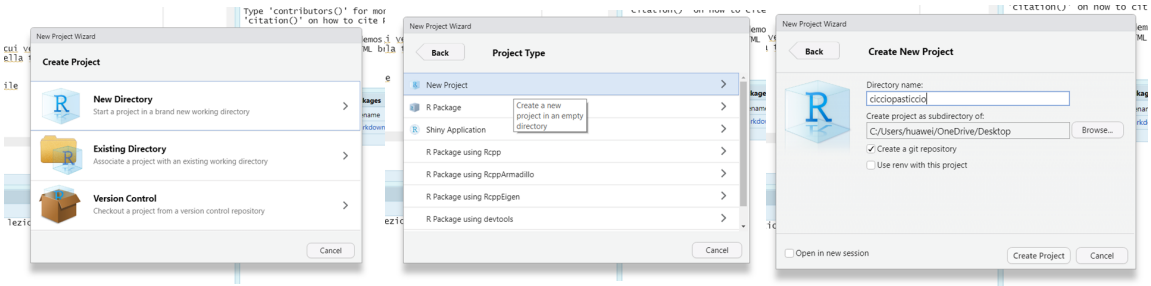
1.2.2.1 Creare un nuovo progetto

Da `File` → New Project e seguire i seguenti step:

In Figure 1.1a selezionare l'opzione `New directory`.

In Figure 1.1b selezionare l'opzione `New Project` (ignorate tutte le altre opzioni).

In Figure 1.1c date un nome al progetto (e.g., `corso-analisi`) e scegliete la cartella entro cui salvarlo (consigliato: I Documenti del vostro computer).



(a) Step 1 (b) Step 2 (c) Step 3

Figure 1.1: Creare un progetto R in 3 step

Figure 1.2 illustra il controllo per capire se effettivamente siete dentro al progetto R. Se non siete dentro il progetto R, andate a cercarlo nella cartella dove lo avete creato e fate doppio click sul file con estensione `Rproj`.

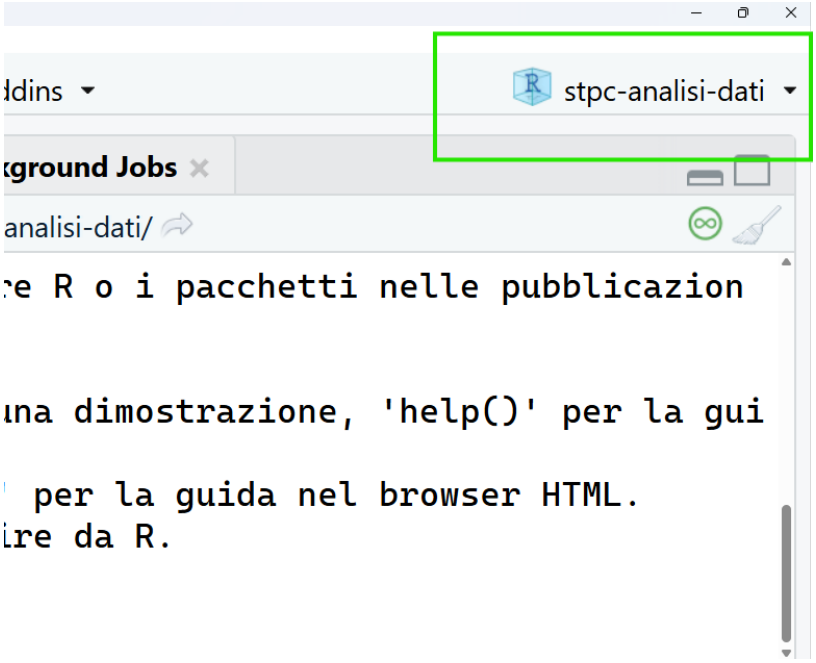


Figure 1.2: Indica il nome del progetto

1.3 Script e reporting

Figure 1.3 illustra come creare un nuovo script, dove è possibile eseguire e salvare una serie di comandi.

In alternativa, si può usare la sequenza di tasti: `Ctrl + shift + n` (`cmd + shift + n`)

Per salvare uno script, basta cliccare sull'icona del salvataggio.

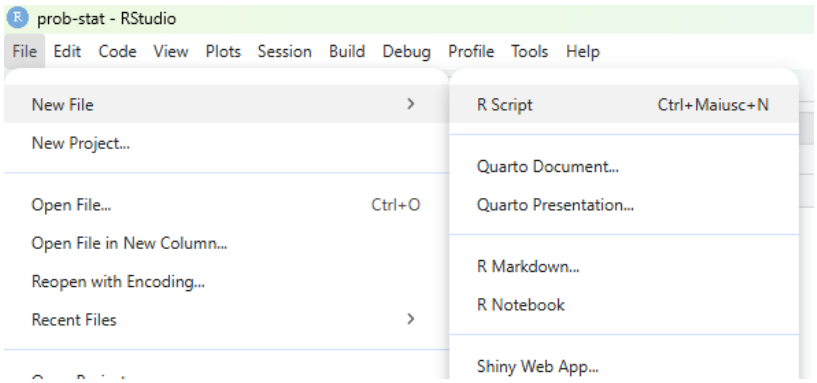


Figure 1.3: Creare un nuovo script

1.4 Reporting con quarto

quarto è un markup language integrato con RStudio.

Permette la creazione di documenti (sia pdf sia html) che possono contenere chunk di codice, ovvero delle parti dove viene eseguito il codice R, per ottenere risultati, tabelle e grafici.

In questo corso, useremo quarto per tutti gli esercizi. Vi verranno forniti dei template, specie all'inizio, nella speranza che riusciate a modificarli e a diventare via via più autonome nel suo utilizzo.

Una parte dell'esame prevede l'utilizzo di quarto per lo svolgimento degli esercizi. Le esercitazioni in itinere sono da svolgere su quarto.

A questo link trovate maggiori informazioni su quarto e una guida veloce sul suo funzionamento.

Figure 1.4a crea un nuovo file quarto, mentre Figure 1.4b illustra come specificare varie opzioni del documento (titolo, autrice).

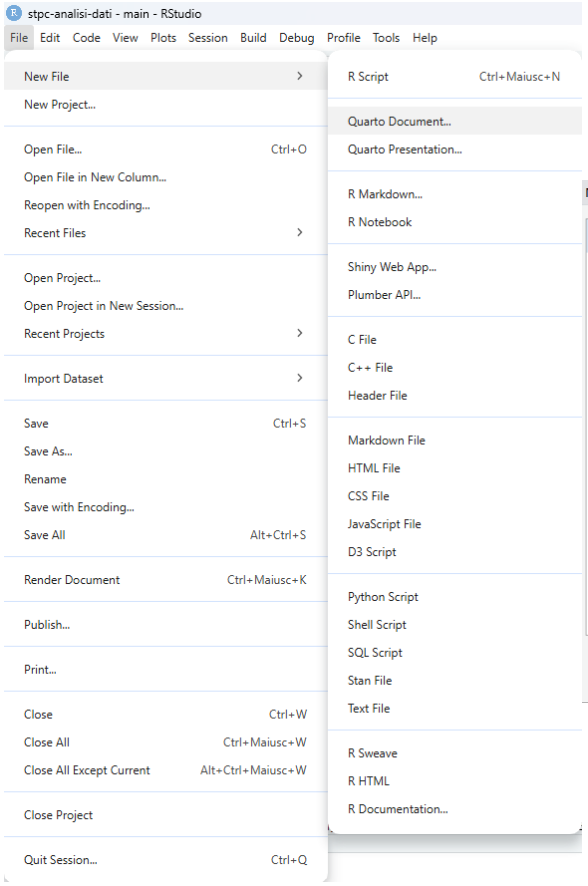
Figure 1.5 vi mostra il default che si ottiene quando si crea un documento quarto.

In verde in Figure 1.5, lo YAML che controlla l'aspetto generale del documento. Va modificato per cambiare i metadati del documento.

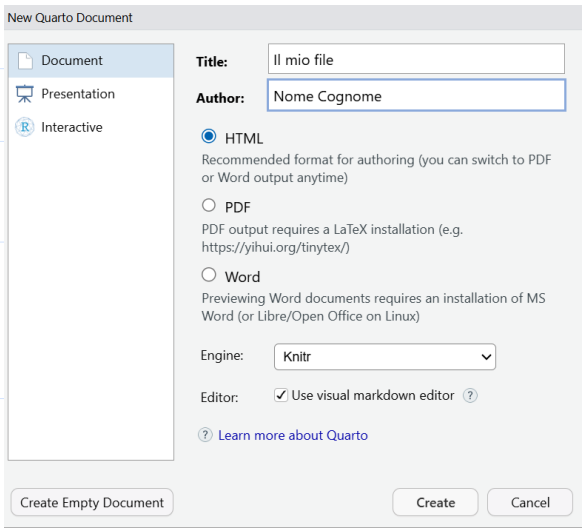
In rosso in Figure 1.5, un chunk di codice eseguibile, ovvero una parte del documento che funziona come un qualsiasi script di R e in quanto tale permette di eseguire codice.

> Modifica importante allo YAML
>
> La parte `embed-resources: true` è fondamentale (e va scritta esattamente seguendo l'indentazione riportata). Serve a fare in modo che tutti i codici, i risultati e i grafici prodotti nel documento rimangano nel documento quando lo compilate in html.

Figure 1.4: Crea un nuovo file quarto



(a) Crea il file



(b) Selezione tipo di file/dati

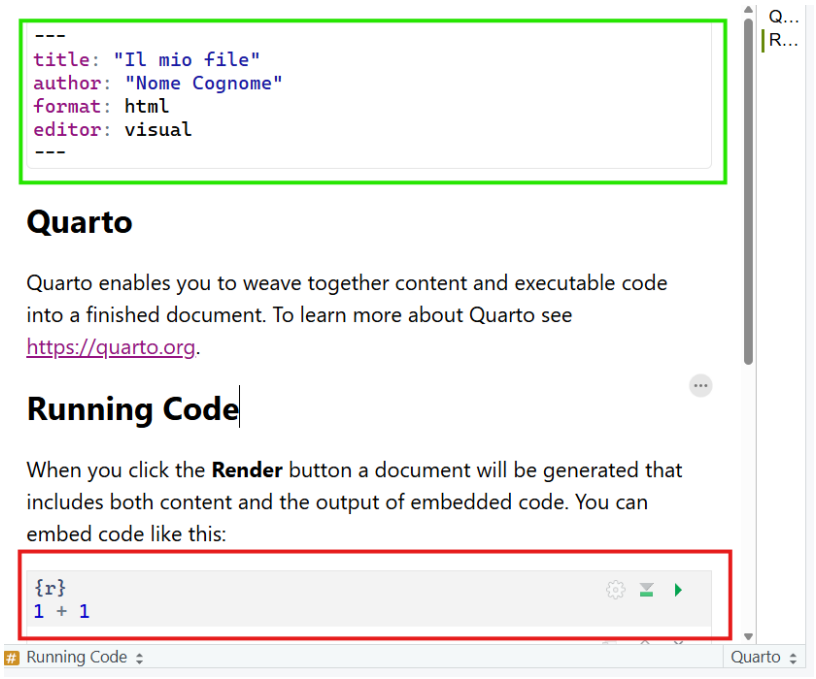


Figure 1.5: Default documento di quarto. In verde lo YAML, in rosso un chunk di codice eseguibile come qualsiasi script R

```
---
title: "Il mio file"
author: "Nome Cognome"
format:
  html:
    embed-resources: true
editor: visual
---
```

Il chunk di codice funziona esattamente come R – è sequenziale, va seguito l'ordine con cui vengono fatte le operazioni

Ogni volta che viene fatto il rendering del documento (`shift + ctrl + k` o freccina azzurra), viene creato un ambiente temporaneo dove viene eseguito il codice.

L'ambiente temporaneo è indipendente dall'ambiente che vedete su R!

Per creare un nuovo chunk: `shift + ctrl + i` oppure Figure 1.6:

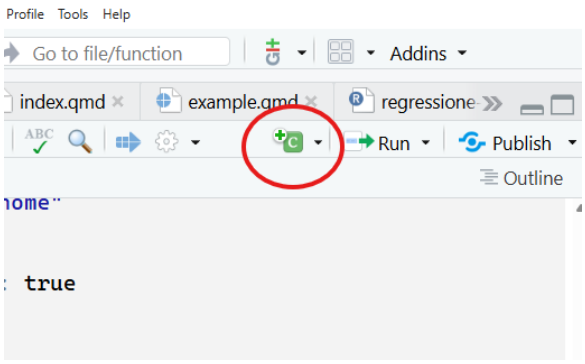


Figure 1.6: Creare un nuovo chunk di codice

1.5 Elementi base

1.5.1 Calcolatrice

```
3 + 2   # più
3 - 2   # meno
3 * 2   # per
3 / 2   # diviso
```

Parentesi e simili:

```
() [] {} "" : ; ,
```

ATTENZIONE

Nei chunk di codice che seguono vedrete spesso l'operatore ; scritto sulla stessa riga di codice dove viene eseguita un'altra operazione, come ad esempio:

```
variabile = 1:5 ; variabile
```

```
[1] 1 2 3 4 5
```

equivale a scrivere

```
variabile = 1:5
variabile
```

```
[1] 1 2 3 4 5
```

1.5.2 Assign (Crea oggetti)

Tutto quello che viene creato in R viene chiamato *oggetto* tramite l'"assegnazione" con specifici simboli, ovvero `<-` oppure `=`:

```
x <- 4
x
```

```
[1] 4
```

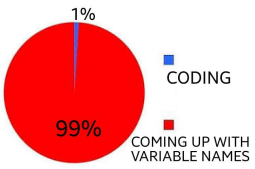
```
X = 5
X
```

```
[1] 5
```

Gli elementi a destra del simbolo (contenuto) vengono assegnati all'elemento a sinistra (contenitore) del simbolo

> Attenzione!
>
> R è case sensitive. `x` e `X` sono due oggetti diversi!

1.5.3 Nominare gli oggetti



Nomi validi sono lettere, numeri, punti, underscore (e.g., `variable_name`)

Buoni nomi per le variabili:

- `mean_rt`
- `sum_age`
- `prop_item`

Pessimi nomi per variabili:

- `x`
- `y`
- `ciccio`

Non deve contenere caratteri speciali come #, &, $, ?, etc.

Non deve essere una parola riservata ovvero quelle parole che sono utilizzate da R con un significato speciale (`?reserved` nella console).

```r
.01 = 5
```

```
Error in 0.01 = 5: membro di sinistra dell'assegnazione (do_set) non valido
```

```r
1 = "a"
```

```
Error in 1 = "a": membro di sinistra dell'assegnazione (do_set) non valido
```

```r
TRUE = 567
```

```
Error in TRUE = 567: membro di sinistra dell'assegnazione (do_set) non valido
```

1.5.3.1 Per convenzione

Queste istruzioni sono valide attraverso diversi linguaggi di programmazione.

Fanno riferimento a nomi di oggetti composti da più nomi e/o molto lunghi:

1. `first_data`, ovvero separare i termini con un underscore (e.g., `my_data`)
2. `firstData`, ovvero la prima parola tutta minuscola, la seconda con la prima lettera maiuscola (e.g., `unitnData` invece di `university_of_trento_data`)

1.5.4 L'ambiente

Gli oggetti finiscono nell'ambiente (`environment`) di R.

La funzione `ls()` permette di vedere tutti gli oggetti all'interno dell'ambiente.

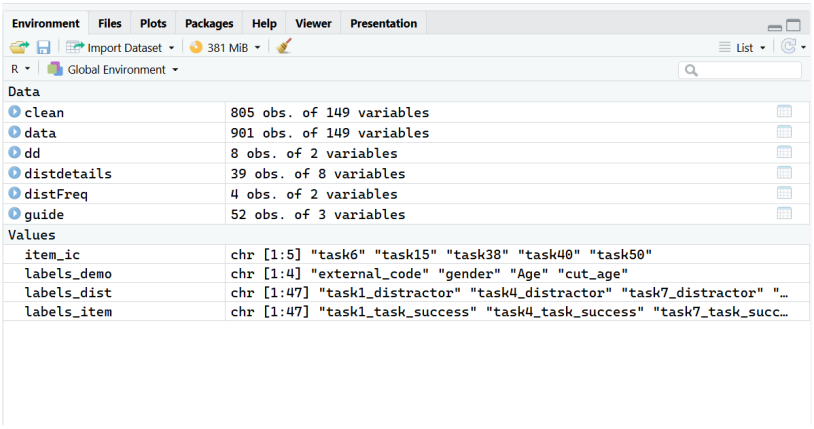


Figure 1.7: Ambiente di R

`rm()` (remove) è la funzione preposta per rimuovere gli oggetti dall'ambiente di R: `rm("oggetto")` (vedi Listing 1.6).

Il codice `rm(list = ls())` permette di eliminare tutti gli oggetti nell'ambiente in un colpo solo (equivale alla scopetta).

Listing 1.4 Ottenere gli elementi nell'ambiente e uso della funzione rm

```
ls() # ottiene tutti gli oggetti nell'ambiente
```

```
[1] "variabile" "x"          "X"
```

Listing 1.5 Ottenere gli elementi nell'ambiente e uso della funzione rm

```
rm(variabile) # rimuove l'oggetto variabile
ls()
```

```
[1] "x" "X"
```

Listing 1.6 Ottenere gli elementi nell'ambiente e uso della funzione rm

```
rm(list = ls()) # rimuove tutto l'ambiente
ls()
```

```
character(0)
```

1.6 Pacchetti e Funzioni

1.6.1 Funzioni

Ogni cosa fatta in R di fatto è riassumibile nelle *operazioni* fatte su *oggetti*

Le operazioni sono le **funzioni**, che permettono di modificare gli oggetti esistenti e di crearne di nuovi.

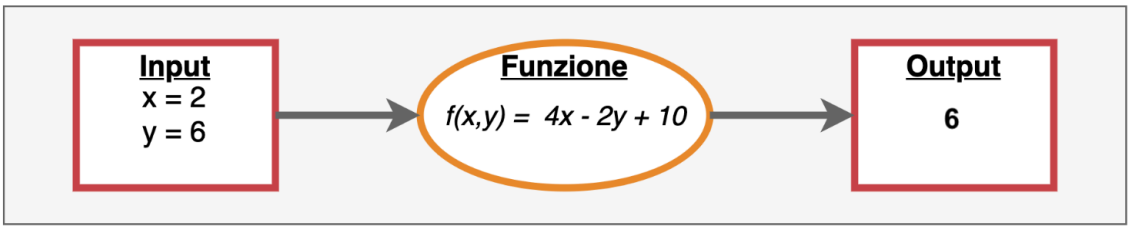
Ad esempio, in Listing 1.7 viene usata la funzione `sum()` per sommare tutti gli elementi all'interno del vettore. Matematicamente, la funzione `sum()` equivale a $\sum_{i=1}^{n} x_i$ dove n è il numero di elementi x_i all'interno del vettore X.

Listing 1.7 Applica la funzione somma per sommare tutti gli elementi all'interno di vettore

```
vettore = 1:5
sum(vettore)
```

```
[1] 15
```

Le funzioni in R sono come le funzioni matematiche $f(\cdot)$, dove viene preso un input, viene elaborato secondo le regole della funzione e si ottiene un output:



1.6.1.1 Elementi di una funzione

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

- Nome della funzione: `mean`
- Parentesi che racchiudono gli argomenti della funzione
- argomenti:
  - **obbligatori**, senza valori predefiniti senza di cui la funzione non può funzionare. Nel caso della funzione `mean()`, l'unico argomento obbligatorio è il vettore `x` di valori di cui si vuole calcolare la media
  - **facoltativi** con valori di default (`trim = 0`, `na.rm = FALSE`), che vengono applicati qualora l'utente non ne specifichi di alternativi. Nel caso della media, non viene applicato il trimming dei valori estremi e non vengono tolti i valori NA.

```
mean(vettore)
```

```
[1] 3
```

```
mean(vettore, trim = 0.3)
```

```
[1] 3
```

Listing 1.8 Permette di accedere all'help della funzione mean

```
?mean
```

`?funzione` (Listing 1.8) permette di accedere all'help della funzione per scoprire esattamente cosa fa una funzione:

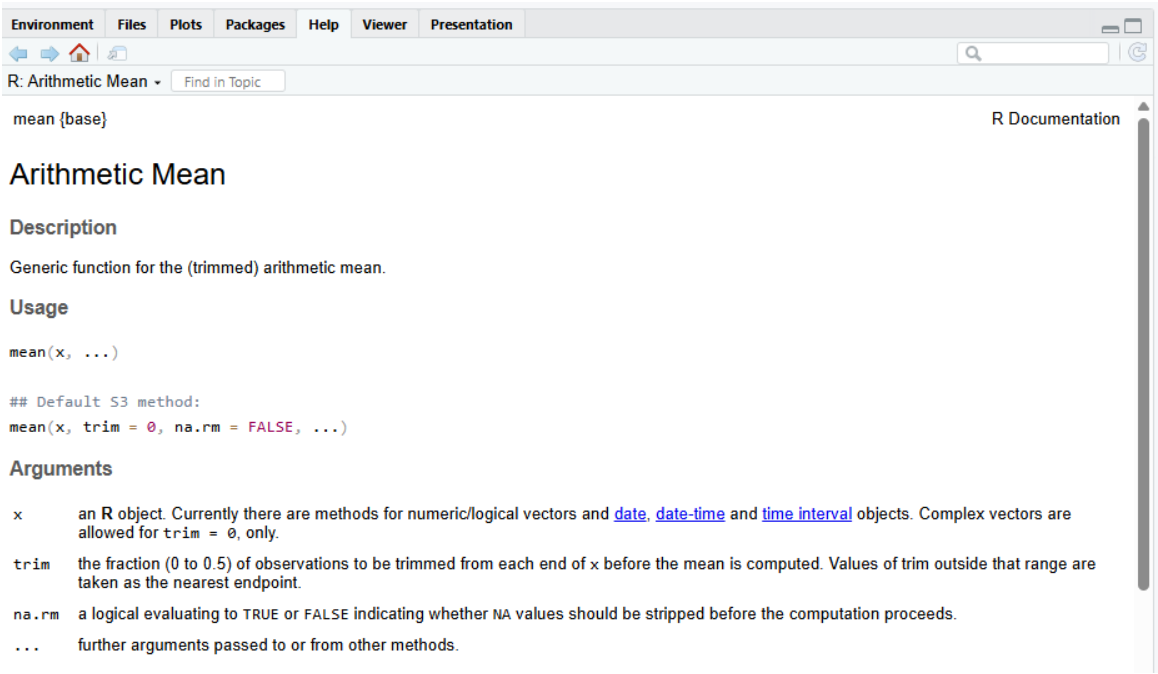


Figure 1.8: Help della funzione mean

1.6.1.2 Installare i pacchetti

Esistono funzioni predefinite all'installazione di R, se ne possono definire di nuove autonomamente, si possono installare dei "pacchetti" dal CRAN (comprehensive R archive network, vedi Listing 1.9) o da un file locale, a seconda delle operazioni che si vogliono svolgere.

Listing 1.9 Codice per installare e caricare un pacchetto

```
install.packages("nomePacchetto") # installa (va fatto correre solo una volta)
library(nomePacchetto) # carica il pacchetto (va fatto correre tutte le volte)
```

1.7 Operatori

1.7.1 Operatori Matematici

Tabella 1.1 illustra alcuni operatori matematici di base, mentre Tabella 1.2 illustra le principali funzioni matematiche disponibili in R.

Table 1.1: Operatori matematici in R

Funzione	Cosa fa?	Esempio	Risultato
`+`	addizione	`5.4 + 6.1`	`11.5`
`-`	sottrazione	`9 - 4.3`	`4.7`
`*`	moltiplicazione	`7 * 1.4`	`9.8`
`/`	divisione	`9/3`	`3`
`%%`	resto	`9%%2`	`1`
`^`	potenza	`15 ^ 2`	`225`

Table 1.2: Funzioni matematiche in R

Funzione	Cosa fa?	Esempio	Risultato
`abs()`	valore assoluto	`abs(-8)`	`8`
`sqrt()`	radice quadrata	`sqrt(225)`	`15`
`exp()`	funzione esponenziale	`exp(0)`	`1`
`log()`	logaritmo naturale	`log(1)`	`0`
`round()`	arrotondamento, intero	`round(1.738)`	`2`
`round()`	arrotondamento	`round(1.738, 2)`	`1.74`

1.7.2 Operatori Relazionali

In R è possibile valutare se una data relazione è vera o falsa. R valuterà le proposizioni e ci restituirà il valore **`TRUE`** se la proposizione è vera oppure **`FALSE`** se la proposizione è falsa. Per fare questo si usano gli operatori relazionali in Tabella 1.3.

1.7.3 Insiemistici

Per capire gli operatori insiemistici (Tabella 1.4), definiamo un oggetto, `x`, come segue:

```
x = 30
```

Gli operatori insiemistici permettono di testare diverse condizioni.

La **congiunzione** testa che siano contemporaneamente vere entrambe le condizioni, ovvero restituisce un risultato vero se e solo se entrambe le condizioni sono vere.

Table 1.3: Operatori relazionali in R

Funzione	Nome	Esempio	Risultato
`==`	uguale	`30 == 30`	`TRUE`
`!=`	diverso	`30 != 30`	`FALSE`
`>` / `>=`	maggiore / maggiore o uguale	`30 > 10`	`TRUE`
		`30 >= 10`	`TRUE`
`<` / `<=`	minore / minore o uguale	`30 < 10`	`FALSE`
		`10 <= 10`	`TRUE`
`%in%`	inclusione	`10 %in% c(1, 2, 10)`	`TRUE`

Table 1.4: Operatori logici in R (con x = 30)

Funzione	Nome	Esempio	Risultato
`&`	Congiunzione	`x > 25 & x < 60`	`TRUE`
`|`	Disgiunzione inclusiva	`x > 25 | x > 60`	`TRUE`
`!`	Negazione	`!(x < 18)`	`TRUE`

La **disgiunzione** testa che sia vera almeno una delle condizioni, ovvero restituisce un risultato vero se almeno una delle condizioni è vera.

La **negazione** inverte il valore logico.

1.8 Tavole di verità

Operazione	Simbolo	In R	Significato
Congiunzione (AND)	$A \cap B$	`&`	Vera se entrambe le condizioni sono vere Table 1.6a.
Disgiunzione (OR)	$A \cup B$	`|`	Vera se almeno una delle due condizioni è vera Table 1.6b.
Negazione (NOT)	$\overline{A}$	`!`	Inverte il valore logico di A Table 1.6c.
Differenza simmetrica (XOR)	$A \Delta B$	`xor(A, B)`	Vera se una e una sola delle due condizioni è vera Table 1.6d.

Lo stesso risultato della funzione `xor()` può essere ottenuto con la combinazione di diversi operatori logici (ed è la strada da preferire)

1.9 R ed errori

Ci sono diversi livelli di **allerta** quando scriviamo codice:

- **`messaggi`**: la funzione ci restituisce qualcosa che è utile sapere, ma tutto liscio. Appaiono spesso quando si caricano dei pacchetti.
- **`warnings`**: la funzione ci informa di qualcosa di *potenzialmente* problematico, ma (circa) tutto liscio. R ci sta dicendo che è successo qualcosa di insolito e di prestare attenzione, ma non per forza è un problema.
- **`error`**: la funzione non solo ci informa di un **errore** ma le operazioni richieste non sono stato eseguito. Solitamente il messaggio di errore ci dice anche cosa sta succedendo

Table 1.6: Tavole di verità

(a) Congiunzione

A	B	$A \cap B$
V	V	V
V	F	F
F	V	F
F	F	F

(b) Disgiunzione

A	B	$A \cup B$
V	V	V
V	F	V
F	V	V
F	F	F

(c) Negazione

A	$\overline{A}$
V	F
F	V

(d) Differenza Simmetrica

A	B	$A \Delta B$
V	V	F
V	F	V
F	V	V
F	F	F

1.9.0.1 Esempi di errori

```
mean(myVector)  #
```

```
Error: oggetto 'myVector' non trovato
```

```
dati = read.csv("i-miei-dati.csv")
```

```
Warning in file(file, "rt"): non è possibile aprire il file 'i-miei-dati.csv':
No such file or directory
```

```
Error in file(file, "rt"): non è possibile aprire la connessione
```

```
lenght(vettore)
```

```
Error in lenght(vettore): non trovo la funzione "lenght"
```

```
(3,4,5)
```

```
Error in parse(text = input): <text>:1:3: ',' inatteso
1: (3,
      ^
```

1.9.0.2 R e l'uso dell'IA

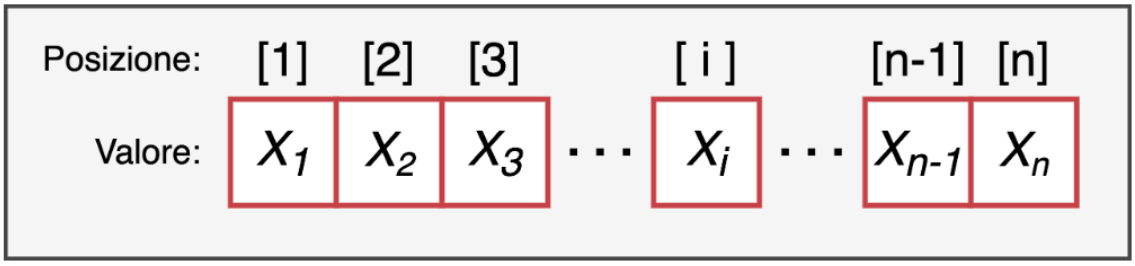
I prompt devono essere il più specifici possibile e centrati su quello che volete fare e come lo volete fare.

Dovete comunicare chiaramente che volete sempre la soluzione in R base, possibilmente senza usare funzioni aggiuntive e dovete sempre chiedere che vi spieghi il codice.

> Warning
>
> All'esame non potete usare l'IA, ma la potete usare per prepararvi e per aiutarvi nello studio

1.10 Vettori

I vettori sono una struttura dati unidimensionale e sono la più semplice presente in R.



Caratteristiche:

- **la lunghezza**: il numero di elementi da cui è formato il vettore. La funzione `length(x)` permette di ottenere la lunghezza del vettore x, ovvero permette di conoscere il numero di elementi di cui è composto il vettore.
- **la tipologia**: la tipologia di dati da cui è formato il vettore. Un vettore infatti deve essere formato da **elementi tutti dello stesso tipo** (vedi Section 1.10.2)!

I vettori sono caratterizzati da:

- **un valore**: il valore dell'elemento che può essere di qualsiasi tipo ad esempio un numero o una serie di caratteri.
- **un indice di posizione**: un numero intero positivo che identifica la sua posizione all'interno del vettore.

1.10.1 Creare un vettore con `c()`

Concatena diversi oggetti insieme per creare un nuovo oggetto unico

```
x = c(1, 2, 3) ; x
```

```
[1] 1 2 3
```

```
X = 1:3 ; X
```

```
[1] 1 2 3
```

```
x == X
```

```
[1] TRUE TRUE TRUE
```

1.10.2 Tipologie di vettore

Tipi di oggetti diversi → Tipi di vettori diversi:

- `num` (`numeric`): Numeri `(-0.56, 3, 4.5)`
- `logi` (`logic`): Logico (`TRUE` vs `FALSE`)
- `chr` (`character`): characters `("a", "b", "c")`
- `factor` (`factor`): factor con diversi livelli

La distinzione più importante è tra vettori di tipo categoriale (`chr` e `factor`) e vettori di tipo numerico (`int` e `num`):

- `integer` e `numeric`: Operazioni matematiche E calcolo delle frequenze
- `character` e `factor`: Calcolo delle frequenze

```
months = c(5, 6, 8, 10, 12, 16); months
```

```
[1]  5  6  8 10 12 16
```

```
weight = runif(6, 3, 18) ; weight
```

```
[1] 14.902078 15.576018 17.067091  8.988758 17.594012  7.397969
```

I vettori logici si ottengono testando delle condizioni sulla base dei valori di altri vettori (Listing 1.10):

Listing 1.10 Testa se ci sono valori maggiori di 12 all'interno del vettore

```
months > 12
```

```
[1] FALSE FALSE FALSE FALSE FALSE  TRUE
```

Il vettore logico risultante può essere utilizzato per filtrare i valori dal vettore originale (vedi Section 1.10.3).

La funzione `rep()` in Listing 1.11 crea un vettore di tipo character ripetendo i K valori per X volte. Nell'esempio in Listing 1.11, le categorie `low`, `medium`, `high` vengono ripetute tutte per due volte (argomento `each = 2`).

La funzione `factor()` forza il vettore ad essere un vettore con K categorie e un numero di valori uguale al numero di elementi dentro il vettore.

Listing 1.11 Creare un vettore di tipo factor usando la funzione rep

```
ses = factor(rep(c("low", "medium", "high"), each = 2))
ses
```

```
[1] low    low    medium medium high   high
Levels: high low medium
```

L'argomento `levels` all'interno della funzione `factor()` (vedi Listing 1.12) permette di specificare l'ordine delle categorie, altrimenti il default di R le mette in ordine alfabetico.

Listing 1.12 Ordinare i livelli della variabile con levels

```
ses = factor(ses, levels = c("low", "medium", "high"))
```

1.10.3 Indicizzare i vettori

I vettori si indicizzano usando le parentesi quadre:

`nome_vettore[indice]` dove l'indice è la posizione all'interno del vettore che si vuole ottenere.

In R, la numerazione delle posizioni all'interno del vettore va da 1 alla lunghezza del vettore.

Dato ad esempio il vettore `nomi` come segue

```r
nomi = c("Pasquale", "Egidio", "Debora", "Luca", "Andrea")
nomi
```

```
[1] "Pasquale" "Egidio"   "Debora"   "Luca"     "Andrea"
```

Possiamo immaginare ogni nome come occupante di una specifica posizione all'interno del vettore, come segue:

Pasquale	Egidio	Debora	Luca	Andrea
1	2	3	4	5

Per cui possiamo estrarre i vari nomi facendo riferimento al loro indice:

`nomi[1]` → Pasquale

`nomi[3]` → Debora

`nomi[seq(2, 5, by = 2)]` → Egidio, Luca

`seq(min, max, by)`

Questa funzione genera un vettore partendo da un valore minimo fino a un valore massimo con uno step incrementale definito dall'utente:

```r
seq(1,5, by = 0.5)
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

L'indicizzazione dei vettori funziona anche su vettori di tipo numerico, come negli esempi seguenti:

```r
weight[2] # secondo elemento di weight
```

```
[1] 15.57602
```

```r
(weight[6] = 15.2) # sostituisce il sesto elemento di weight
```

```
[1] 15.2
```

```r
weight[seq(1, 6, by = 2)] # elementi 1, 3, 5
```

```
[1] 14.90208 17.06709 17.59401
```

```r
weight[2:6] # elementi dal 2 al 6 (incluso)
```

```
[1] 15.576018 17.067091  8.988758 17.594012 15.200000
```

```r
weight[-2] # rimuove il secondo elemento
```

```
[1] 14.902078 17.067091  8.988758 17.594012 15.200000
```

I vettori logici possono essere utilizzati per estrarre determinati valori da un vettore:

```r
weight > 7
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE
```

Restituisce un vettore logico dove si ha `TRUE` se il valore in quella posizione è maggiore di 7 e `FALSE` in ogni altro caso. In questo caso si hanno valori veri (quindi osservazioni con un valore maggiore di 7) nelle seguenti posizioni del vettore:

```
[1] 1 2 3 4 5 6
```

Usando il vettore logico dentro le parentesi quadre, si otterranno esattamente i valori che occupano quelle posizioni:

```r
weight[weight > 7] # solo i valori > di 7
```

```
[1] 14.902078 15.576018 17.067091  8.988758 17.594012 15.200000
```

```r
weight[weight >= 4.5 & weight < 8] # valori compresi tra 4.5 e 8
```

```
numeric(0)
```

1.11 Matrici

Le matrici sono una struttura dati **bidimensionale** (caratterizzate da 2 dimensioni, la funzione `dim()` permette di sapere da quante righe e da quante colonne è composta) dove il numero di righe rappresenta la dimensione 1 e il numero di colonne la dimensione 2.

```r
my_mat = matrix(data = 1:10, nrow = 2, ncol = 5)
my_mat
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

Per creare una matrice in R si può usare la funzione `matrix()`, come segue

```
`matrix(data, nrow, ncol, byrow = FALSE)`
```

dove `data` sono i valori che si vogliono mettere all'interno della matrice, `nrow` e `ncol` specificano il numero di righe e di colonne che si vogliono mettere nella matrice e `byrow = FALSE` (default) indica che i dati vengono organizzati per colonna (ovvero la matrice viene riempita seguendo le colonne).

```r
# crea una matrice con 3 righe e 4 colonne
# (i valori sono sistemati per riga)
A = matrix(1:12, nrow=3, ncol = 4, byrow = TRUE)
# assegna dei nomi alle righe
rownames(A) = paste("a",1:nrow(A), sep ="_")
# assegna dei nomi alle colonne
colnames(A) = paste("b",1:ncol(A), sep ="_")
A
```

```
    b_1 b_2 b_3 b_4
a_1   1   2   3   4
a_2   5   6   7   8
a_3   9  10  11  12
```

Nel caso precedente, la lunghezza del vettore organizzato nella matrice rientrava perfettamente nel numero di celle disponibili all'interno della matrice stessa. Se questo non succede, si hanno due casi:

Il numero di celle è un multiplo della lunghezza del vettore

```r
data = 1:12
matrix(data, nrow = 6, ncol =4)
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    7    1    7
[2,]    2    8    2    8
```

```
[3,]    3    9    3    9
[4,]    4   10    4   10
[5,]    5   11    5   11
[6,]    6   12    6   12
```

Non succede niente, non appaiono warning e la matrice viene creata ripetendo due volte il vettore originale.

Il numero di celle NON è un multiplo della lunghezza del vettore

```
data = 1:12
matrix(data, nrow = 3, ncol =2)
```

```
Warning in matrix(data, nrow = 3, ncol = 2): data length differs from size of
matrix: [12 != 3 x 2]
```

```
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

Si ottiene un warning, ma R crea comunque la matrice usando solo i primi valori del vettore (in questo caso i primi 6).

1.11.1 Caratteristiche

Le matrici di fatto sono la concatenazione bidimensionale di una serie di vettori. Di conseguenza:

- Possono contenere una sola tipologia di dati
- Essendo bidimensionali, sono necessari due indici di posizione (righe e colonne) per identificare un elemento
- Possono essere viste come un insieme di singoli vettori

1.11.2 Indicizzare le matrici

La logica per l'indicizzazione è la stessa che si usa per i vettori, solo che all'interno delle parentesi quadre va inserito anche l'indice per le colonne:

`my_matrix[righe, colonne]`

Il primo indice fa sempre riferimento alle righe, il secondo alle colonne:

Table 1.7: Struttura di una matrice

	[,1]	[,2]	[,3]
[1,]	[1, 1]	[1, 2]	[1, 3]
[2,]	[2, 1]	[2, 2]	[2, 3]
[3,]	[3, 1]	[3, 2]	[3, 3]

Ad esempio, data la matrice `A`:

```
A
```

```
    b_1 b_2 b_3 b_4
a_1   1   2   3   4
a_2   5   6   7   8
a_3   9  10  11  12
```

`A[1, ]` → 1, 2, 3, 4

`A[, 2]` → 2, 6, 10

`A[2, 3]` → 7

1.12 Dataframe

Il dataframe è la struttura più "complessa", utile e potente di R e come le matrici, è composto da due dimensioni (righe × colonne).

A differenza delle matrici, può contenere dati di tipo diverso:

- Ogni elemento (colonna) è un vettore con un nome associato
- Ogni colonna deve avere lo stesso numero di elementi di conseguenza ogni riga ha lo stesso numero di elementi (struttura rettangolare).
- **Tutte le colonne (quindi tutti i vettori contenuti nel dataframe) devono avere la stessa lunghezza**

  data.frame(. . .)

```
sesso = c(rep(c("M","F"), length=length(weight)))
data = data.frame(id = paste0("id", 1:length(weight)),
                  weight, sesso)
data
```

```
   id    weight sesso
1 id1 14.902078     M
2 id2 15.576018     F
3 id3 17.067091     M
4 id4  8.988758     F
5 id5 17.594012     M
6 id6 15.200000     F
```

Per convenzione, i dati sperimentali vengono organizzati in modo che le unità statistiche siano sulle righe e le variabili sulle colonne. In questo modo, sapendo il numero di righe si può ricavare il numero di unità statistiche, sapendo il numero di colonne si può ricavare il numero di variabili misurate su quelle unità statistiche.

1.12.1 Indicizzare i dataframe

I dataframe si possono indicizzare come le matrici utilizzando le parentesi quadre e i rispettivi indici di riga e colonna:

```
data[r, c]
```

Si può anche usare un'indicizzazione basata sull'estrazione delle singole colonne con l'operatore $:

```
data$nome_variabile
```

```
data
```

```
   id    weight sesso
1 id1 14.902078     M
2 id2 15.576018     F
3 id3 17.067091     M
4 id4  8.988758     F
5 id5 17.594012     M
6 id6 15.200000     F
```

```
data[2, 1]
```

```
[1] "id2"
```

```
data$weight
```

```
[1] 14.902078 15.576018 17.067091  8.988758 17.5940
```

```
data$weight[2]
```

```
[1] 15.57602
```

Dai vettori si estraggono tutte le osservazioni che soddisfano un determinato requisito come segue:

```r
weight[weight >= 7]
```

```
[1] 14.902078 15.576018 17.067091  8.988758 17.594012 15.200000
```

Nel caso del dataframe si segue la stessa logica:

Estrae tutte le righe che soddisfano la condizione e tutte le colonne.

```r
data[data$weight >=7, ]
```

```
   id    weight sesso
1 id1 14.902078     M
2 id2 15.576018     F
3 id3 17.067091     M
4 id4  8.988758     F
5 id5 17.594012     M
6 id6 15.200000     F
```

Estrae i valori del vettore `data$id` che soddisfano la condizione sulla variabile weight.

```r
data$id[data$weight >=7 & data$weight < 9]
```

```
[1] "id4"
```

1.12.2 Attributi e caratteristiche

Le funzioni **`names()`** , **`dim()`**, **`nrow()`**, **`ncol()`** sono alcune delle funzioni per ottenere informazioni sulle caratteristiche del dataframe.

La funzione più utile è **`str()`** poiché ci restituisce una veloce overview della struttura del dataframe: dimensioni, tipi di variabili.

```r
names(data) # estrae i nomi del dataset
```

```
[1] "id"     "weight" "sesso"
```

```r
dim(data) # numero di righe e numero di colonne del dataset
```

```
[1] 6 3
```

```r
str(data) # restituisce la struttura
```

```
'data.frame':   6 obs. of  3 variables:
 $ id    : chr  "id1" "id2" "id3" "id4" ...
 $ weight: num  14.9 15.58 17.07 8.99 17.59 ...
 $ sesso : chr  "M" "F" "M" "F" ...
```

```r
nrow(data) # numero di righe (equivale al numero di unità statistiche)
```

```
[1] 6
```

```r
ncol(data) # numero di colonne (equivale al numero di variabili)
```

```
[1] 3
```

> NB
>
> La funzione **`str()`** funziona su tutti gli oggetti di R (vettori, matrici, dataframe, liste). Restituisce sempre le informazioni circa la dimensionalità e il numero (e tipo) di elementi contenuti nell'oggetto.

1.13 Liste

```
list(name1 = obj1, name2 = obj2, name3 = obj3, . . .)
```

Permette di creare liste di oggetti diversi (matrici, vettori, variabili) di diverso tipo (**`numeric`**, **`character`**, **`integer`**). Ogni oggetto nella lista può avere un nome diverso che può essere utilizzato per l'indicizzazione (vedi Section 1.13.1).

Molti dei risultati ottenuti con le funzioni per le analisi statistiche sono organizzati in liste, proprio perché permettono di contenere oggetti di natura diversa e con dimensionalità diversa.

Listing 1.13 Crea una lista con oggetti diversi (vettori e matrici). Ad ogni oggetto viene assegnato un nome

```r
A <- c(1,8,5,0)
M <- matrix(A, 2, 2)
Nomi3 <- c("giorgio", "ugo", "anna")
LV <- c(TRUE, FALSE, TRUE, TRUE)
OgLista <- list(a = A, Mat = M, nomi = Nomi3, X = LV)
str(OgLista)
```

```
List of 4
 $ a   : num [1:4] 1 8 5 0
 $ Mat : num [1:2, 1:2] 1 8 5 0
 $ nomi: chr [1:3] "giorgio" "ugo" "anna"
 $ X   : logi [1:4] TRUE FALSE TRUE TRUE
```

1.13.1 Indicizzare le liste

`lista[[i]]`

`[[i]]`: indice i dell'i-esimo oggetto all'interno della lista a cui si vuole accedere (vedi Listing 1.14)

Listing 1.14 Accede al secondo oggetto della lista con il suo indice

```
OgLista[[2]]
```

```
     [,1] [,2]
[1,]    1    5
[2,]    8    0
```

`lista$nome_oggetto`

Gli oggetti dentro le liste si possono indicizzare anche attraverso i loro nomi usando l'operatore `$` (vedi Listing 1.15):

Listing 1.15 Accede all'oggetto Mat nella lista usando il suo nome

```
OgLista$Mat
```

```
     [,1] [,2]
[1,]    1    5
[2,]    8    0
```

1.14 Importare dati esterni

1.14.1 Importa base

```
read.csv2("file_path", header = TRUE, sep = ";")
```

Il default della funzione prevede che la prima riga contenga i nomi delle variabili (`header = TRUE`) e che il separatore di colonne (ovvero il valore che fa riconoscere ad R le diverse colonne del dataset) sia ";" (`sep = ";"`).

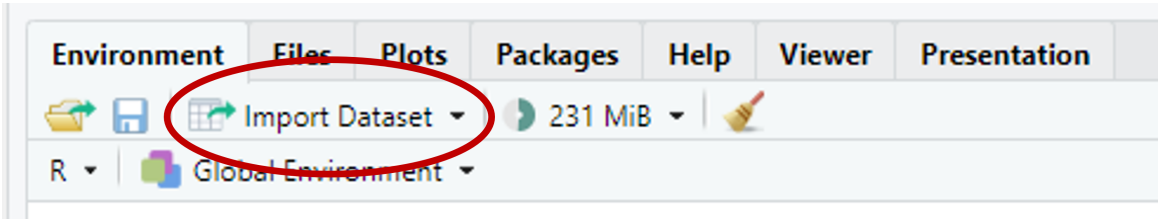
Se si possiede un dataset chiamato "dati" nella cartella "data", questo codice lo importerà all'interno dell'ambiente.

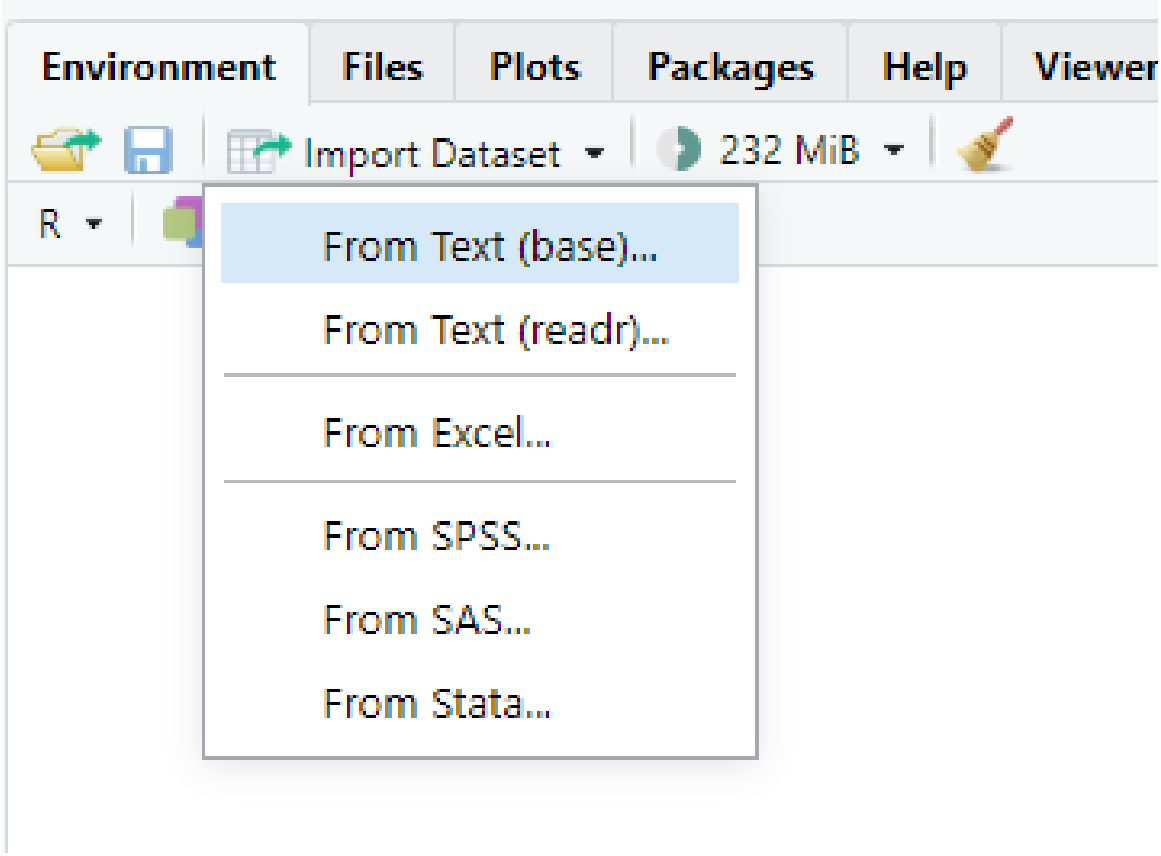
```r
# Lettura file dati esterni
data <- read.csv2("data/dati.csv")
```

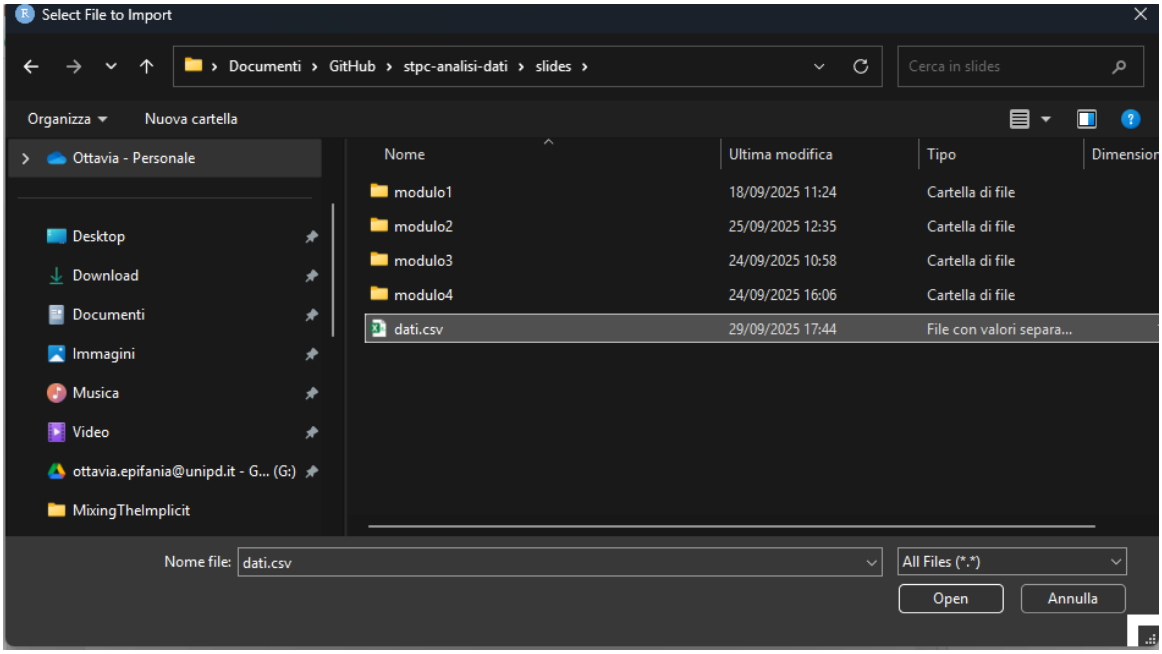
> **Caution**
>
> Nei dataset messi a disposizione per questo corso, si deve sempre usare la funzione `read.csv2()`.
> Perché il codice di sopra funzioni, si assume che la cartella `data` sia all'interno del progetto R creato per il corso (vedi Section 1.2.2).

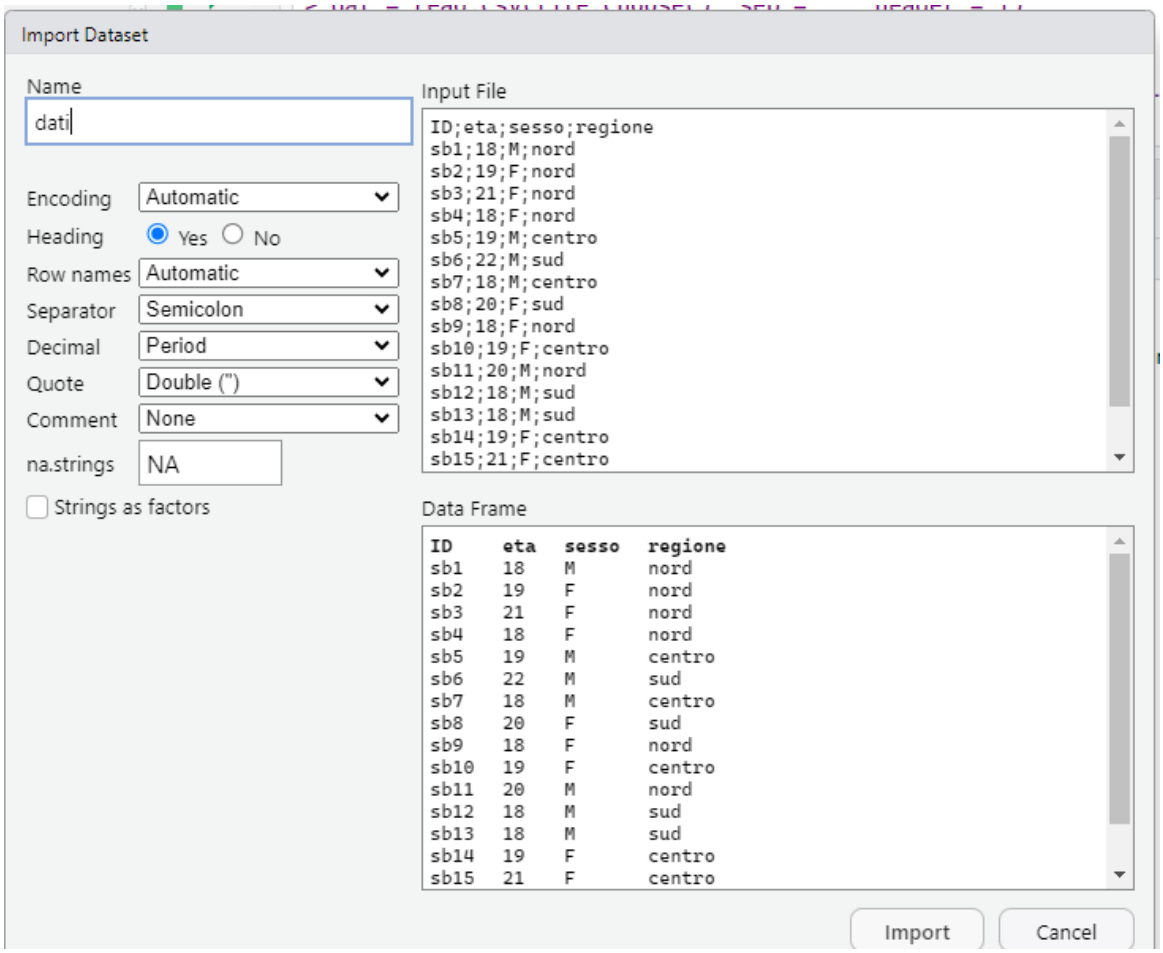
1.14.2 Importa alternativa 1

> **Warning**
>
> Questa alternativa è presentata solo per completezza di informazione, ma è fortemente sconsigliata visto che non può essere automatizzata dentro i documenti quarto.




Environment
Files
Plots
Packages
Help
Viewer
Import Dataset
232 MiB
R
From Text (base)...
From Text (readr)...
From Excel...
From SPSS...
From SAS...
From Stata...


Select File to Import
Documenti › GitHub › stpc-analisi-dati › slides ›
Cerca in slides
Organizza
Nuova cartella
Ottavia - Personale
Desktop
Download
Documenti
Immagini
Musica
Video
ottavia.epifania@unipd.it - G... (G:)
MixingTheImplicit
Nome
Ultima modifica
Tipo
Dimensior
modulo1
18/09/2025 11:24
Cartella di file
modulo2
25/09/2025 12:35
Cartella di file
modulo3
24/09/2025 10:58
Cartella di file
modulo4
24/09/2025 16:06
Cartella di file
dati.csv
29/09/2025 17:44
File con valori separa...
Nome file:
dati.csv
All Files (*.*)
Open
Annulla



1.14.3 Importa alternativa 2

> **Warning**
>
> Questa funzione è la funzione base su cui poggia `read.csv2()`. Cambiano i default, che in questo caso sono ottimizzati per i file csv creati in altre regioni del mondo, dove "," è il separatore di colonna di default e "." è il separatore di decimali di default.

```
dati = read.csv("slides/dati.csv",
                sep =",", dec =".", header = TRUE)
head(dati)
```

```
  ID.eta.sesso.regione
1         sb1;18;M;nord
2         sb2;19;F;nord
3         sb3;21;F;nord
4         sb4;18;F;nord
```

```
5      sb5;19;M;centro
6         sb6;22;M;sud
```

1.15 Sintesi delle strutture tabellari

1.15.1 Riassunto descrittivo: `summary()`

```
summary(data)
```

Riassume il contenuto di strutture di dati, riportando diverse statistiche descrittive (Table 1.8). Tutte le statistiche descrittive possono essere ottenute con funzioni equivalenti.

Table 1.8: Statistiche restituite dalla funzione `summary()` per una variabile numerica

Statistica	Nome	Cosa indica	Funzione equivalente
`Min.`	Minimo	Il valore più piccolo osservato	`min(x)`
`1st Qu.`	Primo quartile (Q_1)	Il valore preceduto dal 25% delle osservazioni e superato dal restante 75%	`quantile(x, 0.25)`
`Median`	Mediana (Q_2)	Il valore centrale della distribuzione ordinata: il 50% delle osservazioni sta sotto, il 50% sopra	`median(x)`
`Mean`	Media aritmetica	Il baricentro della distribuzione: la somma dei valori divisa per il numero di osservazioni	`mean(x)`
`3rd Qu.`	Terzo quartile (Q_3)	Il valore preceduto dal 75% delle osservazioni e superato dal restante 25%	`quantile(x, 0.75)`
`Max.`	Massimo	Il valore più grande osservato	`max(x)`
`NA's`	Valori mancanti	Il numero di osservazioni mancanti (compare solo se ce ne sono)	`sum(is.na(x))`

```
summary(rock)
```

```
      area            peri            shape             perm        
 Min.   : 1016   Min.   : 308.6   Min.   :0.09033   Min.   :   6.30  
 1st Qu.: 5305   1st Qu.:1414.9   1st Qu.:0.16226   1st Qu.:  76.45  
 Median : 7487   Median :2536.2   Median :0.19886   Median : 130.50  
 Mean   : 7188   Mean   :2682.2   Mean   :0.21811   Mean   : 415.45  
 3rd Qu.: 8870   3rd Qu.:3989.5   3rd Qu.:0.26267   3rd Qu.: 777.50  
 Max.   :12212   Max.   :4864.2   Max.   :0.46413   Max.   :1300.00  
```

1.15.2 Struttura: `str()`

```
str(data)
```

Fornisce informazioni circa la struttura generale di un oggetto (qualsiasi oggetto):

```
str(rock) #dataframe
```

```
'data.frame':   48 obs. of  4 variables:
 $ area : int  4990 7002 7558 7352 7943 7979 9333 8209 8393 6425 ...
 $ peri : num  2792 3893 3931 3869 3949 ...
 $ shape: num  0.0903 0.1486 0.1833 0.1171 0.1224 ...
 $ perm : num  6.3 6.3 6.3 6.3 17.1 17.1 17.1 17.1 119 119 ...
```

```
str(weight)
```

```
 num [1:6] 14.9 15.58 17.07 8.99 17.59 ...
```

1.15.3 L'inizio `head()` e la fine `tail()` di un oggetto

`head()`

Le prime 6 osservazioni del dataset

```
head(rock)
```

```
  area    peri     shape perm
1 4990 2791.90 0.0903296  6.3
2 7002 3892.60 0.1486220  6.3
3 7558 3930.66 0.1833120  6.3
4 7352 3869.32 0.1170630  6.3
5 7943 3948.54 0.1224170 17.1
6 7979 4010.15 0.1670450 17.1
```

`tail()`

le ultime 6 osservazioni del dataset

```
tail(rock)
```

```
   area     peri    shape perm
43 5605 1145.690 0.464125 1300
44 8793 2280.490 0.420477 1300
45 3475 1174.110 0.200744  580
46 1651  597.808 0.262651  580
47 5514 1455.880 0.182453  580
48 9718 1485.580 0.200447  580
```

1.15.4 `apply()`

Ci sono diverse funzioni in R che permettono di applicare la stessa funzione più volte su uno stesso oggetto, come ad esempio la funzione `apply()`:

```
apply(data, MARGIN, FUN)
```

`data`: i dati su cui si vuole lavorare. Si può prendere anche una sottoselezione di dati utilizzando l'indicizzazione del dataframe (sia per le righe sia per le colonne, vedi Section 1.12.1).

`MARGIN = 1`: applica la funzione sulle righe (attraverso le colonne). Questo vuol dire che si ottiene un vettore dei risultati di lunghezza uguale al numero di righe del dataframe.

`MARGIN = 2`: applica la funzione sulle colonne (attraverso le righe). Questo vuol dire che si ottiene un vettore dei risultati di lunghezza uguale al numero di colonne del dataframe.

`FUN`: la funzione che si vuole applicare

```
apply(rock, 1, sum)   # Somma per riga
```

```
 [1]  7788.290 10901.049 11495.143 11227.737 11908.762 12006.417 13696.040
 [8] 12571.014 12194.244  9642.812 13963.201 12729.388 14770.169 12468.672
[15] 13498.943 12585.623 15629.464 15589.483 16801.020 14405.155 11408.854
[22] 16371.431 17051.890 11893.602  9077.671  6312.529 10072.432  7571.553
[29]  9250.435  7025.001  9393.452 10245.336  8356.958  7258.715  9692.914
[36] 10305.336  4945.877  2044.761  4813.624  7012.214  7289.872  2624.872
[43]  8051.154 12373.910  5229.311  2829.071  7550.062 11783.780
```

```
apply(rock, 2, mean)  # Media per colonna
```

```
        area         peri        shape         perm
7187.7291667 2682.2119375    0.2181104  415.4500000
```

2 Modulo 2 – Scale di misura e statistica descrittiva univariata

2.1 Tipi di variabili

QUALITATIVE / CATEGORIALI

- Sconnesse: $\{a, b, c\}$
- Ordinate: $\{a, b, c\}$ con $a < b < c$ (oppure $a > b > c$)

QUANTITATIVE / METRICHE

- Discrete: $\mathbb{Z}$
- Continue: $\mathbb{R}$

2.2 Scale di misura (Stevens, 1946)

Le variabili si differenziano in base alla quantità di informazione che può essere ricavata.

- Variabili qualitative
  - **Nominale**: Distingue un insieme di dati in diverse categorie
  - **Ordinale**: Distingue un insieme di dati in diverse categorie che sono ordinabili a seconda della quantità di caratteristica posseduta
- Variabili quantitative
  - **Intervalli equivalenti** Distingue un insieme di dati in diverse categorie ordinabili. La distanza tra le categorie è nota perché c'è un'unità di misura
  - **A rapporti equivalenti**: Come la scala ad intervalli...ma lo 0 indica assenza di caratteristica (non è arbitrario ma è assoluto)

2.2.1 Categoriali - Nominali

Comune di residenza, sesso biologico, patologia clinica ecc.

2.2.1.1 `as.factor()`

`as.factor()`: Converte una variabile in una variabile di tipo **categoriale/nominale** con tanti livelli quanti sono i valori unici nel vettore

Due livelli

```
Genere <- c("m","f","f","m","f","m")
Genere
```

```
[1] "m" "f" "f" "m" "f" "m"
```

```
as.factor(Genere)
```

```
[1] m f f m f m
Levels: f m
```

`as.character()` produce una variabile categoriale, ma senza i livelli che caratterizzano i `factor`

Più livelli

```
diagnosi <- c("schizo", "dep", "narc", "schizo", "dep", "narc")
diagnosi
```

```
[1] "schizo" "dep"    "narc"   "schizo" "dep"    "narc"
```

```
as.factor(diagnosi)
```

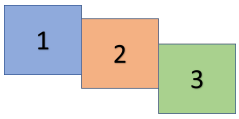
```
[1] schizo dep    narc   schizo dep    narc
Levels: dep narc schizo
```

2.2.2 Scala ordinale

Distingue un insieme di dati in diverse categorie che sono ordinabili a seconda della quantità di caratteristica posseduta. Vale sia l'equivalenza ($=$ o $\neq$) sia l'ordinamento ($<$ o $>$).

Esempi: Titolo di studio, gradimento di un prodotto, valutazione di una malattia (alto, medio, basso).

Le etichette numeriche valgono per il loro ordine (non ha senso compiere operazioni tra di loro).



Attenzione!

Si conosce l'ordine tra le categorie, non si conosce la distanza.
Le etichette numeriche valgono per il loro ordine (non ha senso compiere operazioni tra di loro)

```
factor(A, levels = c("a1", "a2", ..., "ak"))
```

Converte la variabile in una variabile con tanti livelli quanti sono i valori unici e viene forzato un ordine su questi livelli:

Listing 2.1 Utilizzo dell'argomento levels per ordinare i livelli di una variabile categoriale ordinata

```
Voto <- c("alto","medio","medio","basso","alto")
Voto
```

```
[1] "alto"  "medio" "medio" "basso" "alto"
```

Listing 2.2 Utilizzo dell'argomento levels per ordinare i livelli di una variabile categoriale ordinata

```
factor(Voto,levels=c("basso","medio","alto"))
```

```
[1] alto  medio medio basso alto
Levels: basso medio alto
```

Warning

Esiste anche l'argomento `ordered = TRUE`, che crea un fattore di classe `ordered`: alcune funzioni lo trattano in modo diverso da un fattore normale (ad esempio, nei modelli lineari vengono usati contrasti polinomiali). In questo corso non lo useremo: per le variabili ordinali è sufficiente ordinare i livelli attraverso l'argomento `levels`, come mostrato in Listing 2.2.

2.2.3 Scala a intervalli equivalenti

Lo 0 è arbitrario, ma c'è un'unità di misura per cui le categorie si trovano alla stessa distanza. Vale equivalenza ($=$ o $\neq$), ordinamento ($<$ o $>$) e differenza ($+$ o $-$).

Esiste una relazione d'ordine tra le quantità e la distanza tra le quantità è nota e definita da una unità di misura costante e **nota**.

Esempi: Temperatura in Celsius, Q.I., scale di atteggiamento.

Se ci sono 0°C, non vuol dire che non c'è calore!

Prendiamo due temperature, T_a e T_b. Queste temperature in Celsius C sono $T_a(C) = 5$ e $T_b(C) = 10$.

Quello che si può dire:

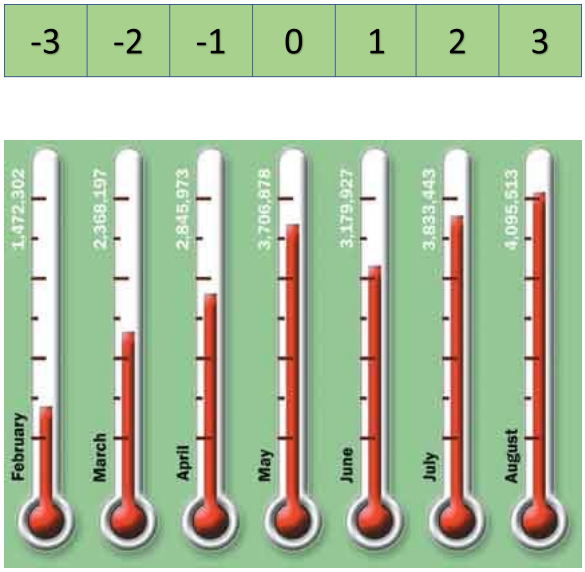
- $T_a(C) < T_b(C)$ (oppure $T_b(C) > T_a(C)$)
- $T_b(C) - T_a(C) = 5$
- Non possiamo dire che $T_b(C) = 2T_a(C)$.

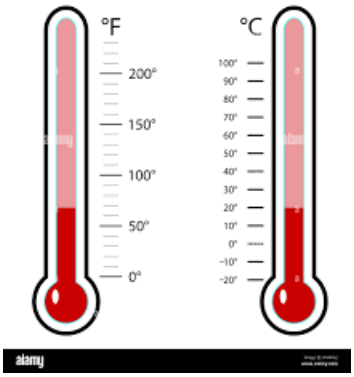
Infatti, volendo trasformare le due temperature in Fahrenheit tramite la relazione:

$$F = \frac{9}{5}C + 32$$

si vedrebbe che $T_a(F) = \frac{9}{5}5 + 32 = 41$ e $T_b(F) = \frac{9}{5}10 + 32 = 50$.

Rimane la relazione d'ordine, per cui anche $T_a(F) < T_b(F)$, ma cambia sia la differenza tra le due quantità $T_b(F) - T_a(F) = 9$ sia la relazione tra le due, per cui si vede chiaramente che $T_b(F) \neq 2T_a(F)$.





2.2.4 Scala a rapporti equivalenti

Lo 0 è assoluto: indica assenza del fenomeno. Vale equivalenza ($=$ o $\neq$), ordinamento ($<$ o $>$), differenza ($+$ o $-$) e rapporto ($\times$ o $\div$).

Esempi: Peso, altezza, valori diagnostici.

La presenza dello 0 assoluto permette di andare oltre la mera relazione d'ordine e distanza tra le quantità.

Si prendano ad esempio due pesi, P_a e P_b. Questi pesi, misurati in chilogrammi K, sono $P_a(K) = 10$ e $P_b(K) = 30$. Quello che si può dire:

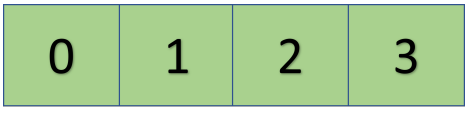
- $P_a(K) < P_b(K)$ $(P_b(K) > P_a(K))$
- $P_b(K) - P_a(K) = 20$
- Ma soprattutto $P_a(K) = \frac{1}{3}P_b(K)$ oppure $P_b(K) = 3P_a(K)$

La relazione tra P_a e P_b rimane **invariante** attraverso le diverse trasformazioni di scala.

Ad esempio, trasformando $P_a(K)$ e $P_b(K)$ in libbre (L) mediante la relazione

$$L = K \times 2.20462$$

per cui $P_a(L) = 10 \times 2.20462 = 22.0462$ e $P_b(L) = 30 \times 2.20462 = 66.1386$. La relazione d'ordine è mantenuta e, al netto dei decimali, rimane anche invariante il rapporto tra i due pesi.





2.2.4.1 `numeric`, `integer()`

Le **variabili metriche** in R sono variabili di tipo `numeric` o `integer`, a seconda che siano continue o discrete

> Attenzione!
>
> In R non esiste una distinzione tra variabili a intervalli equivalenti e variabili a rapporti equivalenti – Sta a voi capire che tipo di variabile sia

```
x = seq(-3, 3, by = .5); x                                    ①
y = rnorm(10, mean = 3, sd=2); y                              ②
z = 1:15; z                                                   ③
```

① La funzione `seq()` genera un vettore di numeri da un minimo a un massimo, con uno step progressivo definito dall'utente
② La funzione `rnorm()` genera un vettore di lunghezza definita estraendo a caso da una distribuzione normale
③ L'operatore `:` genera una sequenza unitaria di valori da un minimo a un massimo

```
 [1] -3.0 -2.5 -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0  2.5  3.0
 [1] 1.88193977 1.23509550 0.08331974 2.05385651 4.87695221 2.04275946
 [7] 6.41210424 6.26516032 4.75113899 3.00929922
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

```
str(x)
```

```
 num [1:13] -3 -2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 ...
```

```r
str(y)
```

```
 num [1:10] 1.8819 1.2351 0.0833 2.0539 4.877 ...
```

```r
str(z)
```

```
 int [1:15] 1 2 3 4 5 6 7 8 9 10 ...
```

2.3 Frequenze

Le frequenze sono le statistiche descrittive più semplici che si possano calcolare su una variabile: indicano quante volte compare ciascuna categoria (o ciascun valore). Si possono calcolare sia sulle variabili categoriali sia su quelle quantitative, ma sono più informative per le prime. Nelle variabili quantitative, soprattutto se continue, quasi ogni valore compare una sola volta, e le frequenze diventano utili solo dopo aver raggruppato i valori in classi.

2.3.1 Assolute e relative

Categoria x_i	n_i	$n_i\% = \frac{n_i}{n} \cdot 100$
x_1	n_1	$n_1\%$
x_2	n_2	$n_2\%$
$\vdots$	$\vdots$	$\vdots$
x_k	n_k	$n_k\%$
Totale	n	100

Frequenza assoluta: La frequenza assoluta semplice di una categoria è il **numero naturale di unità statistiche** che presentano tale categoria. $n_i \geq 0$, con $n_i \in \mathbb{N}$ e $\sum_{i=1}^{k} n_i = n$

Frequenza relativa e frequenza percentuale: La frequenza relativa semplice $f_i = \frac{n_i}{n}$ è data dal rapporto tra la frequenza assoluta semplice di tale categoria e il numero totale di unità statistiche osservate ($0 \leq f_i \leq 1$). Moltiplicata per 100, diventa la **frequenza percentuale**: $n_i\% = \frac{n_i}{n} \cdot 100$, con $0 \leq n_i\% \leq 100$ e $\sum_{i=1}^{k} n_i\% = 100$

2.3.1.1 Frequenze univariate: `table()`

```
table(variabile)
```

Calcola le frequenze per ogni categoria. Funziona sia per le variabili categoriali sia per le variabili metriche (anche se ha poco senso).

```
table(Genere)
```

```
Genere
f m
3 3
```

```
table(diagnosi)
```

```
diagnosi
   dep   narc schizo
     2      2      2
```

2.3.2 Cumulate

Frequenza cumulata: numero o percentuale di unità statistiche che presentano una categoria "minore o uguale" alla categoria corrente.

Frequenze assolute cumulate

$$N_i = n_1 + n_2 + \cdots + n_i = \sum_{j=1}^{i} n_j$$

con $N_1 = n_1$ e $N_k = n$.

Frequenze percentuali cumulate

$$N_i\% = n_1\% + n_2\% + \cdots + n_i\% = \sum_{j=1}^{i} n_j\%$$

con $N_1\% = n_1\%$ e $N_k\% = 100\%$.

Le frequenze cumulate possono essere definite **ricorsivamente**:

$$N_i = N_{i-1} + n_i, \qquad N_i\% = N_{i-1}\% + n_i\%$$

Attenzione

Le frequenze cumulate si calcolano **solo quando le modalità del carattere presentano un ordinamento**.
Non hanno quindi senso per caratteri qualitativi sconnessi (nominali).

Esprimi l'accordo con l'affermazione:

> Insegnare statistica a psicologia è una pessima idea

Modalità x_i	n_i	N_i	n_i%	N_i%
Totalmente in disaccordo	10	10	24.4%	24.4%
Abbastanza in disaccordo	8	18	19.5%	43.9%
Indifferente	6	24	14.6%	58.5%
Abbastanza d'accordo	14	38	34.1%	92.7%
Totalmente d'accordo	3	41	7.3%	100.0%
Totale	41		100.0%	

Le percentuali sono arrotondate al primo decimale, per cui la loro somma può non essere esattamente 100.

2.3.2.1 Calcolare le frequenze cumulate: `cumsum()`

```
opinione = data.frame(
  categoria = c("Totalmente in disaccordo", "Abbastanza in disaccordo",
    "Indifferente",
    "Abbastanza d'accordo", "Totalmente d'accordo"),
  n = c(10, 8, 6, 14, 3)
)
opinione$N = cumsum(opinione$n)
opinione
```

```
                categoria  n  N
1 Totalmente in disaccordo 10 10
2 Abbastanza in disaccordo  8 18
3             Indifferente  6 24
4     Abbastanza d'accordo 14 38
5     Totalmente d'accordo  3 41
```

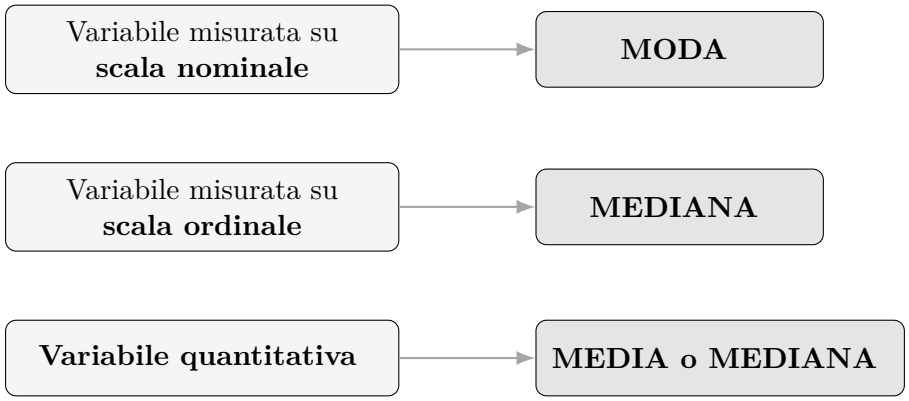
2.4 Indici di posizione

- **indici sintetici** che evidenziano le caratteristiche essenziali della distribuzione della variabile

Presentano livelli maggiori di machiavellismo l3 CEO o l3 impiegat3?

- Si confrontano categorie che rappresentano i livelli o valori tipici.
- Consentono di sintetizzare un insieme di misure tramite un unico valore "rappresentativo" → indice che riassume o descrive i dati e dipende dalla scala di misura dei dati in oggetto.
- moda
- mediana
- quartili/percentili
- media aritmetica

Ogni variabile statistica ha l'indice di posizione adeguato: non tutti gli indici si possono calcolare per ogni variabile → **indice di posizione adeguato**



2.4.1 Moda (Mo)

La moda Mo di una variabile X è la categoria a cui è associata la massima frequenza:

$$Mo(X) = x_i \text{ tale che } n_i = \max(n_1, \dots, n_k)$$
(categoria con massimo valore di n_i)

> Warning
>
> N.B: Si può calcolare per le variabili misurate a qualsiasi livello, anche se di fatto viene calcolata solo per le variabili categoriali, in quanto per altri livelli di misurazione si possono calcolare altri indici più informativi

```r
opinione
```

```
                 categoria  n  N
1 Totalmente in disaccordo 10 10
2 Abbastanza in disaccordo  8 18
3             Indifferente  6 24
4     Abbastanza d'accordo 14 38
5     Totalmente d'accordo  3 41
```

```r
max(opinione$n) # restituisce max(n_i)
```

```
[1] 14
```

```r
# va filtrato il dataframe per ottenere la categoria a cui
# è associata la frequenza massima
opinione[opinione$n == max(opinione$n), "categoria"]
```

```
[1] "Abbastanza d'accordo"
```

```r
# codice alternativo che produce lo stesso risultato
opinione$categoria[opinione$n == max(opinione$n)]
```

```
[1] "Abbastanza d'accordo"
```

2.4.2 Mediana

La mediana Me di una variabile X è la categoria che occupa la posizione centrale o mediana (Pos_{Me}) nella distribuzione ordinata dei dati, ovvero che è:

- preceduta da almeno 50% dei casi
- superata da almeno 50% dei casi

$$Pos_{Me} = \frac{n+1}{2}$$

Il valore di Pos_{Me} va cercato nella distribuzione delle frequenze cumulate e si sceglie la categoria con $N_i \geq Pos_{Me}$

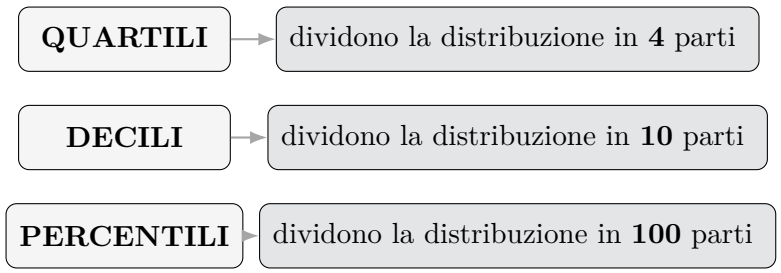
2.4.3 Quantili

Categorie/valori che dividono la distribuzione di frequenza ordinata in più parti.

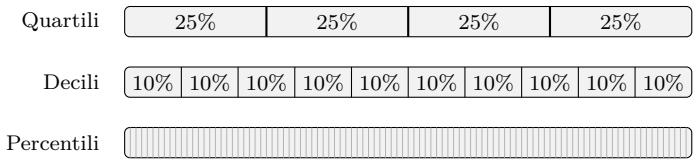
> Qual è il reddito familiare che divide il 25% dei più poveri dal restante 75% ?
>
> Qual è la soglia di reddito oltre cui sta la fascia dei più abbienti ?
>
> Quali sono le persone che hanno performato al meglio nell'ultimo semestre?

Il quantile x_p di ordine p della variabile X è quella categoria/valore che è:

- preceduta da almeno $p\%$ dei casi
- superata da almeno $(100 - p)\%$ dei casi



La distribuzione è la stessa: cambia il modo in cui viene suddivisa



Per il calcolo dei quantili di diverso ordine, il procedimento è analogo a quello della Mediana, che di fatto è un caso particolare di questi indici. In generale la Posizione di ordine p (in centesimi) si ottiene:

$$Pos_p = \frac{n+1}{100} \cdot p$$

2.4.3.1 Trovare i quantili: `quantile()`

```
quantile(x, probs = c(0, .25, .50, .75, 1))
```

Funzione per il calcolo dei quantili da una variabile X (argomento `x`) di diverso ordine (definito dall'argomento `probs`). Di default, vengono calcolati i quartili, compresi anche il minimo (`0`) e il massimo (`1`) della variabile X.

```r
set.seed(123) # garantisce la replicabilità dei risultati
x = rnorm(1000, mean = 0, sd = 1)
quantile(x)
```

```
          0%          25%          50%          75%         100%
-2.809774679 -0.628324243  0.009209639  0.664601867  3.241039935
```

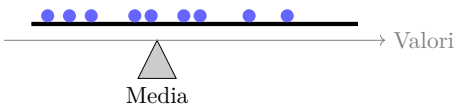
```r
# calcola i decili
quantile(x, probs = seq(0,1, by = .10))
```

```
          0%          10%          20%          30%          40%          50%
-2.809774679 -1.267328289 -0.788690220 -0.486452454 -0.214854912  0.009209639
         60%          70%          80%          90%         100%
 0.242040647  0.513088142  0.841412710  1.254751947  3.241039935
```

2.4.4 Media

La media è una sorta di baricentro della distribuzione dei dati. Data una variabile X composta da n valori:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$



I dati: $X = (0.8, 1.2, 1.6, 2.4, 2.7, 3.3, 3.6, 4.5, 5.2)$, la media $\bar{x} = 2.811$



I dati: $X = (1.18, 1.02, 1.10, 1.16, 1.01, 1.10, 1.05, 1.12, 1.45, 1.52, 1.54, 1.55, 10.06, 14.46)$, la media $\bar{x} = 2.809$

2.4.4.1 `mean()`

`mean(x, na.rm = FALSE)`

Funzione per calcolare la media di un vettore di valori `x`. L'argomento `na.rm = TRUE` serve per ottenere il calcolo della media nonostante la presenza di valori NA

`rowMeans()`: Calcola la media per ogni riga (quindi attraverso le colonne)

`colMeans()`: Calcola la media per ogni colonna (quindi attraverso le righe)

```
mean(x)
```

```
[1] 0.01612787
```

2.5 Indici di variabilità

Il concetto di variabilità si riferisce a quanto i punteggi di una distribuzione sono sparsi, ovvero quanto siano simili o dissimili tra loro.

- Il ricorso a un indice di tendenza centrale (come la media) comporta una forte semplificazione
- È fondamentale capire quanto i dati siano dispersi intorno all'indice di tendenza centrale
- La variabilità è una caratteristica fondamentale delle distribuzioni: *"Variability is the essence of statistics"* (Cobb, 1992)

Gli indici di variabilità possono assumere solo valori positivi (non ha senso parlare di dispersione negativa) o nulli (quando i dati hanno tutti lo stesso valore).

2.5.1 Campo di variazione (Gamma) o range

Il campo di variazione è la differenza tra il valore massimo e il valore minimo osservato:

$$\text{gamma} = x_{\max} - x_{\min}$$

```
team_A <- c(38, 42, 44, 36, 39, 42, 38, 41)
max(team_A) - min(team_A)
```

```
[1] 8
```

Limitazione

Dipende solo da due valori (massimo e minimo) e non considera la distribuzione dei dati intermedi.

2.5.2 La varianza

La varianza σ^2 è la media degli scarti al quadrato tra i dati e la media:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

La varianza assume valore minimo 0 quando tutti i dati sono uguali tra loro e aumenta all'aumentare della dispersione.

Important

La funzione `var()` di R calcola la **varianza campionaria**, che divide per $n-1$ anziché n. Per dati descrittivi (popolazione osservata) si usa n; per stimare la varianza della popolazione da un campione si usa $n-1$.

2.5.3 La deviazione standard

La deviazione standard è la radice della varianza:

$$\sigma = \sqrt{\sigma^2}$$

Riporta l'indice di variabilità sulla **scala della variabile**.

Esempio: Se in un campione di dipendenti la variabile `WorkHoursPerWeek` ha media 40 ore e deviazione standard 2.5, i dipendenti differiscono mediamente di 2.5 ore dal carico di lavoro medio.

La deviazione standard, a differenza della varianza, è espressa nella stessa unità di misura della variabile.

2.5.4 Il coefficiente di variazione

Il coefficiente di variazione è il rapporto tra la deviazione standard e il valore assoluto della media:

$$CV = \frac{\sigma}{|\bar{x}|}$$

Table 2.1: Dataset di Anscombe

Dataset	Media.X	Media.Y	Varianza.X	Varianza.Y	Correlazione
Dataset 1	9	7.5	11	4.13	0.82
Dataset 2	9	7.5	11	4.13	0.82
Dataset 3	9	7.5	11	4.12	0.82
Dataset 4	9	7.5	11	4.12	0.82

Il CV è un indice di **variabilità relativa** fondamentale per eseguire **confronti** tra fenomeni misurati su **scale diverse**.

2.5.5 Differenza interquartilica (IQR)

La differenza interquartilica è la differenza tra il terzo e il primo quartile:

$$IQR = Q_3 - Q_1$$

L'IQR misura la dispersione del 50% centrale dei dati ed è un indice **robusto** perché i valori estremi incidono poco o per nulla.

Outlier: valori oltre $[Q_1 - 1.5 \times IQR \; ; \; Q_3 + 1.5 \times IQR]$.

Il **boxplot** (vedi Section 2.7.3) visualizza Q_1, mediana, Q_3 e gli outlier in un unico grafico.

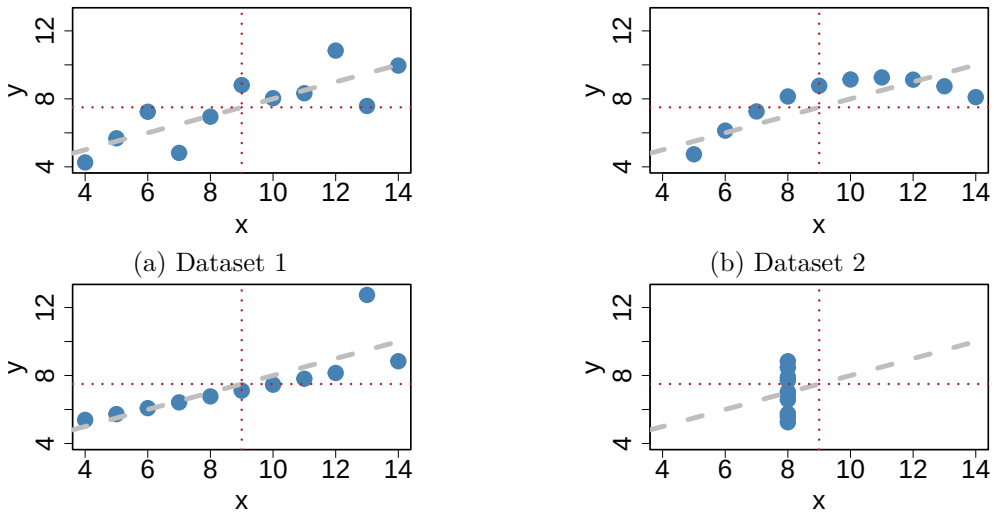
La funzione `IQR(x)` permette di calcolare la differenza interquartilica in R per qualsiasi variabile quantitativa.

2.6 Perché i grafici

> Graphs remind us that the process of statistical inference is not mechanical. This process often involves subjective decisions (e.g., the evaluation, exploration, and/or deletion of outliers) which are an integral part of the analysis. Thus, graphs are among the most appropriate tools to enhance transparency and confer plausibility to such decisions.
>
> Pastore et al. (2017)

Data set di Anscombe (Table 2.1): Quattro dataset, di seguito le statistiche descrittive relative alla variabile x e alla variabile y e la loro correlazione:

Guardando la Table 2.1, sembra che i quattro dataset siano identici, in termini di media, varianza e correlazione tra le variabili x e y. Tuttavia, se si guardano i grafici a dispersione (vedi Section 3.4) tra le variabili x e y dei quattro dataset (Figure 2.1), appaiono delle differenze abbastanza palesi tra i diversi dataset.



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

(d) Dataset 4

Figure 2.1: I grafici raccontano un'altra storia

2.7 Grafici

R fornisce uno strumento molto utile per la rappresentazione grafica dei dati e dei risultati già con il suo pacchetto base `graphics`

> NB
>
> Non avete bisogno di installarlo, è caricato automaticamente nel momento in cui scaricate R e le sue funzioni sono sempre disponibili

`ggplot2` è un pacchetto più avanzato per la creazione dei grafici. Non entreremo nel dettaglio del suo funzionamento perché è un po' più complicato

Alcune funzioni incluse nel pacchetto `graphics`:

```
plot(x)                ①
plot(x, y)             ②
hist(x)                ③
boxplot(x)             ④
barplot(x)             ⑤
```

Table 2.2: Tabelle di frequenza per le categorie A, B, C.

Categoria	n_i
A	21
B	24
C	22

Categoria	n_i
A	12
B	35
C	18

① Grafico di una variabile quantitativa
② Grafico a dispersione di due variabili quantitative (vedi Section 3.4)
③ Istogramma di una variabile quantitativa (Section 2.7.2)
④ Boxplot della distribuzione quartile di una (o più) variabili (Section 2.7.3)
⑤ Grafico a barre per le frequenze delle variabili qualitative (Section 2.7.1)

2.7.1 Grafico a barre

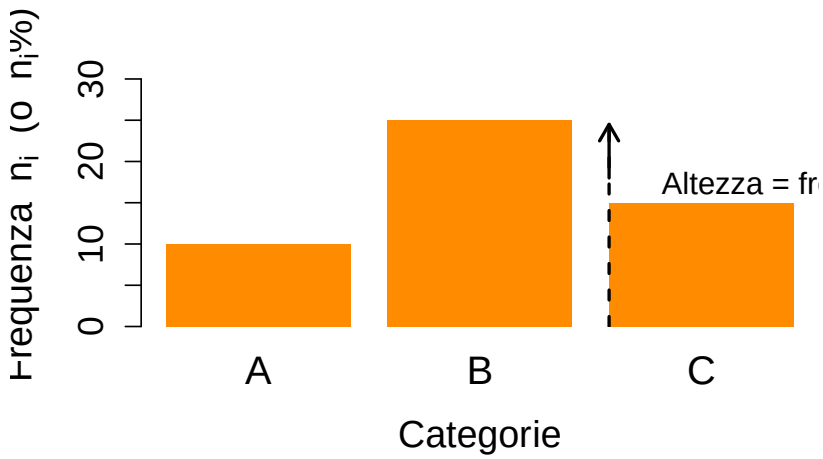
Usato per variabili qualitative nominali o ordinali:

- Asse x: categoria x_i
- Asse y: Frequenza n_i (vale anche per le frequenze percentuali)

L'altezza della barretta è proporzionale alla sua frequenza

```
barplot(x)
```

Dove `x` è il vettore delle frequenze (quello che si ottiene applicando la funzione `table()` a una variabile categoriale)



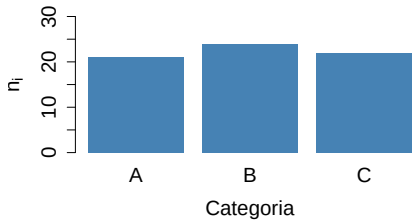


Figure 2.2: Barplot delle tabelle

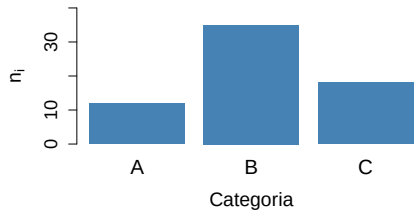


Figure 2.3: Barplot delle tabelle

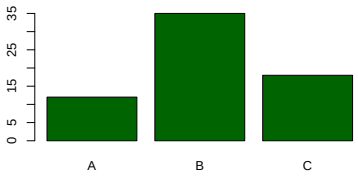
```
dati = factor(rep(c("A", "B", "C"), c(12, 35, 18))); dati
```

```
 [1] A A A A A A A A A A A A B B B B B B B B B B B B B B B B B B B B B B B B B B
[39] B B B B B B B B B C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
Levels: A B C
```

```
table(dati)
```

```
dati
 A  B  C
12 35 18
```

```
barplot(table(dati), col = "darkgreen")
```



2.7.2 Istogrammi

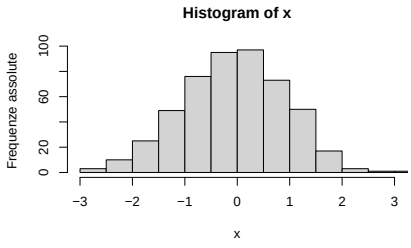
`hist(x, breaks)` produce un istogramma a partire da un vettore `x` di osservazioni quantitative.

Per costruire l'istogramma, l'intervallo dei valori di `x` viene diviso in classi contigue, disposte sull'asse delle x. Per ogni classe si conta quante osservazioni vi ricadono: questa frequenza è l'altezza della barra sull'asse delle y.

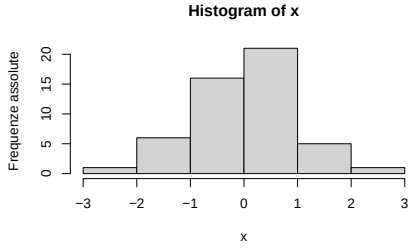
L'argomento `breaks` stabilisce come vengono costruite le classi. Può essere il numero di classi desiderato, una regola per calcolarlo (`"Sturges"`, il default, `"FD"` o `"Scott"`), oppure un vettore con i punti di taglio esatti. Il numero di classi determina la loro ampiezza:

- **Classi strette (molti `breaks`)**: ogni classe raccoglie poche osservazioni, quindi le barre sono basse e alcune classi possono restare vuote. L'istogramma risulta frastagliato e difficile da leggere.
- **Classi larghe (pochi `breaks`)**: ogni classe raccoglie molte osservazioni, quindi le barre sono più alte ma sono poche. L'istogramma diventa troppo grossolano e nasconde la forma della distribuzione.

> Note
>
> Quando `breaks` è un numero, R lo usa come *suggerimento* e lo arrotonda a intervalli "puliti". Per questo con `breaks = 145` non si ottengono esattamente 145 barre.



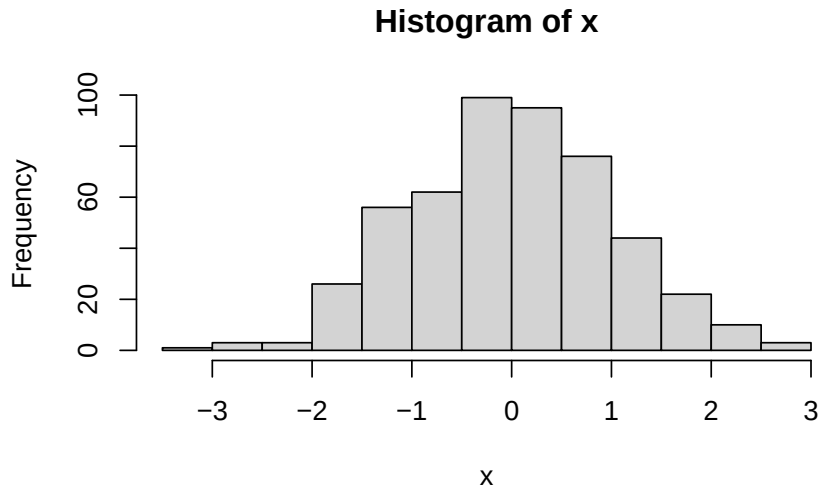
(a) n = 500

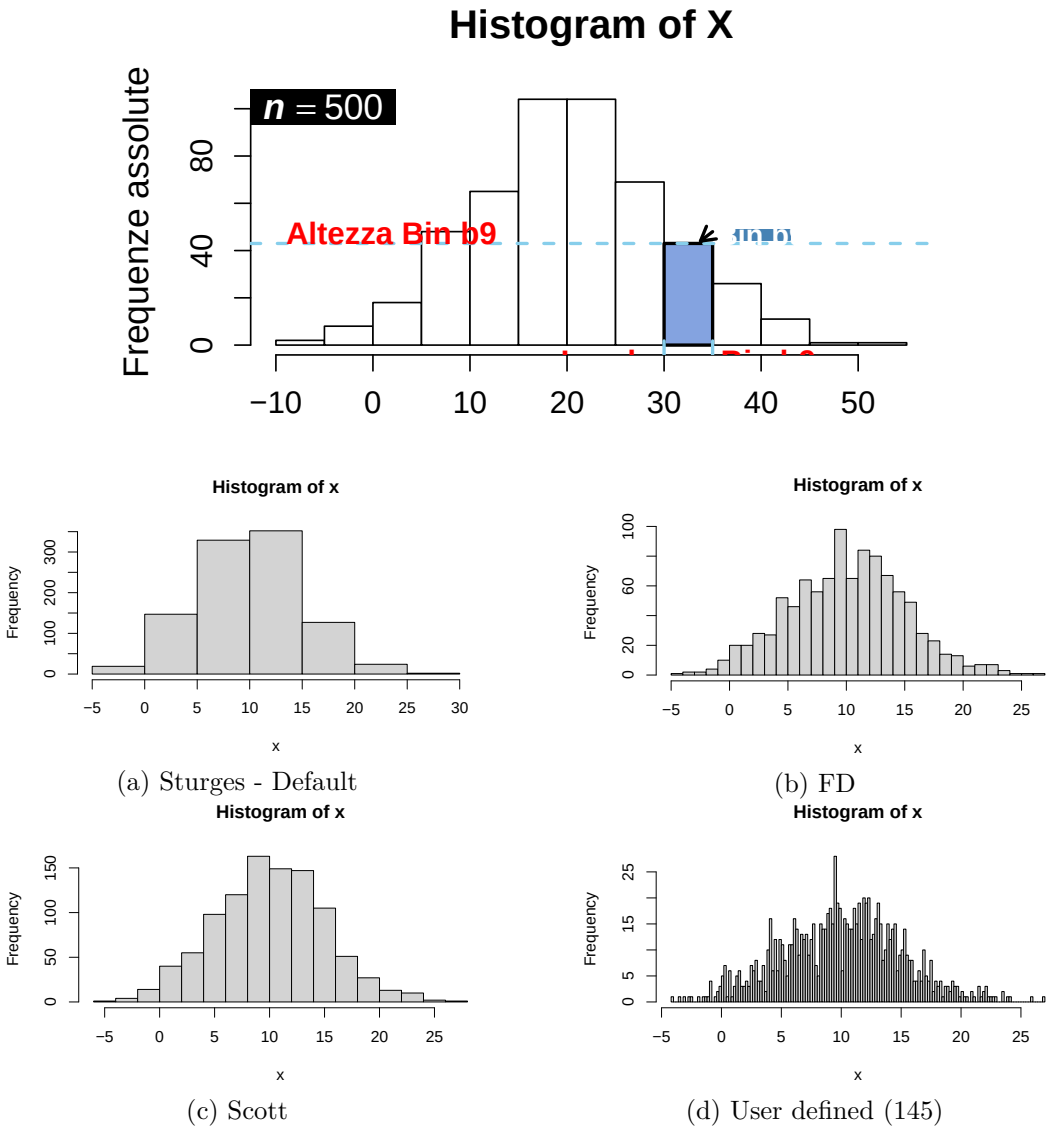


(a) n = 50

Istogrammi con n diverso

```r
x = rnorm(500, mean = 0, sd = 1)
hist(x)
```





(a) Sturges - Default

(b) FD

(c) Scott

(d) User defined (145)

Figure 2.6: Metodi per la definizione dei breaks

Sturges:

$$\frac{\max(x) - \min(x)}{\lceil \log_2(n) + 1 \rceil}$$

Scott:

$$3.5\sqrt{\mathrm{var}(x)}\, n^{-1/3}$$

Freedman-Diaconis

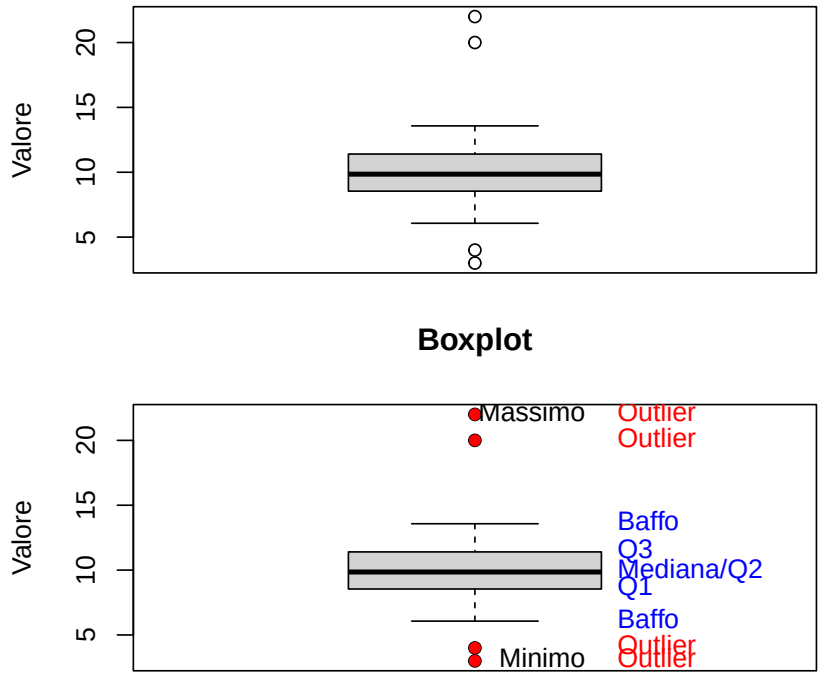
$$2\,(Q_3 - Q_1)\, n^{-1/3}$$

User defined:

$$\frac{\max(x) - \min(x)}{\text{breaks}}$$

2.7.3 Boxplot

`boxplot(x)` produce un boxplot a partire da un vettore `x` di osservazioni quantitative.



2.7.4 Boxplot – Calcolo degli outlier

La lunghezza dei baffi (e di conseguenza la definizione degli outliers) è definita attraverso il metodo Tukey:

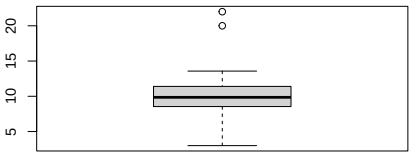
$$[Q_1 - 1.5 \times IQR,\ Q_3 + 1.5 \times IQR]$$

dove $IQR = Q_3 - Q_1$ (Inter-quartile range)

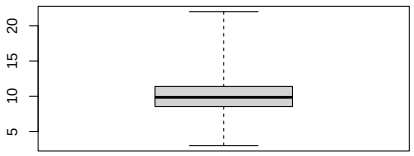
Tutti i valori che cadono al di fuori dei baffi, sono definiti outliers.

Settando l'argomento `range = x` si può cambiare il modo in cui vengono calcolati i baffi:

```
boxplot(x, range = 2)
boxplot(x, range = 0)
```



(a) range = 2 - Meno Conservativo



(a) range = 0 - Minimo e Massimo

Calcolo degli outlier

2.8 Esempi

2.8.1 `barplot()`

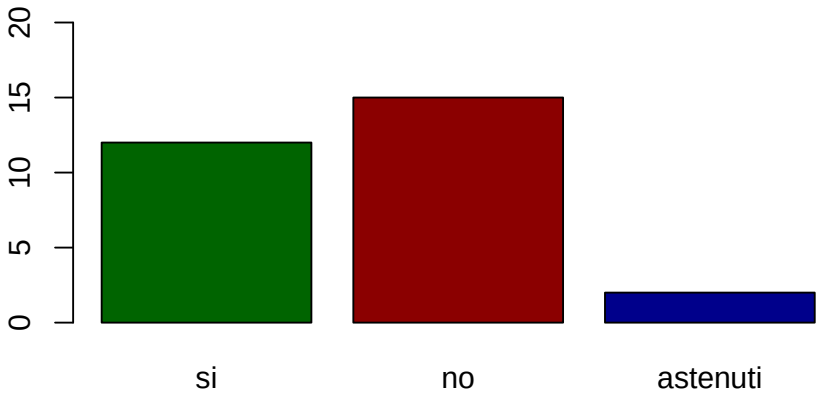
```
voto = sample(rep(c("si", "no", "astenuti"),
           c(12, 15, 2)))
voto = factor(voto, levels = c("si", "no", "astenuti"))
table(voto)

# le frequenze restituite dalla funzione table() vanno usate
# come vettore di base per il barplot

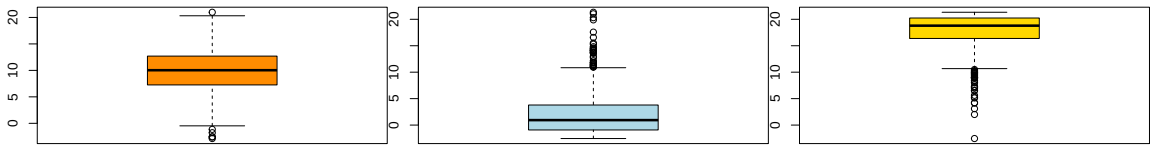
barplot(table(voto),
        col = c("darkgreen", "darkred", "darkblue"),
        ylim = c(0, 20))
```

```
voto
      si       no astenuti
      12       15        2
```

```
barplot(table(voto),
        col = c("darkgreen", "darkred", "darkblue"),
        ylim = c(0, 20))
```



2.8.2 `boxplot()`



(a) boxplot(x) (b) boxplot(y) (c) boxplot(z)

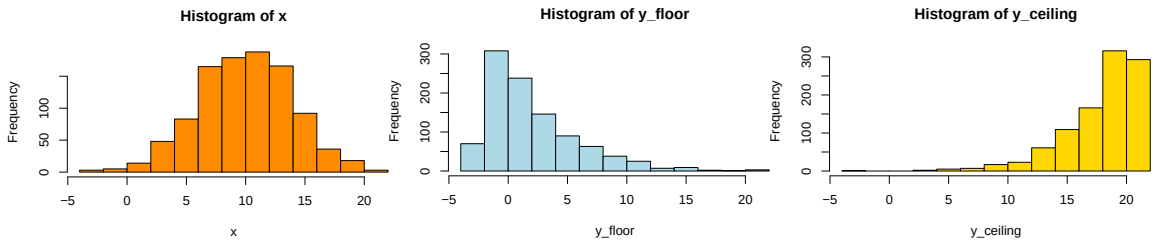
Figure 2.9: Distribuzioni a confronto

Figure 2.9a: La distribuzione è equilibrata attorno alla mediana, situata circa a metà della distribuzione tra il minimo e il massimo

Figure 2.9b: La distribuzione è squilibrata verso il basso, ovvero verso il minimo, presentando una coda più "pesante" verso il massimo

Figure 2.9c: La distribuzione è squilibrata verso l'alto, ovvero verso il massimo, presentando una coda più "pesante" verso il minimo

2.8.3 `hist()`



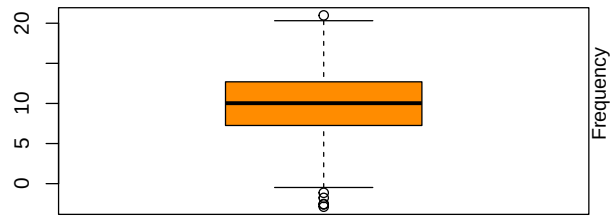
(a) hist(x) (b) hist(y) (c) hist(z)

Figure 2.10: Distribuzioni a confronto - Le stesse dei boxplot

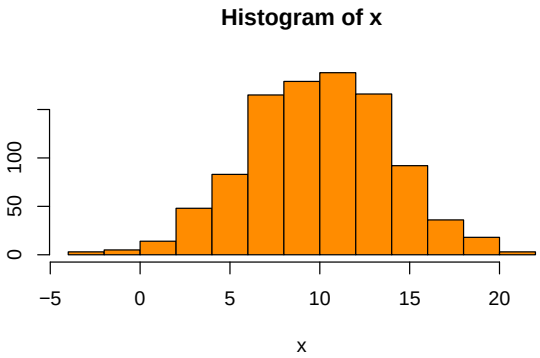
2.8.4 Confronto boxplot vs istogramma

Forzando i boxplot ad essere in orizzontale attraverso l'argomento `horizontal = TRUE` (Figure **??**) si può apprezzare meglio la similitudine tra la forma della distribuzione vista con l'istogramma e la forma della distribuzione vista con il boxplot.

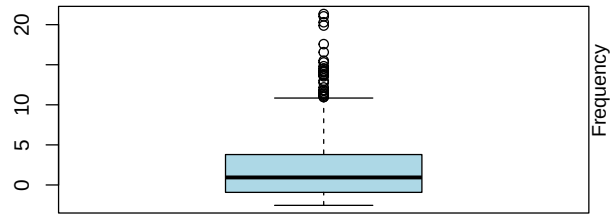
Figure 2.13 illustra gli stessi boxplot disposti nella stessa finestra (Listing 2.3).



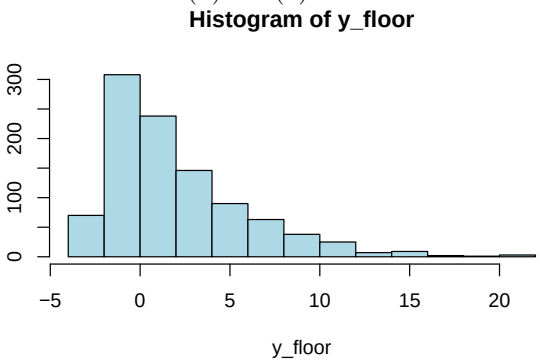
(a) boxplot(x)



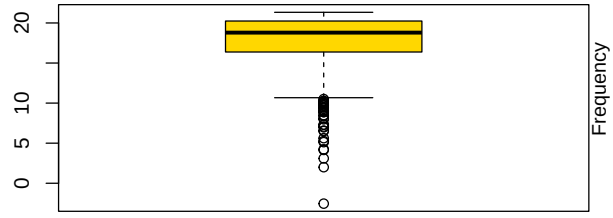
(b) hist(x)



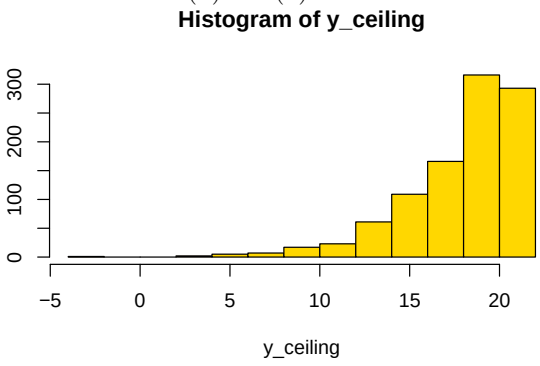
(c) boxplot(x)



(d) hist(x)

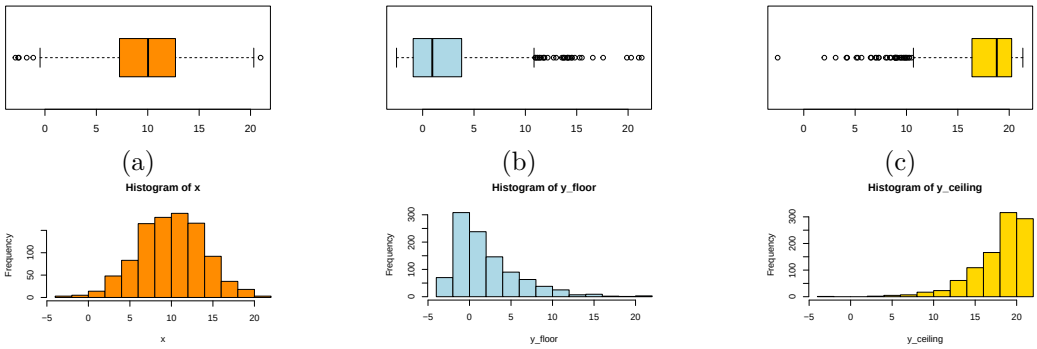


(e) boxplot(x)



(f) hist(x)

Figure 2.11: Boxplot vs Istogramma



Listing 2.3 Finestra grafica con molti boxplot

```
boxplot(x, y, z,
        names = c("Normale", "Pavimento", "Soffitto"),
        col = c("darkorange", "lightblue", "gold"))
```

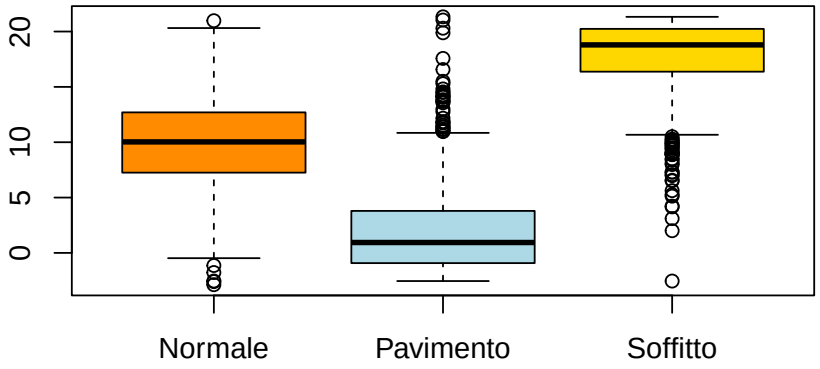


Figure 2.13: Tanti boxplot in un unico grafico

3 Modulo 3 – Statistica descrittiva bivariata

3.1 Variabile statistica doppia

Si consideri un gruppo di 11 persone, per ciascuna delle quali sono state rilevate due variabili categoriali: il titolo di studio (X, con categorie Elementare, Secondaria Inferiore e Secondaria Superiore) e l'occupazione (Y, con categorie Disoccupato, Impiegato e Dirigente). I dati sono riportati in Table 3.1.

Ciascuna delle due variabili può essere descritta separatamente attraverso la sua distribuzione di frequenza. Table 3.2 riporta le frequenze del titolo di studio ($n_{k.}$, il numero di persone che possiedono la categoria x_k), mentre Table 3.2a riporta quelle dell'occupazione ($n_{.j}$, il numero di persone che possiedono la categoria y_j).

Le distribuzioni univariate non dicono nulla su come le due variabili si combinano tra loro: sappiamo quante persone hanno un titolo di studio elementare e quante sono dirigenti, ma non quante persone con un titolo elementare sono dirigenti. Per rispondere a questa domanda bisogna considerare le due variabili *congiuntamente*, contando quante persone presentano ciascuna coppia di categorie (x_k, y_j). Il risultato è la **tabella a doppia entrata** (o tavola di contingenza) in Table 3.3: le righe corrispondono alle categorie di X, le colonne alle categorie di Y, e ogni cella contiene la **frequenza congiunta** n_{kj}, cioè il numero di persone che possiedono contemporaneamente la categoria x_k e la categoria y_j.

Ad esempio, la cella all'incrocio tra la riga SS e la colonna D indica che 3 persone hanno un titolo di studio di secondaria superiore *e* sono dirigenti.

Sommando le frequenze congiunte lungo ciascuna riga si ottengono i totali di riga (in rosso), mentre sommandole lungo ciascuna colonna si ottengono i totali di colonna (in blu). Questi totali sono detti **frequenze marginali**, perché si trovano ai margini della tabella. Confrontando Table 3.3 con Table 3.2 e Table 3.2a si nota che le frequenze marginali di riga coincidono esattamente con la distribuzione di frequenza di X e le frequenze marginali di colonna coincidono con la distribuzione di frequenza di Y. In altre parole, la tabella a doppia entrata contiene al suo interno anche le due distribuzioni univariate. Infine, la cella in basso a destra (11) riporta il numero totale di unità statistiche n, che si ottiene sia sommando i totali di riga sia sommando i totali di colonna.

Table 3.1: Titolo di studio e occupazione rilevati su 11 persone

X: Istruzione	E	E	E	SI	SI	SI	SS	SS	SS	SS	SS
Y: Occupazione	DI	I	I	DI	I	I	D	I	I	D	D

Table 3.2: Distribuzioni di frequenza univariate.

(a) Variabile Istruzione

categoria x_k	$n_{k.}$
Elementare (E)	3
Secondaria Inferiore (SI)	3
Secondaria Superiore (SS)	5

(b) Variabile Occupazione

categoria y_j	$n_{.j}$
Disoccupato (DI)	2
Impiegato (I)	6
Dirigente (D)	3

Table 3.3: Tabella a doppia entrata di Istruzione e Occupazione

	DI	I	D	
E	1	2	0	3
SI	1	2	0	3
SS	0	2	3	5
	2	6	3	**11**

Più in generale, date due variabili X e Y, la loro distribuzione di frequenze congiunte n_{kj} descrive quante unità statistiche presentano ciascuna coppia di categorie (x_k, y_j).

Tip

L'insieme (X, Y) delle terne

$$(x_k, y_j, n_{kj}) \qquad (k = 1, \dots, h;\ j = 1, \dots, m)$$

è detto **variabile statistica doppia**.

3.1.1 Tabella a doppia entrata – Tavola di contingenza

		Y						
		y_1	y_2	...	y_j	...	y_m	$n_{k.}$
X	x_1	n_{11}	n_{12}	...	n_{1j}	...	n_{1m}	$n_{1.}$
	x_2	n_{21}	n_{22}	...	n_{2j}	...	n_{2m}	$n_{2.}$
	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	x_k	n_{k1}	n_{k2}	...	n_{kj}	...	n_{km}	$n_{k.}$
	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	x_h	n_{h1}	n_{h2}	...	n_{hj}	...	n_{hm}	$n_{h.}$
		$n_{.1}$	$n_{.2}$	...	$n_{.j}$	...	$n_{.m}$	n

n_{kj}: **Frequenza congiunta**, definisce il numero di unità statistiche che possiedono congiuntamente la categoria x_k della variabile X e la categoria y_j della variabile Y

$n_{k.}$: **Frequenza marginale di** X, definisce il numero di unità statistiche che possiedono la categoria x_k della variabile X

$n_{.j}$: **Frequenza marginale di** Y, definisce il numero di unità statistiche che possiedono la categoria y_j della variabile Y

n: Numero totale di unità statistiche

$$n_{k.} = \sum_{j=1}^{m} n_{kj} \longrightarrow$$ Numero di unità statistiche che possiedono la categoria x_k della variabile X (**totale della riga** k)

$$n_{.j} = \sum_{k=1}^{h} n_{kj} \longrightarrow$$ Numero di unità statistiche che possiedono la categoria y_j della variabile Y (**totale della colonna** j)

$$\sum_{k=1}^{h}\sum_{j=1}^{m} n_{kj} = \sum_{k=1}^{h} n_{k.} = \sum_{j=1}^{m} n_{.j} = n$$

3.1.2 `table()`

`table(var1, var2)`: Produce la tabella di frequenza tra due variabili categoriali `var1` e `var2`

```
var1 <- as.factor(c("a1","a2","a1","a2","a1","a2","a1"))
var2 <- as.factor(c("b1","b1","b1","b2","b1","b2","b2"))
mytab = table(var1, var2); mytab
```

```
    var2
var1 b1 b2
  a1  3  1
  a2  1  2
```

```
sum(mytab) # permette di ottenere l'ampiezza campionaria
```

```
[1] 7
```

3.1.3 `margin.table()`

`margin.table(table, margin)`: Frequenze marginali per la tabella a doppia entrata

> Tip
>
> `margin = 1`: Frequenze marginali calcolate **attraverso le colonne**
> `margin = 2`: Frequenze marginali calcolate **attraverso le righe**

```
mytab
```

```
    var2
var1 b1 b2
  a1  3  1
  a2  1  2
```

```
margin.table(mytab,1)
```

```
var1
a1 a2
 4  3
```

```
margin.table(mytab,2)
```

```
var2
b1 b2
 4  3
```

3.1.4 `prop.table()`

`prop.table(T, margin = NULL)`: Probabilità condizionate:

> Tip
>
> `margin = NULL`: Probabilità calcolate sul totale n, $p_{kj} = \frac{n_{kj}}{n}$
> `margin = 1`: Probabilità condizionate di riga (**attraverso le colonne**), $p_{j|k} = \frac{n_{kj}}{n_{k.}}$
> `margin = 2`: Probabilità condizionate di colonna (**attraverso le righe**), $p_{k|j} = \frac{n_{kj}}{n_{.j}}$

```
mytab
```

```
    var2
var1 b1 b2
  a1  3  1
  a2  1  2
```

```
prop.table(mytab)
```

```
    var2
var1        b1        b2
  a1 0.4285714 0.1428571
  a2 0.1428571 0.2857143
```

```
prop.table(mytab,1)  # P(B|A)
```

```
prop.table(mytab,2)  # P(A|B)
```

```
    var2
var1        b1        b2
  a1 0.7500000 0.2500000
  a2 0.3333333 0.6666667
```

```
    var2
var1        b1        b2
  a1 0.7500000 0.3333333
  a2 0.2500000 0.6666667
```

3.1.5 Frequenze percentuali e frequenze condizionate

Si consideri un campione di 140 donne per le quali sono state rilevate due variabili categoriali: l'occupazione (casalinga o lavoratrice) e il livello di istruzione (basso, medio o alto). La tabella a doppia entrata con le frequenze assolute e le marginali è riportata in Table 3.4.

Table 3.4: Tabella a doppia entrata di Occupazione e Istruzione (frequenze assolute)

	Basso	Medio	Alto	Totale
Casalinghe	40	25	20	85
Lavoratrici	10	15	30	55
Totale	50	40	50	140

Guardando le frequenze assolute si potrebbe essere tentati di concludere, ad esempio, che tra le donne con istruzione alta ci sono più lavoratrici (30) che casalinghe (20), e che tra quelle con istruzione bassa ci sono molte più casalinghe (40) che lavoratrici (10). Le frequenze assolute, però, non permettono confronti corretti, perché i gruppi hanno dimensioni diverse: nel campione ci sono 85 casalinghe e solo 55 lavoratrici, e anche i tre livelli di istruzione non hanno la stessa numerosità (50, 40 e 50). Un numero più alto in una cella può dipendere semplicemente dal fatto che quel gruppo è più numeroso.

Per confrontare gruppi di dimensione diversa bisogna passare alle **frequenze percentuali condizionate**: invece di dividere ogni cella per il totale generale n (quello che fa la funzione `prop.table(tabella)` quando non viene specificato un margin), la si divide per il totale della sua riga o della sua colonna. In questo modo ogni riga (o ogni colonna) diventa una distribuzione percentuale che somma a 100, e i gruppi diventano confrontabili.

La stessa tabella si può leggere in due modi diversi, a seconda di quale variabile si sceglie come *condizionante*, cioè di quale variabile si fissano le categorie per guardare come si distribuisce l'altra.

Un primo colpo d'occhio sulla struttura della tabella è dato dal mosaic plot in Figure 3.1. La larghezza delle due colonne è proporzionale al numero di casalinghe e di lavoratrici, mentre l'altezza dei rettangoli all'interno di ciascuna colonna rappresenta la percentuale di donne con istruzione bassa, media o alta *in quel gruppo*.

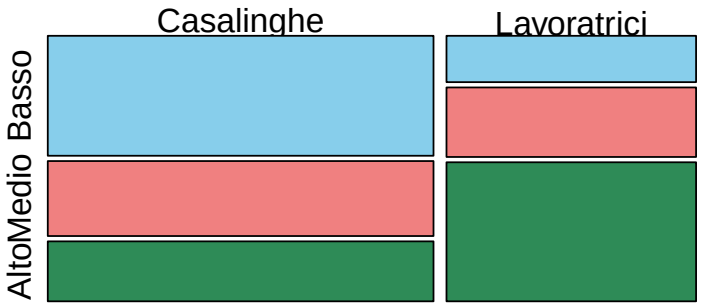


Figure 3.1: Mosaic plot di Occupazione e Istruzione

3.1.5.1 Frequenze percentuali di riga: $P(\textbf{Istruzione} \mid \textbf{Occupazione})$

Le percentuali di **riga** si ottengono dividendo ogni cella per il totale della propria riga (`prop.table(tabella, 1)`). Poiché le righe corrispondono all'occupazione, ogni riga descrive *come si distribuisce il titolo di studio all'interno di ciascun gruppo occupazionale* (Table 3.5 e Figure 3.2).

Table 3.5: Frequenze percentuali di riga: distribuzione dell'Istruzione condizionata all'Occupazione

	Basso	Medio	Alto	Totale
Casalinghe	47	29	24	100
Lavoratrici	18	27	55	100

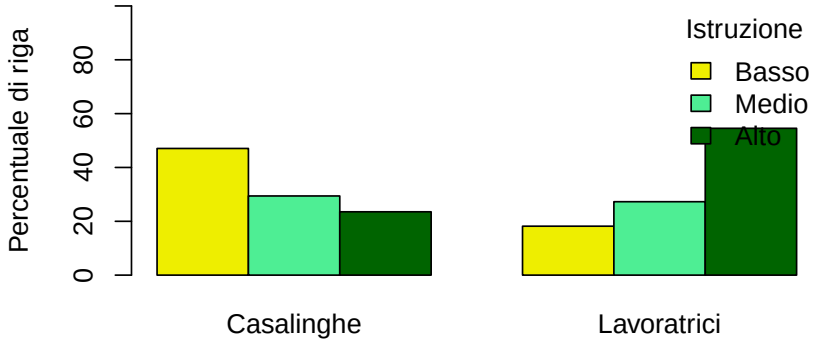


Figure 3.2: Distribuzione percentuale dell'Istruzione nei due gruppi occupazionali

La lettura corretta è sempre del tipo *"tra le donne [categoria di occupazione], il X% ha un titolo di studio [categoria di istruzione]"*. Ad esempio:

- tra le casalinghe, il 47% ha un titolo di studio basso e solo il 24% un titolo alto;
- tra le lavoratrici, il 55% ha un titolo di studio alto e solo il 18% un titolo basso.

Le percentuali di riga descrivono quindi la **composizione** di ciascun gruppo occupazionale: rispondono a domande del tipo *"che titolo di studio hanno le casalinghe?"*.

3.1.5.2 Frequenze percentuali di colonna: $P(\textbf{Occupazione} \mid \textbf{Istruzione})$

Le percentuali di **colonna** si ottengono dividendo ogni cella per il totale della propria colonna (`prop.table(tabella, 2)`). Poiché le colonne corrispondono al titolo di studio, ogni colonna descrive *come si distribuisce l'occupazione all'interno di ciascun livello di istruzione* (Table 3.6 e Figure 3.3).

Table 3.6: Frequenze percentuali di colonna: distribuzione dell'Occupazione condizionata all'Istruzione

	Basso	Medio	Alto
Casalinghe	80	62	40
Lavoratrici	20	38	60
Totale	100	100	100

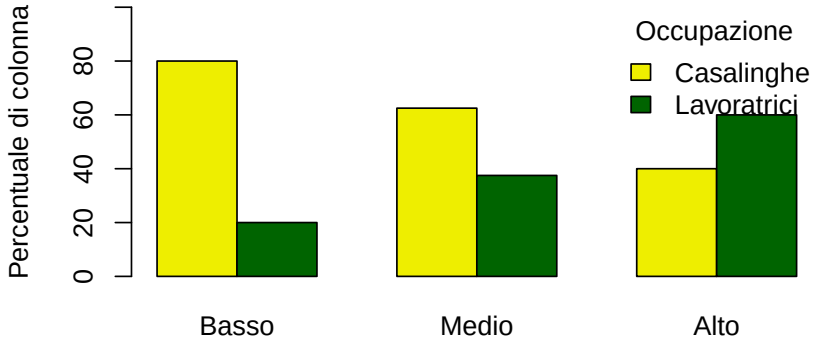


Figure 3.3: Distribuzione percentuale dell'Occupazione nei tre livelli di istruzione

In questo caso la lettura è del tipo *"tra le donne con titolo di studio [categoria di istruzione], il X% è [categoria di occupazione]"*. Ad esempio:

- tra le donne con titolo di studio basso, l'80% è casalinga e solo il 20% lavora;
- tra le donne con titolo di studio alto, la situazione si capovolge: il 60% lavora e il 40% è casalinga.

Al crescere del titolo di studio, quindi, cresce la percentuale di lavoratrici (20%, 38%, 60%).

3.1.5.3 Quale percentuale ha senso calcolare?

Dal punto di vista del calcolo, entrambe le percentuali sono sempre ottenibili, e R le restituisce senza alcun problema. La scelta di quale usare dipende dalla **domanda di ricerca**, cioè da quale variabile ha senso considerare come condizionante e quale come variabile da descrivere.

In questo esempio è più naturale ragionare in termini di $P(\text{Occupazione} \mid \text{Istruzione})$, cioè con le percentuali di **colonna**. Il titolo di studio, infatti, si acquisisce di norma *prima* di entrare nel mondo del lavoro: ha senso chiedersi se e come l'occupazione cambi al variare del titolo di studio, mentre ha molto meno senso pensare che essere casalinga o lavoratrice determini il titolo di studio posseduto. Le percentuali di riga restano comunque utili a scopo descrittivo (per sapere, ad esempio, com'è composto il gruppo delle casalinghe), ma non vanno lette come se l'occupazione "spiegasse" l'istruzione.

Una regola pratica

Si condiziona sulla variabile che si pensa possa *spiegare* o *precedere* l'altra, e si calcolano le percentuali all'interno delle sue categorie. Poi si confrontano le percentuali **tra** le categorie della variabile condizionante: se le distribuzioni sono diverse tra loro, le due variabili sono in qualche modo associate.

Condizionare non vuol dire dimostrare una causa

Scegliere una variabile come condizionante è una scelta di lettura dei dati, non una prova di un legame causale. Che tra le donne con istruzione alta ci siano più lavoratrici non dimostra che sia il titolo di studio a *causare* l'occupazione: le due variabili potrebbero dipendere entrambe da altri fattori, ad esempio la condizione economica della famiglia di origine.

Infine, se le due variabili fossero **indipendenti**, le distribuzioni condizionate sarebbero tutte uguali tra loro e uguali alla distribuzione marginale. In questo caso, ad esempio, in ogni livello di istruzione ci si aspetterebbe di trovare la stessa percentuale di lavoratrici osservata nel campione complessivo, cioè $55/140 \approx 39\%$. Le percentuali osservate (20%, 38% e 60%) sono invece molto diverse tra loro: è proprio questa distanza dalla situazione di indipendenza che l'indice χ^2, presentato nella sezione successiva, permette di quantificare.

3.2 Connessione (Variabili qualitative)

La CONNESSIONE tra due variabili categoriali equivale alla NON INDIPENDENZA!

Tra le variabili X e Y c'è una forma di legame e **relazione** (**connessione**) $\rightarrow$ il variare di una implica un qualche cambiamento **non casuale** nell'altra.

3.2.1 χ^2

Quantifica la DISTANZA DALL'INDIPENDENZA tra due variabili categoriali.

- X con h categorie x_k, con $k = 1, \dots, h$,
- Y con m categorie y_j, con $j = 1, \dots, m$

Richiede il confronto tra **frequenze osservate** (n_{kj}) e **frequenze attese** o **frequenze teoriche** (n^*_{kj}).

n_{kj}	$n_{k.}$
$n_{.j}$	

VS.

n^*_{kj}	$n_{k.}$
$n_{.j}$	

Frequenze osservate Frequenze teoriche

n^*_{kj}: Significato

Data l'ampiezza campionaria n, quali sono le frequenze che avremmo dovuto osservare se le due variabili fossero state indipendenti, ovvero se non avessero alcuna relazione?

$$n^*_{kj} = \frac{n_{k.} \times n_{.j}}{n}, \qquad n = \sum_{k=1}^{h}\sum_{j=1}^{m} n_{kj} = \sum_{k=1}^{h}\sum_{j=1}^{m} n^*_{kj}$$

Siccome le frequenze teoriche sono le frequenze che si sarebbero osservate se le variabili fossero state indipendenti **data l'ampiezza campionaria e i marginali di riga e colonna osservati**, la somma di tutte le frequenze teoriche restituisce l'ampiezza campionaria n. Inoltre, i marginali di riga delle frequenze teoriche sono uguali ai marginali di riga delle frequenze osservate, così come i marginali di colonna delle frequenze teoriche sono uguali ai marginali di colonna delle frequenze osservate.

```
knitr::kable(tabella,booktabs = TRUE,align = "c")
```

Table 3.7: Frequenze osservate

	Basso	Medio	Alto	Totale
Casalinghe	40	25	20	85
Lavoratrici	10	15	30	55
Totale	50	40	50	140

Table 3.8: Frequenze teoriche

	Basso	Medio	Alto	Totale
Casalinghe	$\frac{85 \times 50}{140} = 30.36$	$\frac{85 \times 40}{140} = 24.29$	$\frac{85 \times 50}{140} = 30.36$	85
Lavoratrici	$\frac{55 \times 50}{140} = 19.64$	$\frac{55 \times 40}{140} = 15.71$	$\frac{55 \times 50}{140} = 19.64$	55
Totale	50	40	50	**140**

$$\chi^2 = \sum_{k=1}^{h}\sum_{j=1}^{m} \frac{(n_{kj} - n^*_{kj})^2}{n^*_{kj}}$$

$\chi^2 = 0$: $n_{kj} = n^*_{kj}$ Indipendenza stocastica

$\chi^2 = \chi^2_{\max}$: Relazione di perfetta dipendenza (dove $\chi^2_{\max} = n \times \min[(h-1);(m-1)]$)

$$\tilde{\chi}^2 = \frac{\chi^2}{\chi^2_{\max}}$$

$\tilde{\chi}^2$ è il χ^2 normalizzato, permette di esprimere il grado di associazione tra due variabili categoriali secondo una scala di tipo $[0, 1]$

3.2.2 `chisq.test()`

`chisq.test(table)$statistic`: Calcola il χ^2 osservato a partire da una tavola di contingenza ottenuta con `table()`

```
mytab
```

```
    var2
var1 b1 b2
  a1  3  1
  a2  1  2
```

```
chisq.test(mytab)$statistic
```

```
X-squared
 0.109375
```

3.2.2.1 Calcolo $\chi^2_{\max}$

```
chiMax = sum(mytab) * min(c(nrow(mytab) - 1, ncol(mytab) - 1))
chiMax
```

```
[1] 7
```

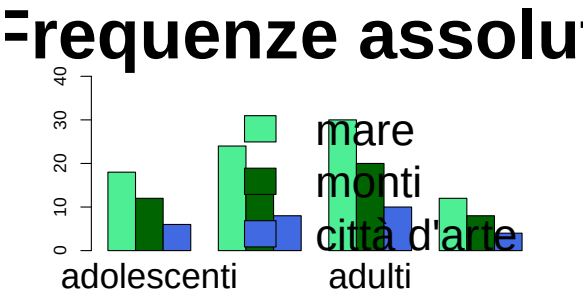
```
chisq.test(mytab)$statistic/chiMax
```

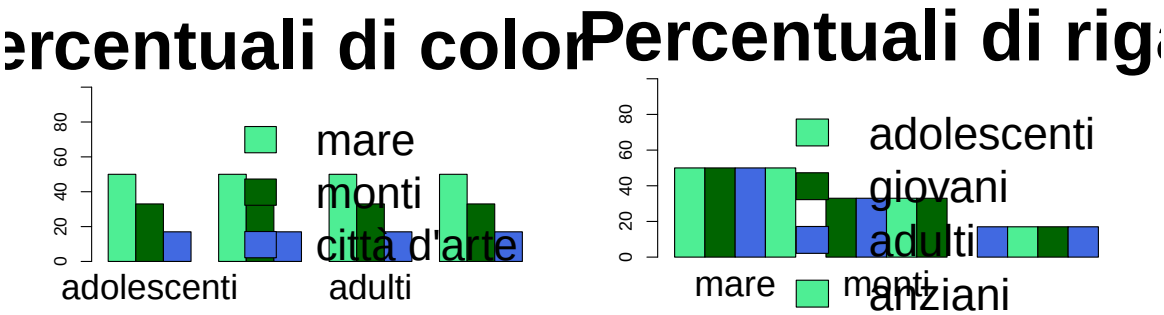
```
Warning in chisq.test(mytab): L'approssimazione al Chi-quadrato potrebbe essere
inesatta
```

```
X-squared
 0.015625
```

3.3 Indipendenza stocastica e massima dipendenza

	adolescenti	giovani	adulti	anziani
mare	18	24	30	12
monti	12	16	20	8
città d'arte	6	8	10	4





Frequenze osservate

```
obs_n = matrix(
  c(18, 24, 30, 12,    # mare
    12, 16, 20, 8,     # monti
    6, 8, 10, 4),      # città d'arte
  nrow = 3, byrow = TRUE) ; obs_n
```

```
     [,1] [,2] [,3] [,4]
[1,]   18   24   30   12
[2,]   12   16   20    8
[3,]    6    8   10    4
```

Frequenze teoriche

```
chisq.test(obs_n)$expected
```

```
     [,1] [,2] [,3] [,4]
[1,]   18   24   30   12
[2,]   12   16   20    8
[3,]    6    8   10    4
```

3.3.0.1 Calcolo χ^2

```
chisq.test(obs_n)$statistic
```

```
Warning in chisq.test(obs_n): L'approssimazione al Chi-quadrato potrebbe essere
inesatta
```

```
X-squared
        0
```

```
sum(obs_n) * min(c(nrow(obs_n) -1,
                   ncol(obs_n) - 1))
```

```
[1] 336
```

	adolescenti	giovani	adulti	Sum
mare	84	0	0	84
monti	0	64	0	64
città d'arte	0	0	20	20
Sum	84	64	20	168



Frequenze osservate

```
obs_n <- matrix(
  c(84, 0, 0,    # mare → solo adolescenti
    0, 64, 0,    # monti → solo giovani
    0, 0, 20),  # città d'arte → solo adulti
  nrow = 3, byrow = TRUE) ; obs_n
```

```
     [,1] [,2] [,3]
[1,]   84    0    0
[2,]    0   64    0
```

Frequenze teoriche

```
chisq.test(obs_n)$expected
```

```
     [,1]      [,2]      [,3]
[1,]   42 32.000000 10.000000
[2,]   32 24.380952  7.619048
[3,]   10  7.619048  2.380952
```

```
[3,]    0    0   20
```

Table 3.11: Frequenze osservate

84	0	0
0	64	0
0	0	20

Table 3.12: Frequenze Teoriche

42	32.000000	10.000000
32	24.380952	7.619048
10	7.619048	2.380952

```
chisq.test(obs_n)$statistic
```

```
Warning in chisq.test(obs_n): L'approssimazione al Chi-quadrato potrebbe essere
inesatta
```

```
X-squared
      336
```

```
sum(obs_n) * min(c(nrow(obs_n) -1,
                   ncol(obs_n) - 1))
```

```
[1] 336
```

3.4 Associazione (variabili quantitative)

3.4.1 Relazione lineare

Si vuole studiare la **relazione lineare** tra due variabili quantitative X e Y misurate sulle stesse unità statistiche. Per ciascuna unità i (con $i = 1, \dots, n$) si dispone quindi di una coppia di valori (x_i, y_i). Ad esempio, x_i può essere l'età e y_i il peso della stessa persona, oppure le ore di studio e il voto all'esame dello stesso studente.

Le coppie (x_i, y_i) possono essere rappresentate come punti su un piano cartesiano, con X sull'asse orizzontale e Y sull'asse verticale. Questo è il **grafico a dispersione** (*scatterplot*, Figure 3.4), che è il primo strumento da utilizzare per farsi un'idea della relazione tra due variabili quantitative.

Per relazione lineare si intende quanto le coppie di punti (x_i, y_i) tendono a disporsi lungo una retta. Osservando il grafico a dispersione, la relazione può essere descritta in base a due caratteristiche:

- **Direzione**: la relazione è *positiva* se all'aumentare dei valori di X tendono ad aumentare anche i valori di Y (la nuvola di punti "sale" da sinistra verso destra). È *negativa* se all'aumentare dei valori di X i valori di Y tendono a diminuire (la nuvola di punti "scende").
- **Forza**: quanto i punti sono vicini alla retta. Più i punti sono stretti attorno alla retta, più la relazione lineare è forte. Più sono sparsi, più la relazione è debole. Se i punti non mostrano nessuna tendenza a disporsi lungo una retta, non c'è relazione lineare tra le due variabili.

L'assenza di relazione *lineare* non implica l'assenza di *qualsiasi* relazione. Due variabili possono essere legate da una relazione non lineare, ad esempio a forma di U o di U rovesciata, che gli indici presentati in questa sezione non sono in grado di cogliere. Per questo motivo è sempre buona norma guardare il grafico a dispersione prima di calcolare qualsiasi indice.

Si hanno a disposizione due indici, l'indice di **covarianza**, che permette di studiare la direzione della relazione tra le variabili, e il **coefficiente di correlazione lineare di Bravais-Pearson**, che permette di studiarne la direzione E l'intensità.

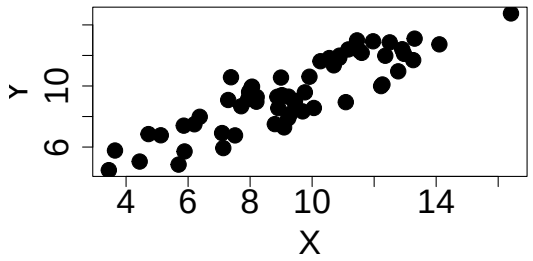


Figure 3.4: Grafico a dispersione delle variabili X e Y

Due indici: la **covarianza** (direzione della relazione) e il **coefficiente di correlazione lineare di Bravais-Pearson** (direzione e intensità)

3.4.2 Indice di covarianza campionaria

Scatter plot in R

`plot(x, y)`: Grafico di dispersione di due variabili quantitative appaiate `(x,y)` aventi lo stesso numero di osservazioni

In Figure 3.5 è riportato lo scatterplot tra due variabili. I punti rossi rappresentano le coppie di osservazioni (x_i, y_i) di ciascuna unità statistica i. Le linee tratteggiate indicano la media di X (linea verticale) e la media di Y (linea orizzontale), che diventano i **baricentri** per il calcolo della covarianza.

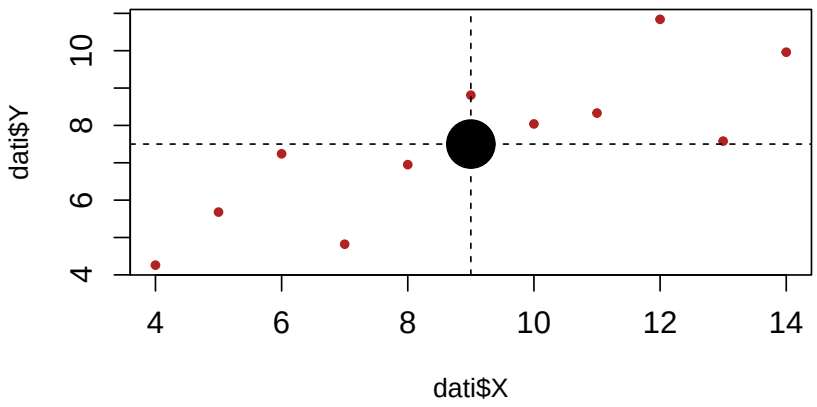
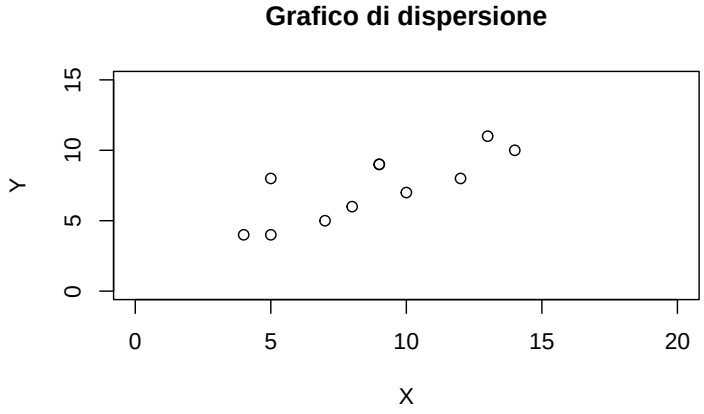


Figure 3.5: Grafico a dispersione delle due variabili X e Y.

I baricentri danno origine a un nuovo sistema di assi cartesiani, la cui origine è il punto di incontro tra le due medie, $(\bar{x}, \bar{y})$. In questo nuovo sistema, ogni punto è descritto dai suoi **scarti** dalle medie, $(x_i - \bar{x})$ e $(y_i - \bar{y})$, e il piano risulta diviso in quattro quadranti.

La covarianza è la media dei prodotti di questi scarti. Il segno del prodotto $(x_i - \bar{x})(y_i - \bar{y})$ dipende dal quadrante in cui cade il punto: è positivo quando entrambi gli scarti hanno lo stesso segno (quadranti in alto a destra e in basso a sinistra) e negativo quando hanno segno opposto (quadranti in alto a sinistra e in basso a destra). Se la maggior parte dei punti cade nei quadranti "positivi", la covarianza sarà positiva; se cade in quelli "negativi", sarà negativa.

```
X <- c(9,8,12,9,10,14,5,4,13,7,5)
Y <- c(9,6,8,9,7,10,8,4,11,5,4)
plot(X,Y,main="Grafico di dispersione", xlim=c(0,20),ylim=c(0,15))
```



La covarianza è quindi una misura del grado di *dispersione* e *covariazione* di due variabili quantitative appaiate (X, Y) attorno ai loro baricentri (medie) $\bar{x}$ e $\bar{y}$

$x = (x_1, x_2, \ldots, x_n)$

$y = (y_1, y_2, \dots, y_n)$

$$d = ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$$

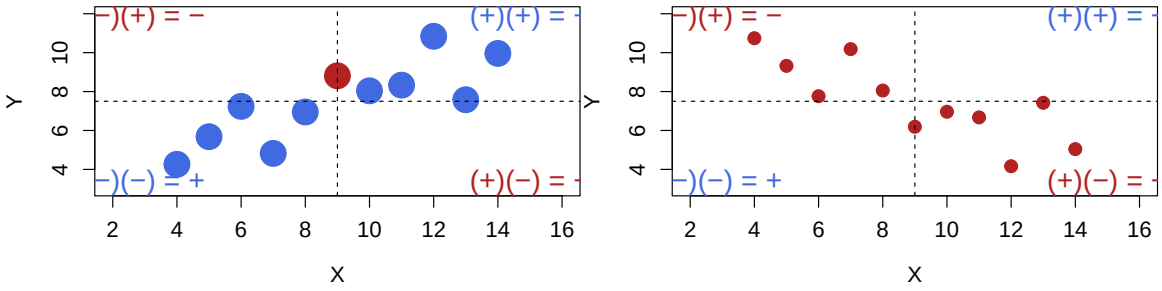
$$\text{Cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \tag{3.1}$$

È (quasi) la **media dei prodotti degli scarti** di ogni variabile dalla propria media.

> Note
>
> Nota: alcuni testi dividono per n invece che per $n-1$ (covarianza "descrittiva"). La funzione `cov()` di `R` divide per $n-1$, quindi useremo sempre questa versione. Per coerenza, in questo modulo $sd(X)$ indica la deviazione standard calcolata dalla funzione `sd()`, che divide anch'essa per $n-1$ (a differenza di σ del Modulo 2, che divide per n). Nel calcolo di r e di $\hat{\beta}$ la scelta non cambia il risultato, purché covarianza e deviazioni standard usino lo stesso divisore.

3.4.3 Il "nuovo" sistema cartesiano

Figure 3.6 riporta due esempi per illustrare l'andamento della nuvola dei punti in base al segno dei prodotti tra gli scarti, sapendo che il prodotto $(x_i - \bar{x})(y_i - \bar{y})$ è più spesso positivo se $\text{Cov}(X, Y) > 0$ (Figure 3.6a), più spesso negativo se $\text{Cov}(X, Y) < 0$ (Figure 3.6b).



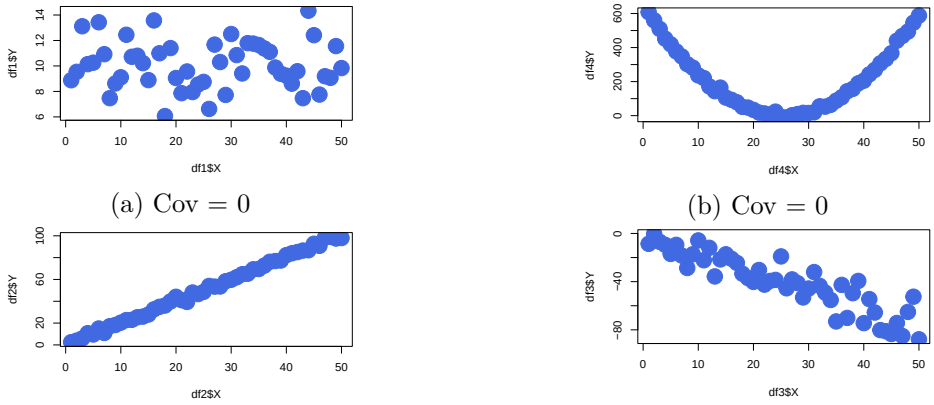
(a) $\text{Cov}(X, Y) > 0$ (b) $\text{Cov}(X, Y) < 0$

Figure 3.6: Covarianza

Chiaramente, se gli scarti sono entrambi positivi (quindi sono sopra la media di X e Y), il loro prodotto sarà positivo. Se sono entrambi negativi (ovvero entrambi sotto la media di X e Y), il loro prodotto sarà comunque positivo. I punti dello scatter plot si troveranno (*per la maggior parte*) nel primo e terzo quadrante (Figure 3.6a).

Viceversa, se lo scarto dalla X è positivo (sopra la media) e dalla Y è negativo (sotto la media), il loro prodotto sarà negativo. Similmente, se lo scarto dalla X è negativo (sotto la media) e dalla Y è positivo (sopra la media), il loro prodotto sarà negativo. I punti dello scatter plot si troveranno (*per la maggior parte*) nel secondo e quarto quadrante (Figure 3.6b).

La covarianza può assumere **tutti i valori reali**. Il suo valore dipende dall'unità di misura delle variabili → ci dice la **direzione** della relazione ma non la sua **intensità**. L'unica cosa che si può concludere quindi è se esiste una relazione positiva o negativa, e, al massimo, se c'è *assenza di relazione lineare*.



(a) Cov = 0

(b) Cov = 0

(c) Cov > 0

(d) Cov < 0

Figure 3.7: Esempi di relazione Covarianza

3.4.3.1 Calcolo della covarianza

Età (in mesi) e peso (in kg) di un bambino rilevati in 6 momenti

	eta	peso
1	3	6
2	6	8
3	9	9
4	12	10
5	24	13
6	36	20

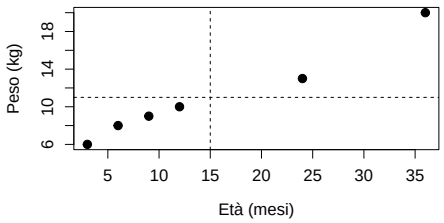


Figure 3.8: Grafico a dispersione

Dal grafico, si vede chiaramente che al crescere del peso, si osserva un aumento dell'età e che all'aumentare dell'età si osserva una crescita del peso (sono valide entrambe le interpretazioni, osserviamo covariazione non causalità).

La covarianza si può calcolare a mano (nel senso con codice R) data la formula in Equation 3.1 (vedi Listing 3.1).

	x	y	x - media(x)	y - media(y)	prodotto
1	3	6	-12	-5	60
2	6	8	-9	-3	27
3	9	9	-6	-2	12

Listing 3.1 Codice per ottenere gli elementi di base del calcolo della covarianza.

```
calcolo <- data.frame(x = bimbo$eta, y = bimbo$peso,
                      sx = bimbo$eta - mean(bimbo$eta),
                      sy = bimbo$peso - mean(bimbo$peso))
calcolo$prodotto <- calcolo$sx * calcolo$sy
colnames(calcolo) <- c("x", "y", "x - media(x)", "y - media(y)", "prodotto")
calcolo
```

```
4 12 10           -3           -1        3
5 24 13            9            2       18
6 36 20           21            9      189
```

$\bar{x} = 15$, $\bar{y} = 11$, $\sum_i (x_i - \bar{x})(y_i - \bar{y}) = 309$

$$\text{Cov}(X, Y) = \frac{309}{6 - 1} = 61.8$$

3.4.3.2 La funzione `cov()`

`cov()`: Restituisce la covarianza campionaria (Equation 3.1) tra due variabili quantitative appaiate (X, Y) aventi lo stesso numero di osservazioni

```
X <- c(9,8,12,9,10,14,5,4,13,7,5)
Y <- c(9,6,8,9,7,10,8,4,11,5,4)
Z <- -Y
cov(X,Y)
```

```
[1] 6.309091
```

```
cov(X,Z)
```

```
[1] -6.309091
```

```
cov(bimbo$eta, bimbo$peso)
```

```
[1] 61.8
```

3.4.4 Correlazione campionaria

Per stabilire l'**intensità** della relazione la covarianza va *normalizzata*, ovvero divisa per il suo valore massimo, ovvero il prodotto tra le deviazioni standard delle due variabili.

$$-sd(X)\,sd(Y) \leq \text{Cov}(X,Y) \leq +sd(X)\,sd(Y),$$

per cui il coefficiente di correlazione di Bravais-Pearson r si ottiene con:

$$r = \frac{\text{Cov}(X,Y)}{sd(X)\,sd(Y)}. \tag{3.2}$$

Si ottiene quindi un indice che ha un range di variabilità molto bene definito:

$$-1 \leq r \leq 1,$$

dove $r = +1$ indica un legame **perfettamente lineare e diretto**, $r = -1$ indica un legame **perfettamente lineare e inverso** e $r = 0$ indica **incorrelazione**, ovvero la mancanza di una relazione *lineare*. Tuttavia le variabili potrebbero avere qualche altra forma di relazione (quadratica, esponenziale, logaritmica ecc.).

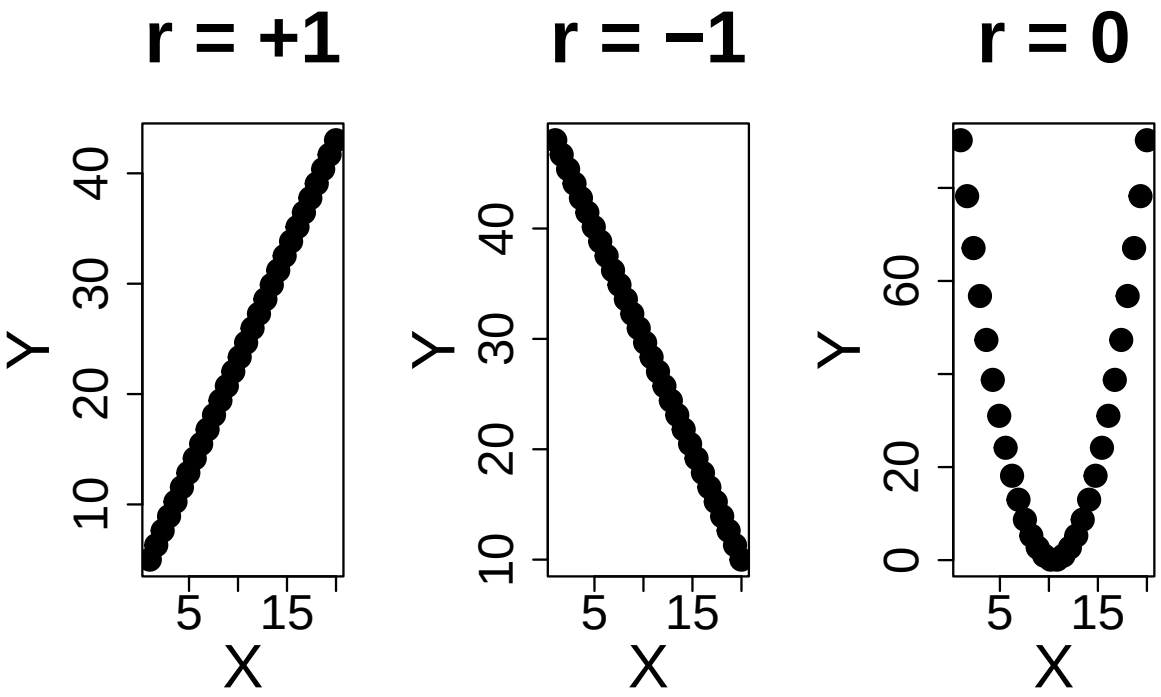


Figure 3.8: Esemplificazioni dei valori limite del coefficiente di correlazione.

3.4.4.1 La funzione `cor()`

`cor()`: Restituisce la correlazione (lineare) campionaria (Equation 3.2) tra due variabili quantitative appaiate (X, Y) aventi lo stesso numero di osservazioni:

```
X <- c(9,8,12,9,10,14,5,4,13,7,5)
Y <- c(9,6,8,9,7,10,8,4,11,5,4)
Z <- -Y
cor(X,Y)
```

```
[1] 0.7921487
```

```
cor(X,Z)
```

```
[1] -0.7921487
```

Riprendendo l'esempio del peso e dell'età del bambino:

```
cov(bimbo$eta, bimbo$peso) / (sd(bimbo$eta) * sd(bimbo$peso))
```

```
[1] 0.9860185
```

```
cor(bimbo$eta, bimbo$peso)
```

```
[1] 0.9860185
```

$$r = \frac{61.8}{12.59 \cdot 4.98} = 0.986$$

Le variabili presentano una **fortissima correlazione positiva**: i punti tendono a disporsi lungo una retta crescente.

Dove si può trovare r

- Attendibilità test-retest
- Forme parallele di un test (coefficiente di equivalenza)
- Accordo tra le valutazioni di due giudici
- Consistenza interna di un questionario
- ...

Statistics doesn't lie, but liars use statistics

Due variabili possono essere fortemente correlate senza che esista alcun legame tra di loro

https://www.tylervigen.com/spurious-correlations

Correlazione $\neq$ causalità

L'ignoranza afferma o nega rotondamente; la scienza dubita (Voltaire)

3.4.5 Tipi di relazione

ATTENZIONE

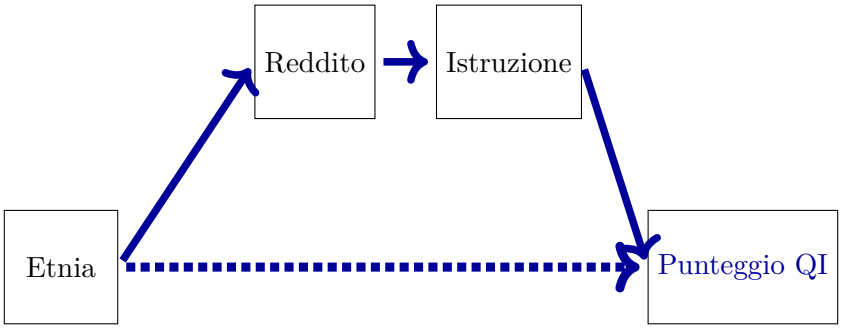
Le diverse forme di relazione (diretta, indiretta, spuria) fanno riferimento a ipotesi sui *meccanismi causali* che legano le variabili. Gli indici di associazione visti non permettono di distinguere tra queste forme: una correlazione elevata può essere il risultato di una relazione diretta, indiretta o spuria. Stabilire che X è causa di Y richiede, oltre alla covariazione tra le due variabili, che X preceda Y nel tempo e che si possano escludere spiegazioni alternative, condizione che in genere può essere soddisfatta solo attraverso un disegno sperimentale.

3.4.5.1 Relazione diretta



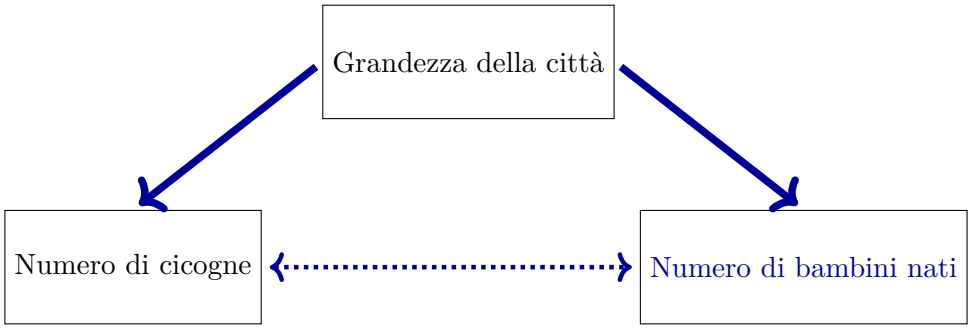
Si parla di **relazione diretta** tra due variabili X e Y quando X influenza Y senza che intervengano altre variabili: la variazione di X si riflette direttamente in una variazione di Y ($X \rightarrow Y$). L'associazione osservata tra le due variabili è quindi dovuta interamente al legame che le unisce, e non alla presenza di una terza variabile che agisce su entrambe o che fa da tramite tra l'una e l'altra.

3.4.5.2 Relazione indiretta



Si parla di **relazione indiretta** tra due variabili X e Y quando X influenza Y non in modo diretto, ma attraverso una o più variabili intermedie, chiamate **mediatori** (Z): $X \to Z \to Y$. Una variazione di X produce una variazione di Z, che a sua volta produce una variazione di Y. L'associazione osservata tra X e Y è quindi reale, ma il legame tra le due variabili *passa attraverso* il mediatore. Se si tiene costante il mediatore (ad esempio confrontando solo unità con lo stesso valore di Z), l'associazione tra X e Y scompare (mediazione totale) o si riduce (mediazione parziale, quando oltre al percorso indiretto esiste anche una componente diretta $X \to Y$).

3.4.5.3 Relazione spuria



Si parla di **relazione spuria** tra due variabili X e Y quando tra le due si osserva un'associazione, ma questa non è dovuta a un legame causale tra X e Y: entrambe dipendono da una terza variabile Z, detta **variabile confondente** (o causa comune), che le fa variare insieme ($X \leftarrow Z \to Y$). L'associazione tra X e Y è quindi solo apparente: se si tiene costante Z (ad esempio confrontando solo unità con lo stesso valore di Z), l'associazione tra X e Y scompare.

3.5 Previsione variabili quantitative

3.5.1 Regressione lineare

La correlazione permette di stabilire quanto covariano due variabili, x e y: **modello simmetrico** per determinare *l'intensità della relazione*.

Con un modello di regressione si vuole andare più a fondo per capire **quanto** varia y in funzione di ogni cambiamento in x: stabilire non solo se le variabili sono associate o meno ma la portata della variazione.

La relazione è *asimmetrica*: si vuole spiegare y in funzione di x (questo non vuol dire in alcun modo che la relazione sia causale).

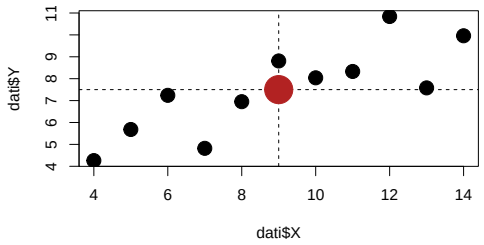


Figure 3.9: Scatterplot X e Y

	X	Y
1	10	8.04
2	8	6.95
3	13	7.58
4	9	8.81
5	11	8.33
6	14	9.96
7	6	7.24
8	4	4.26
9	12	10.84
10	7	4.82
11	5	5.68

La Figure 3.9 permette di capire che la relazione tra le due variabili è positiva e, calcolando il coefficiente di correlazione, si ottiene $r = 0.82$, a indicare una forte relazione tra le due variabili. Tuttavia, in base a queste informazioni, non è possibile dire quanto cambia y in funzione di ogni cambiamento in x, né è possibile fare una predizione su quale valore di y si può osservare dato un valore di x. Per poterlo fare, è necessario usare un modello statistico idoneo che, in questo caso, prevede la costruzione di una retta che passi attraverso la nuvola di punti (Figure 3.10). Si tratta quindi di trovare, tra le infinite rette che possono passare per un piano cartesiano, quella che permette di predire *sufficientemente bene* tutti i punti (ovvero le osservazioni), ovvero quella retta che permetta di *minimizzare le distanze delle osservazioni*.

La regressione lineare tra una *variabile indipendente* X (predittore) ed una *variabile dipendente* Y (criterio) calcola la migliore approssimazione lineare che identifica il legame tra le due variabili utilizzando la struttura di dati bivariata (x_i, y_i) osservata su un campione di n osservazioni.

$$x = (x_1, x_2, \dots, x_n)$$
$$y = (y_1, y_2, \dots, y_n)$$
$$d = ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$$

$$Y = \beta_0 + \beta X \tag{3.3}$$

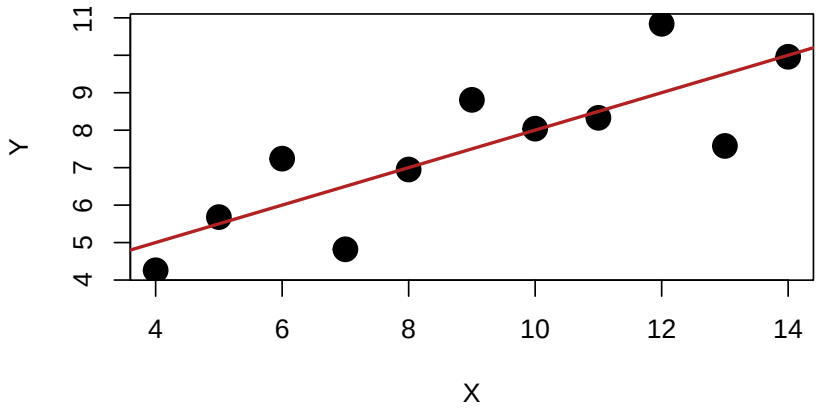


Figure 3.10: Una retta che passi attraverso le coppie di osservazioni

$$y_i = \beta_0 + \beta x_i + \varepsilon_i \quad (i = 1, \dots, n) \tag{3.4}$$

La Equation 3.3 descrive la retta di regressione, cioè la relazione lineare tra X e Y:

La Equation 3.4 descrive quindi ogni osservazione i come la somma di una parte spiegata dalla retta, $\beta_0 + \beta x_i$, e di un errore ε_i, che rappresenta la distanza tra il valore osservato y_i e la retta.

Dopo aver stimato β_0 e β dai dati secondo il *principio dei minimi quadrati* (vedi Section 3.5.2), si può predire il valore di y per un dato valore di x:

$$y_i^* = \hat{\beta}_0 + \hat{\beta} x_i \tag{3.5}$$

Nella Equation 3.5 l'errore non compare, perché la predizione y_i^* è il punto sulla retta stimata.

Significato dei coefficienti

y_i^*: Valore di y atteso per l'i-esima osservazione
β_0: Valore dell'intercetta, ovvero *valore atteso in y quando $x_i = 0$*
β: Coefficiente angolare ovvero la pendenza della retta. Definisce *il cambiamento medio atteso in y per ogni cambiamento unitario in x*
ε_i: Componente d'errore, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$

3.5.2 Principio dei minimi quadrati

La retta di regressione viene tracciata seguendo il principio dei minimi quadrati, secondo cui si cercano i valori di β_0 e β che rendono minima la somma delle distanze al quadrato dei punti dalla retta (Equation 3.6, somma dei quadrati degli errori):

$$SQE(\beta_0, \beta) = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta x_i))^2 \tag{3.6}$$

In corrispondenza dei valori che la minimizzano, $\hat{\beta}_0$ e $\hat{\beta}$, si ottengono i valori attesi $y_i^* = \hat{\beta}_0 + \hat{\beta} x_i$ e i residui $e_i = y_i - y_i^*$, per cui il valore minimo dell'SQE è $\sum_{i=1}^{n} (y_i - y_i^*)^2 = \sum_{i=1}^{n} e_i^2$.

Si può dimostrare che i valori che minimizzano la Equation 3.6 sono, per il coefficiente angolare β,

$$\hat{\beta} = \frac{\text{Cov}(X, Y)}{sd(X)^2} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{3.7}$$

e, per l'intercetta β_0,

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}\, \bar{x} \tag{3.8}$$

A partire dai dati osservati, si ottengono le stime $\hat{\beta}_0$ e $\hat{\beta}$, che sostituite nella Equation 3.3 definiscono la retta di regressione stimata, $y^* = \hat{\beta}_0 + \hat{\beta} x$. È la retta che passa più vicina possibile a tutti i punti e che si usa per fare predizioni (Equation 3.5).

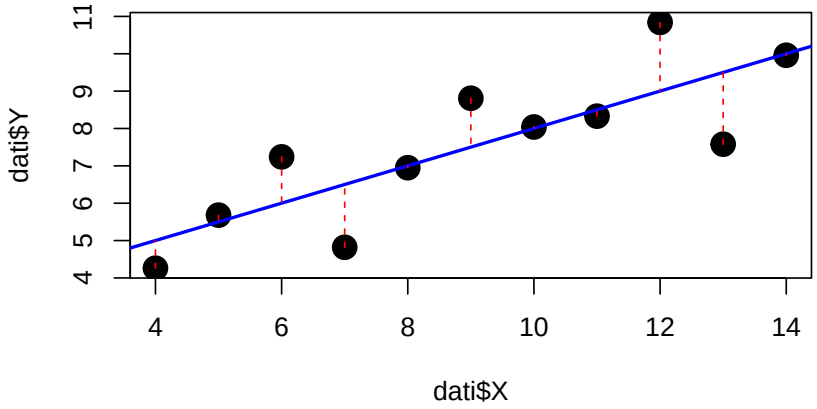


Figure 3.11: Retta di regressione stimata con il principio dei minimi quadrati

3.5.3 Proprietà della retta di regressione

3.5.3.1 Coefficiente angolare e correlazione

Nella regressione **con un solo predittore**, il coefficiente angolare e il coefficiente di correlazione sono in stretta relazione tra loro, tanto che, conoscendo le deviazioni standard di X e Y, si può ottenere l'uno a partire dall'altro:

$$\hat{\beta} = \frac{\text{Cov}(X,Y)}{sd(X)^2} \qquad\qquad r = \frac{\text{Cov}(X,Y)}{sd(X)\,sd(Y)}$$

Moltiplicando e dividendo $\hat{\beta}$ per $sd(Y)$ si ottiene:

$$\hat{\beta} = \frac{\text{Cov}(X,Y)}{sd(X)\,sd(X)} \cdot \frac{sd(Y)}{sd(Y)} = \underbrace{\frac{\text{Cov}(X,Y)}{sd(X)\,sd(Y)}}_{r} \cdot \frac{sd(Y)}{sd(X)} = r\,\frac{sd(Y)}{sd(X)} \tag{3.9}$$

Se X e Y sono standardizzate, cioè $sd(X) = sd(Y) = 1$, il coefficiente angolare coincide con il coefficiente di correlazione. Nella regressione con più predittori questa relazione non vale più, perché ogni coefficiente dipende anche dalle correlazioni tra i predittori.

Dalla Equation 3.9 e dalla Equation 3.8 seguono due proprietà:

1. Il **segno** del coefficiente angolare è lo stesso di r, perché le deviazioni standard sono sempre positive: $r > 0 \Rightarrow$ retta crescente, $r < 0 \Rightarrow$ retta decrescente, $r = 0 \Rightarrow$ retta orizzontale.
2. La retta passa sempre per il **punto delle medie** $(\bar{x}, \bar{y})$: sostituendo $x = \bar{x}$ in $y^* = \hat{\beta}_0 + \hat{\beta}x$ e usando $\hat{\beta}_0 = \bar{y} - \hat{\beta}\,\bar{x}$, si ottiene $y^* = \bar{y} - \hat{\beta}\,\bar{x} + \hat{\beta}\,\bar{x} = \bar{y}$.

Esempio

```
b <- cov(bimbo$eta, bimbo$peso) / var(bimbo$eta)
b0 <- mean(bimbo$peso) - b * mean(bimbo$eta)
c(b0, b)
```

```
[1] 5.1477273 0.3901515
```

Retta di regressione: $y^* = 5.15 + 0.39x$
Quale sarebbe il peso previsto a 48 mesi? $y^*_{48} = 5.15 + 0.39 \cdot 48 = 23.88$ kg

3.5.4 Indice di adattamento R^2

Il principio dei minimi quadrati, con cui viene stimata la retta di regressione, garantisce che la retta ottenuta sia la migliore tra le infinite rette che si possono tracciare tra i punti, nel senso che è quella che rende minima la somma dei quadrati degli errori (Equation 3.6). Tuttavia, non dice nulla sulla bontà del modello, cioè sulla sua capacità di spiegare la relazione tra le variabili. In altre parole, non dice quanta parte della varianza di Y è effettivamente spiegata dal modello e quindi, dato che c'è un solo predittore, quanta varianza di Y è spiegata da X.

Per poter parlare di varianza spiegata e bontà del modello, bisogna ricorrere alla statistica R^2 (Equation 3.10) che quantifica la proporzione di **varianza di regressione** (ovvero la

proporzione di varianza spiegata dal modello) rispetto alla varianza totale, ovvero la varianza d'errore sommata alla varianza di regressione:

$$R^2 = 1 - \frac{SQE/(n-1)}{\text{Var}(Y)} = 1 - \frac{\sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{3.10}$$

dove $\frac{SQE}{n-1}$ è la varianza dei residui, ovvero degli scarti del modello. Come stima della varianza della componente d'errore σ^2 si usa invece $\hat{\sigma}_\varepsilon^2 = \frac{SQE}{n-2}$, perché dai dati sono stati stimati due parametri (β_0 e β).

$0 \leq R^2 \leq 1$: Più R^2 è vicino ad 1, migliore è il modello.

Nella regressione con un solo predittore, R^2 è uguale al quadrato del coefficiente di correlazione tra X e Y: $R^2 = r^2$, con cui ha una relazione come riportato in Figure 3.12.

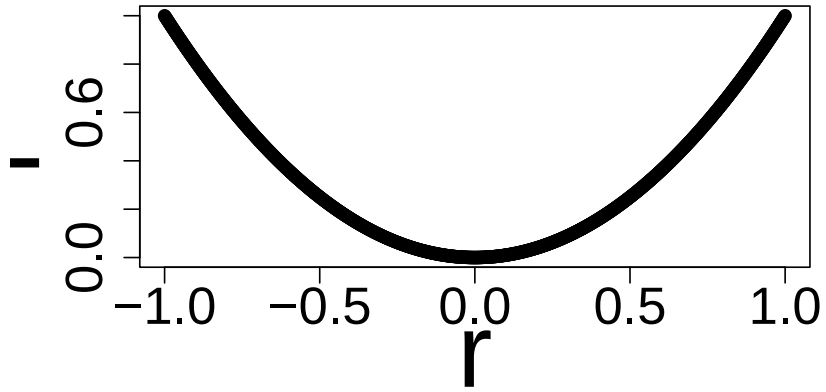


Figure 3.12: Relazione quadratica tra r e R^2.

3.5.4.1 Esempi di relazione tra r e R^2

- In Figure 3.13a la relazione è positiva ma moderata ($r = 0.59$): i punti sono piuttosto dispersi intorno alla retta e i residui sono grandi, quindi X spiega solo una parte della varianza di Y ($R^2 = 0.35$).
- In Figure 3.13b la relazione è negativa e forte ($r = -0.9$): i punti sono vicini alla retta e il modello spiega gran parte della varianza di Y ($R^2 = 0.81$).
- In Figure 3.13c la relazione è positiva e molto forte ($r = 0.98$): i punti sono quasi allineati, i residui sono piccolissimi e $R^2 = 0.96$ è vicino a 1.
- In Figure 3.13d la relazione è praticamente assente ($r = -0.14$): la retta è quasi orizzontale e, pur essendo la migliore secondo i minimi quadrati, spiega pochissima varianza di Y ($R^2 = 0.02$).

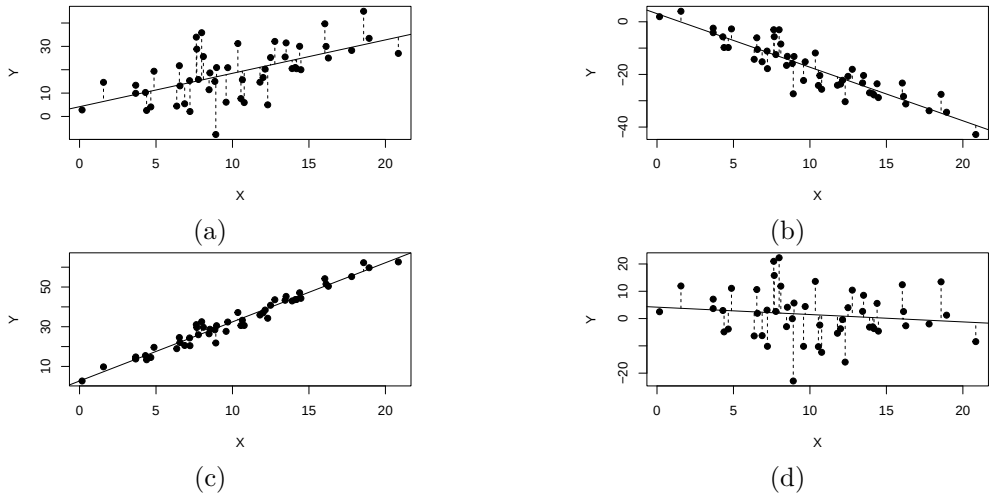


Figure 3.13: Retta di regressione e residui (linee tratteggiate) in quattro insiemi di dati simulati.

3.5.5 La funzione: `lm()`

I parametri del modello in R si stimano con la funzione `lm()` (**l**inear **m**odel):

`lm(Y ~ X)`: Restituisce la stima dei parametri del modello di regressione lineare (secondo il principio dei minimi quadrati) dove si predicono i valori di Y a seconda dei valori di X.

`~`: (Per i PC con tastierino numerico): `ALT + 126`, Per i MAC: `ALT + 5`. Altrimenti cercate "tilde" su google e copiatela e incollatela.

Come ottenere tilde (`~`) su Windows11 e 10

Una soluzione è aggiungere il layout **Stati Uniti** come seconda tastiera e passare da una all'altra quando serve.
Windows 11

1. Aprite **Impostazioni** (`Win + I`) e andate in **Data/ora e lingua** → **Lingua e area geografica**.
2. Accanto a **Italiano**, cliccate sui tre puntini **...** e scegliete **Opzioni lingua**.
3. Nella sezione **Tastiere**, cliccate su **Aggiungi una tastiera** e selezionate **Stati Uniti** (QWERTY).

Windows 10

1. Aprite **Impostazioni** → **Ora e lingua** → **Lingua**.
2. Cliccate su **Italiano** e poi su **Opzioni**.
3. In **Tastiere**, cliccate su **Aggiungi una tastiera** e selezionate **Stati Uniti** (QWERTY).

Come si passa da una tastiera all'altra

Premete `Win + Spazio`, oppure cliccate sulla sigla **ITA/ENG** nella barra delle applicazioni, vicino all'orologio.
Come scrivere la tilde (~)

- Con la tastiera **Stati Uniti**: `Shift` + il tasto in alto a sinistra, sotto `Esc` e accanto all'`1` (lo stesso tasto del backtick `` ` ``).
- Con la tastiera **italiana**: tenete premuto `Alt` e digitate `126` sul tastierino numerico. Su molti computer funziona anche `AltGr + ì`.

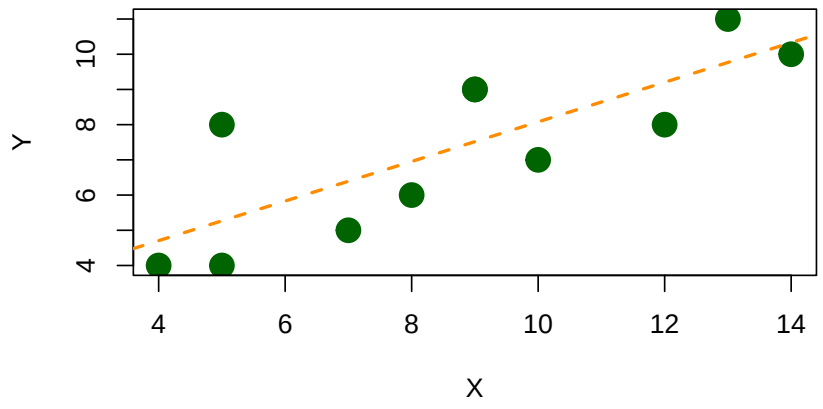
```
X <- c(9,8,12,9,10,14,5,4,13,7,5)
Y <- c(9,6,8,9,7,10,8,4,11,5,4)
lm(Y~X) # stima il modello
```

```

Call:
lm(formula = Y ~ X)

Coefficients:
(Intercept)            X
     2.4554       0.5624
```

```
plot(X, Y, pch = 19, col = "darkgreen", cex = 2) # disegna il grafico
abline(lm(Y ~ X), lty = 2, lwd = 2, col = "darkorange") # aggiunge la retta
```



3.5.6 Assunti del modello di regressione lineare

- **Indipendenza dei predittori dall'errore**

  I predittori sono misurati senza errore, ovvero i predittori e gli errori di misura non sono correlati nella popolazione (per questo è importante il campionamento casuale)

- **Indipendenza delle osservazioni**

  $\forall i \neq j$, $\varepsilon_i \perp\!\!\!\perp \varepsilon_j$ (le osservazioni sono state campionate indipendentemente l'una dall'altra)

- **Linearità**

  Il valore atteso dell'errore per un dato valore di x è 0

- **Normalità**

  Gli errori si distribuiscono normalmente con media 0 e varianza σ^2, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Dato che, fissato x_i, la quantità $\beta_0 + \beta x_i$ è costante, ne segue che anche y_i, per un dato valore di x_i, si distribuisce normalmente con media $\beta_0 + \beta x_i$ e varianza σ^2: per ogni valore di x, i valori di y si distribuiscono normalmente attorno alla retta. Questo non significa che la variabile Y nel suo complesso debba essere normale: l'assunzione riguarda gli errori, ed è per questo che si verifica sui residui.

- **Varianza costante** (omoschedasticità)

  La varianza degli errori è costante per qualunque valore di X

Gli errori ε_i della popolazione non si osservano. Per verificare le assunzioni si usano quindi i loro "rappresentanti" nel campione, i **residui** $e_i = y_i - y_i^*$, cioè le differenze tra i valori osservati e i valori attesi in base alla retta stimata.

Per mostrare come si verificano le assunzioni, si prende ad esempio la relazione tra il voto a un esame (`voto`) e le ore di studio (`ore`):

```
head(studio)
```

```
   ore voto
1 18.3 26.4
2 18.7 25.5
3  5.7 19.9
4 16.6 23.5
5 12.8 19.9
6 10.4 22.1
```

```
modello <- lm(voto ~ ore, data = studio)
modello
```

```

Call:
lm(formula = voto ~ ore, data = studio)

Coefficients:
(Intercept)          ore
    14.8678       0.6243
```

I valori attesi y_i^* si ottengono con la funzione `fitted()` e i residui con la funzione `residuals()`:

```
studio$attesi <- fitted(modello)      # valori attesi y*
studio$residui <- residuals(modello)  # residui y - y*
head(studio)
```

```
   ore voto   attesi    residui
1 18.3 26.4 26.29279  0.1072084
2 18.7 25.5 26.54252 -1.0425177
3  5.7 19.9 18.42642  1.4735784
4 16.6 23.5 25.23146 -1.7314560
5 12.8 19.9 22.85906 -2.9590587
6 10.4 22.1 21.36070  0.7392975
```

I residui sono semplicemente la differenza tra i valori osservati e i valori attesi:

```
head(studio$voto - studio$attesi)
```

```
[1]  0.1072084 -1.0425177  1.4735784 -1.7314560 -2.9590587  0.7392975
```

Indipendenza dei predittori dall'errore

I predittori sono misurati senza errore, cioè i predittori e gli errori non sono correlati nella popolazione. Per questo è importante il campionamento casuale.

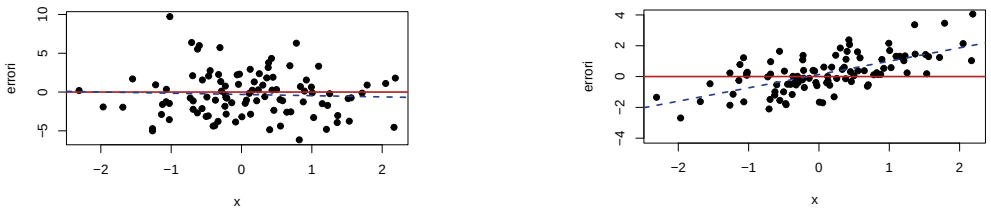
Questa assunzione **non si può verificare con i residui**. Per come è costruita la retta dei minimi quadrati, i residui sono sempre non correlati con il predittore, anche quando l'assunzione è violata:

```
cor(studio$residui, studio$ore)
```

```
[1] -8.515291e-18
```

L'assunzione va quindi garantita a monte, con il disegno dello studio: campionando i partecipanti in modo casuale e usando misure affidabili del predittore.

In una simulazione gli errori veri sono noti e si può vedere che cosa significa l'assunzione (Figure 3.14). Se è rispettata (Figure 3.14a), gli errori non hanno alcuna relazione con x: la retta che descrive la relazione tra errori e x (tratteggiata, in blu) coincide quasi con la retta orizzontale in 0 (continua, in rosso). Se è violata (Figure 3.14b), gli errori crescono al crescere di x e la retta tratteggiata si allontana da quella orizzontale. Con dati reali, però, gli errori veri non si conoscono: questo grafico si può costruire solo in una simulazione e serve a capire il significato dell'assunzione, non a verificarla.



(a) (b)

Figure 3.14: Relazione tra il predittore x e gli errori in due insiemi di dati simulati.

Indipendenza delle osservazioni

Gli errori di osservazioni diverse sono indipendenti tra loro, $\forall i \neq j$, $\varepsilon_i \perp\!\!\!\perp \varepsilon_j$, cioè le osservazioni sono state campionate indipendentemente l'una dall'altra.

Anche questa assunzione dipende soprattutto da **come sono stati raccolti i dati**. È violata, per esempio, quando ci sono misure ripetute sulla stessa persona, studenti della stessa classe o dati raccolti nel tempo.

Se i dati hanno un ordine, per esempio l'ordine di raccolta, si possono guardare i residui in quell'ordine (Figure 3.15). Se le osservazioni sono indipendenti, i residui oscillano attorno allo 0 senza alcun andamento.

```
plot(studio$residui, type = "b", pch = 19,
     xlab = "ordine di raccolta", ylab = "residui")
abline(h = 0, col = "firebrick", lwd = 2)
```

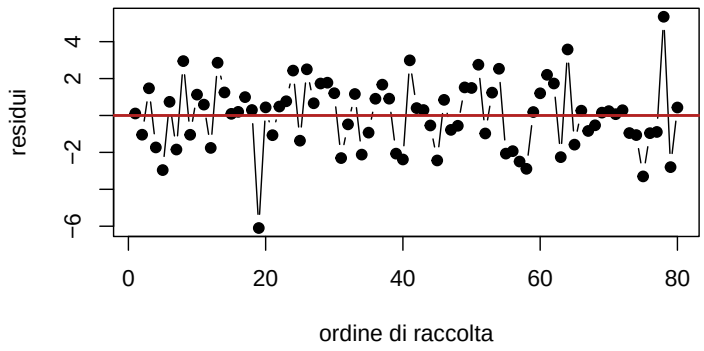


Figure 3.15: Residui del modello nell'ordine di raccolta dei dati.

Linearità

Il valore atteso dell'errore per un dato valore di x è 0, cioè la relazione tra X e Y è descritta bene da una retta.

Per verificarla si rappresentano i residui in funzione del predittore. Se la relazione è lineare, i residui si distribuiscono **casualmente** attorno allo 0, senza alcun andamento, come nella Figure 3.16.

```r
plot(studio$ore, studio$residui, pch = 19,
     xlab = "ore", ylab = "residui")
abline(h = 0, col = "firebrick", lwd = 2)
```

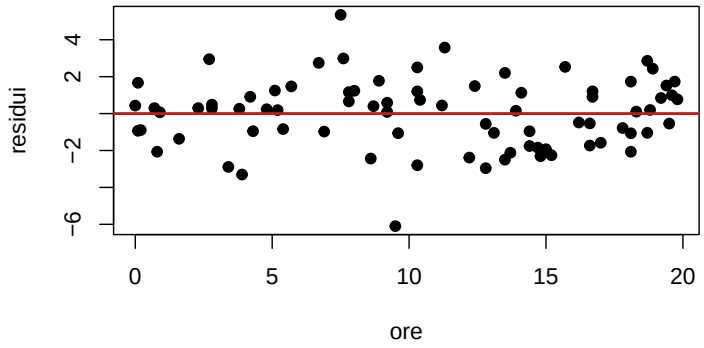
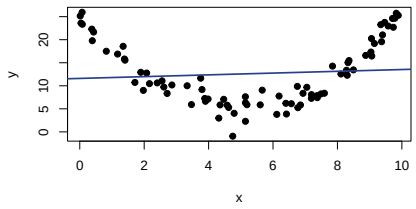
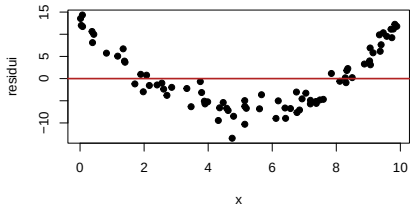


Figure 3.16: Residui in funzione del predittore nell'esempio delle ore di studio.

Nella Figure 3.17, invece, la relazione tra x e y non è lineare (Figure 3.17a). I residui hanno una chiara forma a "U" (Figure 3.17b): il modello lineare sovrastima i valori al centro e li sottostima agli estremi, quindi l'assunzione di linearità è violata.



(a) Dati e retta di regressione



(b) Residui in funzione di x

Figure 3.17: Esempio di violazione dell'assunzione di linearità.

Normalità degli errori

Gli errori si distribuiscono normalmente con media 0 e varianza σ^2, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Di conseguenza, anche y_i è distribuita normalmente, con media $\beta_0 + \beta x_i$.

La media dei residui è sempre 0, per come è costruita la retta dei minimi quadrati, quindi da sola non dice nulla sulla normalità:

```r
mean(studio$residui)
```

```
[1] -2.791617e-17
```

Per verificare la normalità si possono usare tre strumenti.

1. **Gli indici descrittivi dei residui.** Se i residui sono normali, si distribuiscono in modo simmetrico attorno allo 0: la mediana deve essere il più vicina possibile a 0, e il minimo e il massimo (così come il primo e il terzo quartile) devono essere simili in valore assoluto.

```
summary(studio$residui)
```

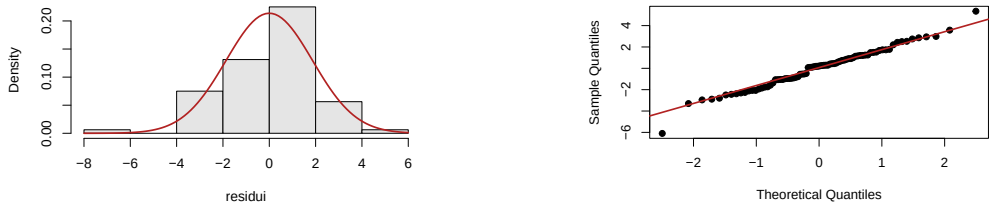
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-6.0988 -1.0629  0.2153  0.0000  1.2028  5.3498
```

2. **L'istogramma dei residui** (Figure 3.18a), a cui si sovrappone la distribuzione normale con media 0 e deviazione standard pari a quella dei residui (in rosso). Se i residui sono normali, l'istogramma segue la curva.

3. **Il Q-Q plot dei residui** (Figure 3.18b), che confronta i quantili dei residui con quelli di una distribuzione normale. Se i residui sono normali, i punti stanno **sulla retta**.

```
hist(studio$residui, freq = FALSE, main = "",
     xlab = "residui", col = "gray90")
curve(dnorm(x, mean = 0, sd = sd(studio$residui)),
      add = TRUE, col = "firebrick", lwd = 2)

qqnorm(studio$residui, pch = 19, main = "")
qqline(studio$residui, col = "firebrick", lwd = 2)
```



(a) Istogramma (b) Q-Q plot

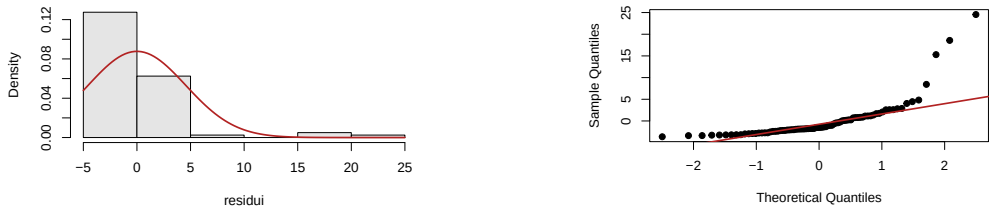
Figure 3.18: Verifica della normalità dei residui nell'esempio delle ore di studio.

Nella Figure 3.19, invece, i residui hanno una distribuzione asimmetrica, con una coda lunga a destra: l'istogramma non segue la curva normale (Figure 3.19a) e nel Q-Q plot i punti si allontanano dalla retta agli estremi (Figure 3.19b).

Varianza costante (omoschedasticità)

La varianza degli errori è costante per qualunque valore di X.

Per verificarla si rappresentano i residui in funzione dei valori attesi. La variabilità dei residui deve essere **costante** per tutti i valori attesi, cioè la "nuvola" di punti deve avere la stessa ampiezza da sinistra a destra, come nella Figure 3.20.



(a) Istogramma (b) Q-Q plot

Figure 3.19: Esempio di violazione dell'assunzione di normalità degli errori.

```
plot(studio$attesi, studio$residui, pch = 19,
     xlab = "valori attesi", ylab = "residui")
abline(h = 0, col = "firebrick", lwd = 2)
```

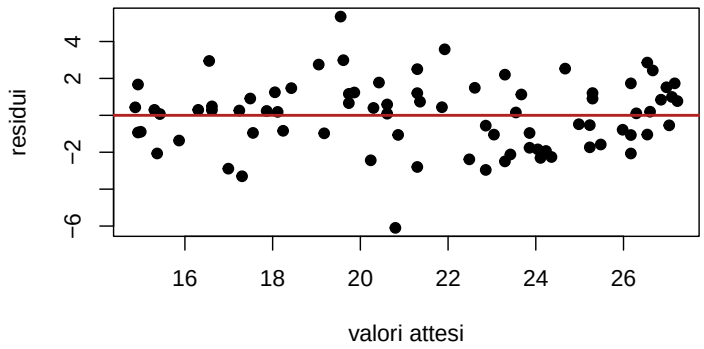
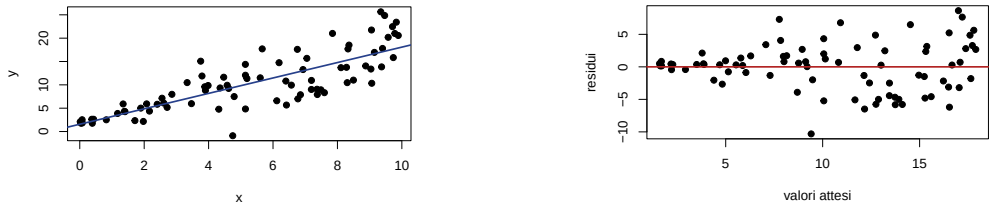


Figure 3.20: Residui in funzione dei valori attesi nell'esempio delle ore di studio.

Nella Figure 3.21, invece, i punti si allargano al crescere di x (Figure 3.21a) e i residui hanno una forma "a imbuto" (Figure 3.21b): sono sempre più dispersi al crescere dei valori attesi. In questo caso si parla di eteroschedasticità.



(a) Dati e retta di regressione (b) Residui in funzione dei valori attesi

Figure 3.21: Esempio di violazione dell'assunzione di omoschedasticità.

3.5.7 Cosa guardare nei residui

Assunzione	Come si verifica	Codice R	Cosa si deve vedere
Indipendenza dei predittori dall'errore	Non si verifica con i residui: si garantisce con il campionamento casuale e con misure affidabili	—	—
Indipendenza delle osservazioni	In base a come sono stati raccolti i dati; residui nell'ordine di raccolta	`plot(res, type = "b")`	Nessun andamento nell'ordine
Linearità	Residui in funzione del predittore	`plot(x, res)`	Nuvola casuale attorno a 0
Normalità degli errori	Indici descrittivi, istogramma e Q-Q plot dei residui	`summary(res)`, `hist(res)`, `qqnorm(res)`	Distribuzione simmetrica e a campana, punti sulla retta
Omoschedasticità	Residui in funzione dei valori attesi	`plot(attesi, res)`	Ampiezza costante della nuvola (niente imbuti)

`res` nella tabella è l'oggetto R che contiene i residui.

Tutto parte da due funzioni:

```
attesi <- fitted(modello)
residui <- residuals(modello)
```

3.6 Materiale supplementare

3.6.1 Outliers e casi influenti

Outlier: Il valore di un'osservazione nella variabile dipendente è anomalo dato il valore del predittore (ha un residuo molto grande)

Leverage (leva): Quanto il valore di un'osservazione nel predittore è lontano dalla media del predittore → quanto *potenzialmente* può spostare la retta

Influential case: Una singola osservazione che riesce a spostare l'intera retta di regressione

Un'osservazione può essere:

- Un outlier (ma non un influential case)
- Un punto ad alta leva in linea con gli altri (ma non un influential case)
- Un outlier influente (anomalo **e** ad alta leva)

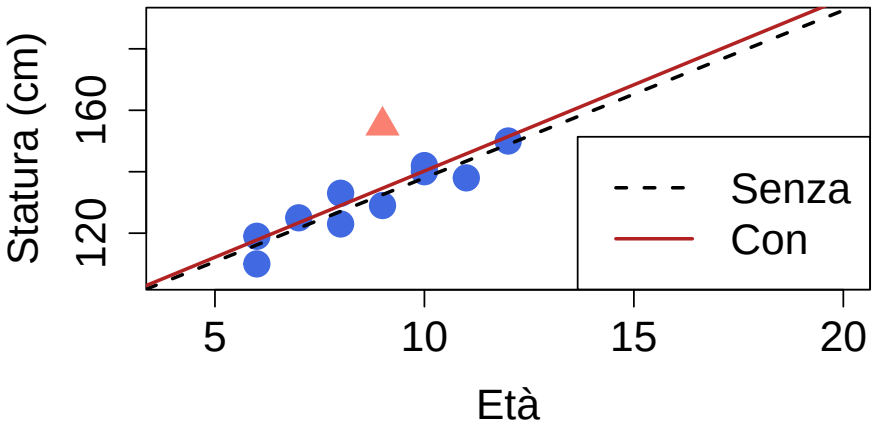


Figure 3.22: Un outlier (triangolo rosa)

Il triangolo rosa in Figure 3.22 è un outlier: è valore anomalo rispetto alla nuvola di punti, ma essendo vicino alla media dell'età sposta la retta di poco (soprattutto in alto, l'intercetta).

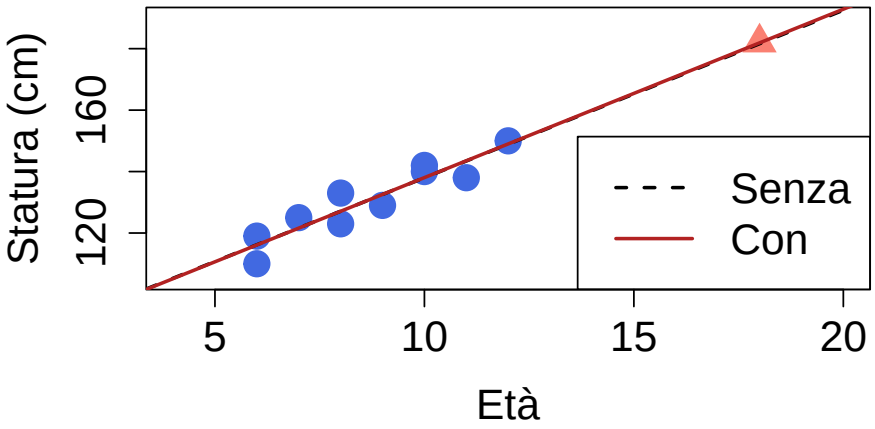


Figure 3.23: Una leva (triangolo rosa)

Il triangolo rosa in Figure 3.23 è lontano dagli altri sull'asse delle x (alta leva) e, di conseguenza, anche sull'asse delle y. Considerando solo la statura sarebbe un valore estremo, ma in regressione conta la y *dato* il valore di x: il punto è in linea con l'andamento dei dati (residuo quasi nullo), quindi non è un outlier e la retta non cambia.

Il triangolo rosa in Figure 3.24 è anomalo **e** lontano dagli altri sull'asse delle x: ne sposta **da solo** drasticamente la pendenza.

3.6.1.1 Leverage e distanza di Cook

Leverage (h_i). Misura quanto il valore x_i di un'osservazione è lontano dalla media del predittore. Nella regressione semplice si calcola come:

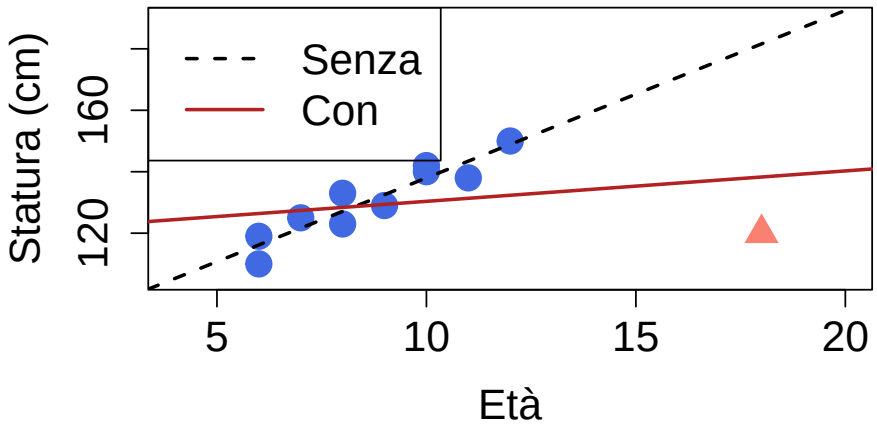


Figure 3.24: Un outlier influential case (triangolo rosa)

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

- Dipende **solo da** x: la y non entra nel calcolo. Per questo il leverage indica quanto un punto *potrebbe* spostare la retta, non quanto la sposta davvero.
- Varia tra $\frac{1}{n}$ (quando $x_i = \bar{x}$) e 1. La media dei leverage è $\frac{2}{n}$, dove 2 è il numero di parametri stimati ($\hat{\beta}_0$ e $\hat{\beta}$).
- Regola pratica: un'osservazione ha un leverage alto se $h_i > 2 \times \frac{2}{n}$, cioè se supera il doppio del leverage medio.

Distanza di Cook (D_i). Misura quanto cambiano *tutti* i valori attesi se si toglie l'osservazione i e si ristima il modello:

$$D_i = \frac{\sum_{j=1}^{n}\left(y_j^* - y_{j(-i)}^*\right)^2}{2\,s^2}$$

dove $y_{j(-i)}^*$ è il valore atteso per l'osservazione j stimato *senza* l'osservazione i, e $s^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-2}$ è la varianza dei residui.

Combina residuo e leverage: un'osservazione è influente solo se è **sia** anomala (residuo grande) **sia** ad alta leva. Un outlier vicino a $\bar{x}$ o un punto ad alta leva sulla retta hanno una distanza di Cook bassa.

Regola pratica: un'osservazione è da guardare con attenzione se $D_i > \frac{4}{n}$ (Figure 3.25). Se $D_i > 1$, l'osservazione è sicuramente influente.

```
influente <- rbind(dat, c(soggetto = 11, eta = 18, statura = 120))
mod_inf <- lm(statura ~ eta, data = influente)
influente$leva <- hatvalues(mod_inf)
influente$cook <- cooks.distance(mod_inf)
```

```
round(influente, 2)
```

```
   soggetto eta statura leva cook
1         1   6     110 0.20 0.29
2         2   6     119 0.20 0.06
3         3   7     125 0.15 0.00
4         4   8     123 0.11 0.01
5         5   8     133 0.11 0.01
6         6   9     129 0.09 0.00
7         7  10     140 0.09 0.04
8         8  10     142 0.09 0.05
9         9  11     138 0.11 0.02
10       10  12     150 0.14 0.21
11       11  18     120 0.70 9.25
```

Una soglia spesso usata: $D_i > 4/n$

```
plot(influente$cook, type = "h", lwd = 3,
     xlab = "osservazione", ylab = "distanza di Cook")
abline(h = 4 / nrow(influente), col = "firebrick", lty = 2, lwd = 2)
```

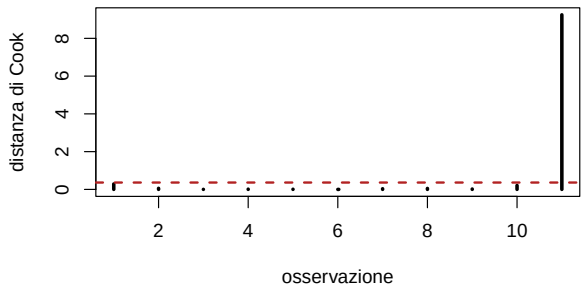


Figure 3.25: Distanza di Cook

```
which(influente$cook > 4 / nrow(influente))
```

```
[1] 11
```

4 Funzioni riassuntive dei moduli

4.1 Operatori

4.1.1 Aritmetici

Simbolo	Significato	Note
+	Addizione	
-	Sottrazione	
*	Moltiplicazione	
/	Divisione	
^	Elevamento a potenza	

4.1.2 Di confronto

Simbolo	Significato	Note
<	Minore	
>	Maggiore	
<=	Minore uguale	
>=	Maggiore uguale	
==	Uguale	
!=	Diverso	

4.1.3 Logici

Simbolo	Significato	Note
&	Congiunzione	Intersezione tra insiemi $\wedge$ ($\cap$)
|	Disgiunzione	Unione tra insiemi $\vee$ $\cup$
!	Negazione	$\neg$
`xor(A, B)`	Disgiunzione esclusiva	
%in%	Inclusione	

4.2 Valori di verità

Valore	Significato	Note
TRUE	Vero	Es. X>2 confronta ogni valore di X con 2
FALSE	Falso	

4.3 Operatori e funzioni utili

Funzione	Significato	Note
`<-`	Operatore di assegnazione	Uguale a `assign()` oppure `=`
`c(...)`	Concatenazione di valori	`c(1:n)` restituisce valori da 1 a n con passo 1; `c(1,n)` restituisce solo 1 e n
`vettore[i]`	Operatore di selezione per i vettori, dove `i` è l'indice della posizione a cui si vuole accedere	Es. `A<-c(1,8,5,0)`; `A[c(1,4)]` restituisce valori in posizione 1 e 4
`dati[i, j]`	Operatore di selezione per strutture di dati (matrici, dataframe), dove `i` è l'indice della riga, `j` della colonna	Es. `data[4, 12]` restituisce il valore della cella che si trova all'incrocio tra la quarta riga e la dodicesima colonna
`data$variabile`	Visualizza la variabile richiamata	
`read.csv2("")`	Legge un file di dati esterno	`sep =";", dec =",", header = TRUE`

4.4 Convertitori

Funzione	Significato	Note
`as.numeric(A)`	Converte in variabile numerica	
`as.factor(A)`	Converte in variabile categoriale/nominale	Richiede variabile testuale
`factor(A, levels=c("a1",...,"ak"))`	Converte in variabile categoriale con i livelli nell'ordine indicato	Va inserito l'ordine dei livelli teorici
`as.matrix(data.frame)`	Converte un data.frame in oggetto numerico	

4.5 Strumenti

Funzione	Significato	Note
`remove(...)`	Rimuove valori assegnati	
`rm(...)`	Rimuove oggetti	
`rm(list=ls())`	Rimuove tutto ciò che è in memoria	
`ls()`	Visualizza oggetti in memoria	
`example(comando)`	Fornisce esempio d'uso	
`?comando`	Mostra guida	

4.6 Funzioni matematiche

Funzione	Significato	Note
`log(...)`	Logaritmo naturale	
`exp(...)`	Esponenziale	
`abs(...)`	Valore assoluto	
`sqrt(...)`	Radice quadrata	
`round(...)`	Arrotondamento	
`sign(...)`	Funzione segno	
`pi`	Costante pi greco	
`max(...)`	Valore massimo	
`min(...)`	Valore minimo	
`sum(...)`	Somma	
`mean(...)`	Media	
`mean(..., trim=n)`	Media troncata	Elimina outlier con 0<=n<=0.5
`sd(...)`	Deviazione standard	
`var(...)`	Varianza	
`median(...)`	Mediana (2° quartile)	
`quantile(X, probs)`	Calcola quantile	
`IQR(...)`	Differenza interquartilica, $Q_3 - Q_1$	
`range(...)`	Intervallo (min e max)	Identico a `c(min(X),max(X))`
`rep(n, k)`	Replica `n` per `k` volte	
`dim(...)`	Dimensioni oggetto tabellare	`dim(...)[1]` righe, `dim(...)[2]` colonne
`cumsum(...)`	Somma cumulata	
`length(...)`	Numero elementi di un vettore	
`which(...)`	Posizione di valori che soddisfano condizione	Es. `which(x<3)`

Funzione	Significato	Note
`levels(...)`	Etichette assegnate a fattori	
`numeric(n)`	Vettore vuoto lunghezza n	
`rownames(...)`	Nomi righe	
`colnames(...)`	Nomi colonne	
`summary(...)`	Riassume contenuto	Anche modelli `lm(...)`
`str(X)`	Controlla struttura oggetto	
`cov(X,Y)`	Covarianza campionaria	
`cor(X,Y)`	Correlazione lineare	$-1 \leq r \leq 1$

4.7 Matrici

Funzione	Significato	Note
`matrix(y, r, c)`	Matrice $r \times c$ con valori y	Riempimento per colonne
`matrix(y, r, c, byrow=TRUE)`	Matrice	Riempimento per righe
`t(M)`	Trasposta di M	
`%*%`	Prodotto tra matrici	`*` fa invece il prodotto elemento per elemento

4.8 Liste e Data Frame

Funzione	Significato	Note
`list()`	Costruisce lista di oggetti	Prima creare oggetti, poi richiamarli nella lista
`data.frame(...)`	Costruisce tabella di variabili	Variabili devono avere stessa lunghezza
`apply(DATI,MARGIN,FUN)`	Applica funzione per righe (`1`)/colonne (`2`)	Es. `apply(DATI,1,sum)`
`cbind(...)`	Unisce colonne	Variabili devono avere stessa lunghezza
`rbind(...)`	Unisce righe	
`head(df)`	Mostra prime 6 righe dataframe	
`tail(df)`	Mostra ultime 6 righe dataframe	

4.9 Distribuzioni di frequenza

Funzione	Significato	Note
`table(A)`	Tabella frequenze di variabile	
`table(A,B)`	Tabella di contingenza	
`margin.table(T,margin)`	Frequenze marginali	`1` righe, `2` colonne
`T/sum(T)`	Frequenze relative	
`prop.table(T,margin)`	Probabilità condizionate	
`cumsum(table(X))`	Frequenze cumulate assolute	
`cumsum(table(X)/length(X))`	Frequenze cumulate relative	
`table(...)/length(...)`	Frequenze relative	

4.10 Funzioni grafiche

Funzione	Significato	Note
`hist(X,breaks)`	Istogramma	`breaks=` Calcolo ampiezza bin
`boxplot(X)`	Boxplot	anche `boxplot(X,Y,names=c("X","Y"))`
`runif(N,0,1)`	`N` valori casuali presi da distribuzione uniforme	tra 0 e 1
`density(X,bw)`	Stima densità	Usare `plot(density(...))`
`par(mfrow=c(n1,n2))`	Imposta finestra grafica multipla	`n1`=righe, `n2`=colonne
`plot(X,Y)`	Grafico di dispersione	

4.11 Associazione variabili categoriali

Funzione	Significato	Note
`chisq.test(T)$statistic`	Calcola chi-quadrato osservato	
`chisq.test(T)$expected`	Calcola frequenze teoriche	

4.12 Regressione lineare

Funzione	Significato	Note
`lm(Y~X)`	Modello regressione lineare	Y dipendente, X indipendente
`abline(retta,...)`	Traccia retta sul grafico	`abline(lm(Y~X))`
`cor(X,Y)^2`	Bontà di adattamento (R^2)	
`retta$residuals`	Residui modello	Es. `plot(X,E); abline(0,0)`