Laboratorio R

Sommario

Definizioni

Statistica descrittiva:

Metodi e strumenti per *organizzare* e *visualizzare i dati*.
Statistica descrittiva Univariata: Prende in analisi una sola variabile di una determinata popolazione
Statistica descrittiva Bivariata: Prende in analisi due variabili di una determinata popolazione
Statistica descrittiva Multivariata: Prende in analisi n variabili di una determinata popolazione

Statistica inferenziale:

Raccolta di metodi e strumenti che permettono di generalizzare un determinato fenomeno osservato in un campione all'intera popolazione.

- **Campione:** sottoinsieme selezionato degli elementi appartenenti alla popolazione
- **Popolazione:** insieme di persone, oggetti o fenomeni di interesse
- **Variabile:** qualsiasi caratteristica osservabile e misurabile che possa assumere almeno due stati o livelli
- **Osservazione:** valore che assume una variabile in un determinato elemento della popolazione di interesse.
- **Unità statistica:** elemento unitario della popolazione statistica

Variabile Quantitativa: Variabile che assume valori numerici, si dice **continua** se assume valori rappresentabili in un insieme continuo, se invece assume solo valori interi si dice **discreta**.
Variabile Qualitativa: Variabile che non assume valori numerici può essere **ordinale** se i valori che assume sono organizzabili seguendo un ordine altrimenti è **categorica** se i valori assunti non sono ordinabili e rientrano in categorie.

Campionamento:

Creare un campione con il minimo grado di distorsione (bias).
Quando parliamo di campionamento casuale significa affidarsi al caso per selezionare quali elementi della popolazione entreranno a far parte del nostro campione. L'intuizione è che se ogni elemento della popolazione ha la stessa probabilità di essere scelto a caso, allora, scegliendo a caso un numero sufficiente di entità allora le caratteristiche del mio campione saranno simili a quelle della popolazione.

Tipi di dato in R

- **Numeric:** numeri (interi, reali, complessi)
- **Character:** sequenze di caratteri (vanno messe tra virgolette)
- **Logical:** valori logici (T, F, NA – not avaliable)

Funzioni per verificare il tipo:

- `Is.logical()`
- `Is.numeric()`
- `Is.character()`

Funzioni per convertire il tipo:

- `as.logical()`
- `as.numeric()`
- `as.character()`

Gestione area di lavoro

- `setwd("path")`: serve per impostare una nuova directory di lavoro
- `getwd()`: controlla qual è l'attuale directory di lavoro
- `q(save ="yes/no")`: salvare o non salvare l'area di lavoro

Oggetti:

In R, il modo per poter riutilizzare un'entità è quello di salvarla in memoria nella forma di un oggetto identificato da un nome.
La sintassi è la seguente:

```
nome_oggetto <- contenuto_oggetto
```

Gestione degli oggetti:

- `ls()`: lista dei nomi degli oggetti
- `ls.str()`: dettagli degli oggetti salvati in memoria
- `ls.(pat ="s")`: elenco degli oggetti il cui nome contiene il carattere s
- `rm(nome_oggetto)`: elimina l'oggetto
- `rm(list=ls())`: elimina tutto

Tipi di oggetto:

I principali tipi di oggetto in R sono i seguenti: **vettori, fattori, matrici, array e Data Frame**.
Ogni tipo di oggetto è caratterizzato da un insieme id proprietà.
Per stabilire la classe di un oggetto (o il tipo di un dato) ci sono le seguenti funzioni

- `class() #restituisce il tipo di dati contenuto nel vettore`
- `is.matrix()`
- `is.array()`
- `is.list()`

- **`is.data.frame()`**
- **`is.na()`**
- **`is.vector()`**

Vettori:

Un vettore è una sequenza ordinata di **elementi di uno stesso tipo** (numeric, character, logical) ciascuno separato da una virgola.
Tra i modi in cui si può costruire un vettore, il più comune è con l'utilizzo della funzione c().

- **`f.num.vec`** <- c(1,4,5,3.3)
- **`hoc.vec`** <- c("Pino", "Rodolfo", "Gianciccio")
- **`m.vec`** <- c(1:5) //intervallo di numeri da 1 a 5

Funzioni su vettori:

- **`append(v1, v2, … , vn) o append(v1, v2, after ="numero")`:** funzione che mi consente di concatenare vettori, il parametro after serve per dire in che posizione far avvenire la concatenazione.

- **`seq(1,7)`:** serve per creare sequenze di numeri, in questo caso da 1 a 7, opzionale è il parametro by per esempio seq(1,7, by=2) che mi permette di avere i numeri da 1 a 7 a intervallo di 2.
- **`str(nome_vettore)`:** fornisce una panoramica sulle caratteristiche del vettore
- **`length(nome_vettore)`:** restituisce la dimensione del vettore

- **`nome_vettore[1]`:** restituisce il primo elemento
- **`nome_vettore[c(1,12)]`:** restituisce il primo e il 12 elemento, mentre con c(1:12) dal primo al 12esimo
- **`nome_vettore[-c(2,6)]`**: restituisce tutti gli elementi eccetto il secondo e il sesto.

- **`head(nome_vettore)`:** restituisce i primi elementi del vettore
- **`tail(nome_vettore)`:** restituisce l'ultima parte del vettore

- **`nome_vettore [>,==, <, <=. >=, !=, &, |] condizione`:** restituisce il risultato in seguito all'operazione logica
- **`which(nome_vettore [>, ==, <, >=. <=. !=] condizione)`:** restituisce gli indici che rispettano la condizione
- **`nome_vettore %in% c(1, 9.3)`:** permette di verificare quali elementi del vettore sono presenti nell'altro vettore.

Creazione di un sottovettore:

- **`sample(nome_vettore, x)`:** estrae da un vettore un sottoinsieme casuale di x elementi
- **`nuovo_vettore <- vecchio_vettore[vecchio_vettore > 0]`**

Ordinamento vettore:

- **`sort()`:** ordina gli elementi del vettore
- **`order()`:** restituisce per ogni elemento del vettore ordinato la posizione che questo occupava nel vettore originale
- **`max(), min()`:** massimo e minimo elemento del vettore
- **`which.max(), which.min()`:** indice del valore massimo e minimo

- **`unique()`:** ci fornisce l'elenco degli elementi presenti nel vettore
- **`table()`:** ci dice la frequenza dei vari elementi del vettore

Salvataggio del contenuto del vettore:

- **`cat(nome_vettore, file = "nomefile.txt", sep = "\n" o "\t" o ";", append = T/F)`**
- **`scan()`:** carica in un vettore i dati salvati in un file

Data Frame:

Tipo di oggetto usato in R per memorizzare e gestire matrici di dati.
Un dataframe è pensabile come una tabella in cui le righe corrispondono alle osservazioni e le colonne alle variabili.
Possiamo codificare un dataframe codificando ogni informazione in un vettore e poi unendo questi vettori.
Ricordiamo che il dataframe può contenere diversi tipi di dati al proprio interno.

```
> PoS <- c("ADJ", "ADV", "N", "CONJ", "PREP")
> TokenF <- c(421, 337, 1411, 458, 455)
> TypeF <- c(271, 103, 735, 18, 37)
> class <- c("open", "open", "open", "closed", "closed")
```

Questi sono i vettori e li unisco dentro un dataframe con il seguente comando:

`nome_df <- data.frame(PoS, TokenF, …, class)`

Manipolazione Data Frame:

- **`colnames(nome_df)`** – elenca le colonne che sono contenute nel df.
- **`Rownames(nome_df)`** – elenca le righe che ci sono nel df.

Entrambe queste funzioni possono essere utilizzate anche per modificare il nome delle righe/colonne.

- **`nome_df[indice]`** oppure **`nome_df["nome_colonna"]`** – ottengo un df avente i dati di solo quella colonna.
  Posso anche fare **`nome_df[1,]`** o **`nome_df[4,]`** per scegliermi le parti che voglio che stiano nel mio sotto-dataframe.

Notazione che serve per selezionare dati che soddisfano una condizione:

- **`nome_df[nome_df$nome_colonna > 10]`**
  - **`$ ->`** serve per accedere alla colonna
  - In questo caso prende in considerazione le righe del df dove il valore di nome_colonna > 10
- **`nome_df[nome_df$nome_colonna > 10, 3]`**
  - Prende in considerazione soltanto le righe dove il valore della terza colonna è > 10.
- **`nome_df[nome_df$nome_colonna == "qualcosa", "altra_colonna"]`**
  - crea un vettore contenente i valori di altra_colonna per le righe in cui nome_colonna è uguale a "qualcosa".
- **`nome_df[nome_df$colonna1 == "qulcosa" & colonna3 > 1000, ]`**
  - stessa cosa ma con due condizioni

Ordinamento:

- **`nome_df[order(nome_df$nome_colonna), ]`**
  - ordina il df in base ad una colonna
- **`nome_df[order(nome_df$nome_colonna, nome_df$nome_colonna2), ]`**
  - ordina il df in base a più colonne

Aggiungere variabili e osservazioni

- *`ndf$random <- sample(500:1500, 5); ndf`* # nuova variabile
- `ndf$tandom <- NULL; ndf` #elimino variabile
- `ndf["P", ] <- c(515, 25, "closed"); ndf` # nuova osservazione
- `ndf["P", "nome_colonna2"] <- 55; ndf` #modifica valori
-

<u>Come importare un dataframe, dopo averlo inserito nella cartella di lavoro:</u>

- PRIMA DEVO FARE **setwd"path")**

  - **`corpora <- read.table("corpora.txt", header = T, fileEncoding= "UTF-8", stringsAsFactors=T) #TXT`**

  - **`PW.prod <- read.table("pictureWord.RAW.csv", header = T, fileEncoding = "UTF-8", sep = ";", stringsAsFactors = T) #CSV`**

Ovviamente è necessario cambiare nomi!!!
`stringAsFactors = T` -> è un attributo che trasforma i vettori di stringhe in dati di tipo FATTORE ovvero corrispondenti alle variabili categoriali. (i valori vengono chiamati "livelli"). Importante da spuntare per far riconoscere i valori di variabili categoriali come fattori.
`read.table()` crea il df che può avere i seguenti parametri:

- **`File`**: il file di testo dove sono caricati i dati
- **`Header:`** valore logico che indica se la prima riga è l'intestazione [se marcato yes la prima riga del file viene interpretata come il nome delle colonne]
- **`Sep:`** il carattere usato come separatore
- **`Dec:`** il carattere usato come separatore decimale
- **`Row.names`**: un vettore contenente i nomi delle osservazioni oppure il numero della colonna che li contiene [se impostato automatic dà degli indice alle righe. Non riconosce la prima colonna come nomi delle righe]

Statistica I:

Distribuzione:

Rappresentazione del modo in cui le modalità (i valori) di una variabile si distribuiscono in un dato campione o in una popolazione di interesse.

- **Distribuzione Unitaria**: si riporta ogni valore per ogni unità statistica
  - **`> nome_df$nome_colonna`**
- **Distribuzione di frequenza**: rappresentazione grafica o tabellare delle frequenze delle modalità di una variabile.

Frequenza Assoluta:

Numero di unità statistiche in cui una variabile X assume una certa modalità Xi.

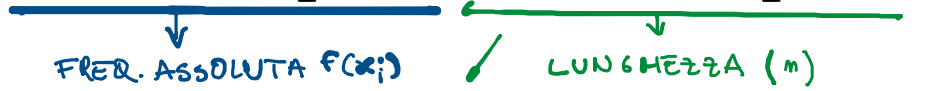
>table(nome_df$nome_colonna) – per ogni valore della variabile ci dice quante osservazioni ci sono.
>table() è una funzione che ha un comportamento diverso a seconda del numero di vettori che riceve in input:
Se riceve **un vettore** restituisce la distribuzione di frequenza, se riceve **due vettori** restituisce una tabella di contingenza (con k vettori, un array k-dimensionale).
Possiamo dividere le modalità in intervalli (regola pratica il range deve essere tra 5 e 20)

```
>table(cut(nome_df$nome_colonna, breaks 10))
```

Frequenza relativa:

Rapporto tra la frequenza assoluta della modalità Xi e il numero di unità statistiche nel nostro campione indicato con n. [f(xi) / n]

```
>prop.table(table(nome_df$nome_colonna))
Oppure:
>table(nome.df$nome_colonna)/length(nome.df$nome_colonna))
```



Frequenza percentuale:

Si tratta della frequenza relativa moltiplicata x 100

```
>prop.table(table(nome_df$nome_colonna)) * 100
```

Frequenza cumulata:

Frequenza di tutte le unità statistiche che presentano una modalità minore o uguale ad Xi

```
>cumsum(table(nome_df$nome_colonna)) #assoluta
>cumsum(prop.table(table(nome_df$nome_colonna))) #relativa
```

Grafici in R:

I grafici sono **strumenti descrittivi** che ci permettono di descrivere i nostri dati e renderli conoscitivi ad altre persone. Permettono di avere in maniera chiara un'idea sulla distribuzione dei dati.
Il modo più semplice per realizzare un grafico è con la funzione plot().

```
>plot(c(1, 3, 4, 6), c(1, 44, 33, 12))
>plot(nome_df$colonna_df)
```

Praticamente plot è una funzione generica per creare dei grafici dove il **primo vettore** che gli do mi rappresenta l'ascissa e il **secondo vettore** l'ordinata.

ScatterPlot:

Lo scatterplot è anche detto **grafico a dispersione**, qui ogni osservazione è rappresentata da un punto in uno spazio cartesiano i cui assi fanno riferimento ai valori delle due variabili quantitative in analisi.
Per fare uno scatterplot basta eseguire i seguenti comandi:

```
vettore <- c(1,3,4,5,7,53,12)
plot(vettore) # avrò un pallino per ogni punto del vettore.
```

Lo scatterplot è il grafico più comune per quando mi ritrovo a lavorare con delle variabili **QUANTITATIVE**:

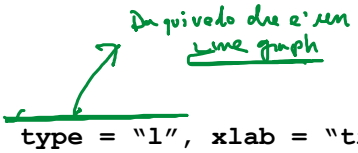
Voglio a questo punto modificare l'aspetto del grafico per renderlo più preciso:

```
> plot(alcuni.numeri, main = "uno scatterplot") # titolo
> plot(alcuni.numeri, ylab = "ampiezza") # etichetta asse y
> plot(alcuni.numeri, xlab = "rango") # etichetta asse x
> plot(alcuni.numeri, xlim = c(0,9)) # ampiezza asse x
> grid() # aggiungiamo una griglia al grafico
```

Lo uso per rappresentare graficamente il rapporto tra due variabili.
> plot(es1$CONSTRUCTION, es1$V_CHANGPOSS, ylab="V_change", xlab = "construction")

LinePlot:

Per creare un lineplot è necessario eseguire il seguente comando:



>plot(cumsum(prop.table(table(nome_df$colonna_df))), type = "l", xlab = "titolo X", ylab = "titolo Y"); grid() → parametro opzionale mette la GRIGLIA

Si utilizza per studiare la distribuzione di variabile quantitative.

Grafico a Torta:

I grafici a torta sono utilizzati per mostrare **frequenze di variabili categoriali** sotto forma di cerchio suddivisi in spicchi dalla grandezza variabile in base alla frequenza relativa delle varie modalità della variabile in oggetto.
Per realizzare un grafico a torta.

```
Class_freq <- table(nome_df$nome_colonna)
Pie(class_freq)
```

Possiamo anche ampiamente modificare il grafico a torta:

```
> pie(classFreqs, col = c("blue", "black")) # cf. colors()
> pie(classFreqs, col = c(rgb(1,1,1), rgb(81/255,0,0)))
> pie(classFreqs, col = rainbow(length(classFreqs))) #crea
  tanti colori dell'arcobaleno quanti sono i valori delle classi ottenuti con
  length
> pie(classFreqs, col = grey.colors(length(classFreqs)))
> pie(classFreqs, col = grey.colors(2))
> pie(classFreqs, col = sample(colors())) # colori casuali
```

Diagramma a Barre:

I diagrammi a barre (ortogrammi) sono utili per rappresentare la distribuzione di **variabili categoriali**:

- Ogni barra è associata ad una modalità della variabile in questione
- L'altezza delle barre è proporzionale alla frequenza delle modalità
- Tutte le barre hanno la stessa larghezza.

Per realizzare un diagramma a barre in R per prima cosa è necessario stabilire una tabella di frequenza con **table()**:

>classFreqs <- table(nome_df$Variabile)

Successivamente è possibile creare il grafico:

>barplot(classFreqs)

Per quanto riguarda le manipolazioni grafiche:

```
> barplot(classFreqs)
> barplot(classFreqs, horiz = T) #barre orizzontali
> barplot(classFreqs, cex.names = 0.5) #riduce la dimensione delle
etichette
> barplot(classFreqs, col = grey.colors(2))
> barplot(classFreqs, space = 0)#nessuno spazio tra colonne
> text(barplot(classFreqs), classFreqs, pos = 1, labels = classFreqs)
```

Si usa in esercizi come "**rappresentare la distribuzione di frequenza** della variabile x in base alla variabile y che ha il valore "n" o "s".

Istogramma:

Servono per quando abbiamo a che fare con variabili di tipo **quantitativo/continuo.**
Come per i diagrammi a barre, anche negli istogrammi i valori sono rappresentati per mezzo di barre, ma:

- **Ogni barra è associata ad una modalità o ad un intervallo di modalità** (**bins**) della variabile in questione
- L'area delle barre è proporzionale alla frequenza delle modalità o dell'intervallo di modalità.
- Non necessariamente tutte le barre devono avere la stessa larghezza
- Grafico adatto per **variabili numeriche** (continue)
- **Altezza barra** = numero di osservazioni in cui il valore delle x ricade in un certo intervallo, l'altezza è anche la **densità della classe** ovvero il rapporto tra la frequenza relativa della classe e la sua ampiezza.

In R si può realizzare grazie a **hist():**

```
> RTs.inliers <- PW.prod[PW.prod$Inlier == 1,5]
> hist(RTs.inliers)
> hist(RTs.inliers, breaks = 5)
```

In un istogramma l'altezza di ogni barra viene determinata sulla base delle densità della classe:

- Rapporto tra la frequenza relativa di una classe e la sua ampiezza
- Data una classe I: d(I) = FreqRel(I) / |I|

Quando le classi hanno tutte la stessa ampiezza, l'altezza di ciascuna barra è proporzionale alla sua frequenza (e quindi alla sua area):

```
> hist(RTs.inliers, freq = T) # plotta frequenze
> hist(RTs.inliers, freq = F) # plotta densità e l'area totale è 1
```

Quindi quando l'ampiezza delle basi è la stessa l'istogramma non cambia, se l'ampiezza è diversa l'istogramma cambia.

```
> bpoints <- c(0, 500, 600, 700, 800, 1500)
> hist(RTs.inliers, main = "", breaks = bpoints, freq = F)-> density
> hist(RTs.inliers, main = "", breaks = bpoints, freq = T)-> freq.
```

Quello che davvero conta negli istogrammi non è tanto l'altezza, ma l'AREA dell'istogramma (frequenza)
Altezza di ogni barra = **DENSITÀ** della classe (rapporto **frequenza relativa** / **ampiezza dell'intervallo**).
freq = F → il nome dell'asse delle y diventa "density", ma l'istogramma non cambia, se ho tutte le barre con la **stessa ampiezza.**

Se prendo tutti i rettangoli e ne sommo le aree, il risultato sarà 1, infatti → ciascuna area plotta la **frequenza relativa.**
Se l'area è la frequenza relativa, la freq relativa la divido per l'ampiezza dell'intervallo e ottengo la densità
se metto **freq = T**, l'istogramma non cambia, perché l'ampiezza delle basi è la stessa.
Se invece dividessi l'intervallo in parti NON uguali, l'istogramma avrà rappresentazioni diverse se plotto la densità o la frequenza (cambiando l'intervallo, il valore della densità cambia)

- stesse dimensioni → stesso istogramma per frequenza e per densità
- dimensioni diverse → nella densità il valore quantitativo nasce dall'area; quindi, le rappresentazioni cambiano
- densità → abbiamo a che fare con una variabile di tipo continuo

Statistica II

Gli indici statistici si dividono in due gruppi: indici di **tendenza centrale** e di **dispersione**.

Indici di tendenza centrale – summary():

- Consentono di **localizzare la distribuzione**, ovvero individuare il punto o i punti attorno al quale o ai quali si concentra.
- Sono anche descritti come gli indici che cercano di **riassumere il comportamento di una variabile** con un **valore**.
- In R posso tracciare una retta o più segmenti su un grafico già attivo con le funzioni:
  - **abline()** – argomenti = intercetta e coefficiente angolare
  - **lines()** – argomenti = 2 vettori
  - **segments()**
  - **points()** – per tracciare punti

Media Aritmetica:

Rapporto tra la somma dei valori (Xi) assunti da una variabile in un campione e la numerosità del campione (n).
La media aritmetica è un indice molto **sensibile alla presenza di outliers**.

```
> RTs.inliers <- PW.prod[PW.prod$Inlier == 1,5]
> mean(RTs.inliers)
> mean(table(blind.prod$Class)) #frequenza media
```

Mediana:

In un insieme di dati ordinati, la mediana è il valore che occupa la posizione centrale.

```
> median(RTs.inliers)
> median(blind.prod$FeatureOrder)
```

Ordinando le osservazioni del nostro campione, la mediana è calcolata nel seguente modo:

- Se il campione è costituito da un numero di spari di osservazioni, essa corrisponde al valore dell'osservazione che occupa la posizione **(n+1)/2** dove n è la dimensione del campione
- Se il numero di osservazioni è pari essa è stimata a partire dai due valori che occupano le posizioni **n/2** e **(n/2)+1**

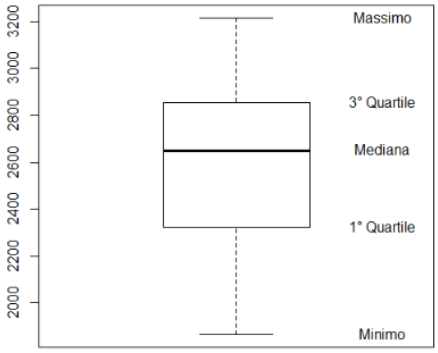
- La mediana **non è influenzata** dalla presenza di **outliers**

Quartili:

Un gruppo particolare di quantili sono i quartili che dividono il campione ordinato in quattro gruppi di numerosità uguale:

```
> quantile(c(1,2,2,5,7,9,9), .25) # i.e. primo quartile
> quantile(c(1,2,2,5,7,9,9), .5) # i.e. secondo quartile
> quantile(c(1,2,2,5,7,9,9), .75) # i.e. terzo quartile
```

Il comando per avere tutti gli indici di tendenza centrale: **`summary(nome df).`**



I quartili possono essere visualizzati anche per mezzo di un **Boxplot** (o diagramma a scatola e baffi)

```
> boxplot(as.vector(table(blind.prod$Class)))
> summary(as.vector(table(blind.prod$Class)))
```

I baffi si estendono fino ai valori massimi o minimi del campione a meno che questi non distino più di **1.5 volte** lo scarto interquartile. A quel punto stiamo parlando di outliers.

```
> boxplot(RTs.inliers)
> boxplot.stats(RTs.inliers)
#estrae I dati che formano
 le component del boxplot
```

Moda:

La modalità (o classe) a cui corrisponde la frequenza (assoluta o relativa) massima della distribuzione.

- A seconda della natura della variabile, la moda può fare riferimento **ad una modalità**

```
> max.freq.f0 <- max(table(blind.prod$FeatureOrder))
> which(table(blind.prod$FeatureOrder) == max.freq.f0)
```

- Oppure **ad una classe**

```
> max.freq.f <- max(table(blind.prod$FeatureIt))
> which(table(blind.prod$FeatureIt) == max.freq.f)
```

- **Distribuzioni K-modali:**
  - Non necessariamente una distribuzione deve avere una sola moda, nel caso bimodale ci sono 2 picchi simili nelle frequenze.

Indici di Dispersione:

Forniscono un'indicazione di quanto i dati **siano dispersi** o concentrati rispetto agli indici di tendenza centrale. Mai riportare un indice di tendenza centrale senza riportare anche un indice di dispersione.

Range (campo di variazione):

Si tratta della **differenza tra il valore massimo della distribuzione e quello minimo (Xmax – Xmin).**

```
> RTs.inliers <- PW.prod[PW.prod$Inlier == 1,5]
> max(RTs.inliers) - min(RTs.inliers)
> diff(range(RTs.inliers)) # range() non restituisce il range, ma
  solo I due valori estremi!
```

Range() da solo non restituisce il range ma solo i due valori estremi!

Scarto Interquartile:

Differenza tra il terzo e il primo quatrtile ($X_{0.75} - X_{0.25}$).

- IQR(RTs.inliers)

Entrambi questi indici ignorano i valori che non sono agli estremi.

Scarto:

Per ogni dato, la sua distanza dalla media, in pratica nell'esempio vediamo il vettore ex e la distanza di ogni elemento di ex dalla media di ex stesso.

```
> ex <- c(0, 0, 1, 3, 4, 4)
> scarti <- ex - mean (ex)
```

Scarto medio Assoluto:

Media dei valori assoluti degli scarti.

```
> mean(abs(scarti))
> mean(abs(RTs.inliers - mean(RTs.inliers)))
```

$$s_m = \frac{\sum_{i=1}^{n} |x_i - \overline{X}|}{n}$$

Varianza della Popolazione:

Devianza = somma dei quadrati degli scarti della media.
La devianza è usata in molte statistiche, ma non è un indice di dispersione ottimale data la sua sensibilità alla dimensione del campione.
Varianza della popolazione = Somma dei quadrati degli scarti dalla media (devianza) divisa per N (numero di elementi della popolazione):

```
> sum(abs(scarti^2)) / length(scarti)
```

$$\sigma^2 = \frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}$$

Varianza campionaria:

Quando cerchiamo di stimare la varianza di una popolazione a partire da un campione, dobbiamo correggere la varianza dividendo la somma dei quadrati degli scarti dalla media per i gradi di libertà.
Gradi di libertà: numero di misure indipendenti meno il numero di parametri calcolati da queste misure (in questo caso la media)

```
> var(ex)
> sum(abs(scarti^2)) / (length(scarti) - 1)
```

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \overline{X})^2}{n-1}$$

Deviazione Standard o Scarto Quadratico Medio:

Essendo una misura di grandezza quadratica rispetto alla media, la varianza non è direttamente comparabile agli altri indici statistici (es alla media).
La deviazione standard è definita come radice quadrata della varianza.

```
> sd(ex)
> sqrt(var(ex))
```

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{X})^2}{n-1}}$$

Intuitivamente non è molto diverso dallo scarto quadratico medio assoluto:
una misura di dispersione della media.
A differenza di altri indici (come la media), **la varianza e la deviazione standard non cambiano al cariare dei riferimenti delle misurazioni**:

```
> mean(ex + 5); mean(ex)
> var(ex + 5); var(ex)
> sd(ex + 5); sd(ex)
```

Sono al pari di altri indicatori sensibili ad altri tipi di trasformazione come, per esempio, il cambio di misura da cm a mm.

```
> mean(ex * 5); mean(ex)
> var(ex * 5); var(ex)
> sd(ex * 5); sd(ex)
```

Standardizzazione:

Questi indicatori possono essere anche usati per trasformare i dati in modo da rendere comparabili valori provenienti da scale diverse.
Per esempio, in quale esame sono stato più bravo: LA, LG o Fonologia seguita in Svizzera?

```
> v.applicata <- c(18, 18, 20, 20, 20, 24)
> v.generale <- c(18, 18, 20, 30, 30, 27)
> v.fonologia.ch <- c(3, 3.5, 3, 5, 5.5, 4.5)
```

Per standardizzazione si intende il cambiamento della scala della variabile in modo che la media sia uguale a ‘ e la deviazione standard sia uguale a 1 (Scala z):
Per standardizzare una variabile occorre trasformare tutte le sue misurazioni in Punteggi z (Z scores)

Punteggi Z (Z scores):

Rapporto tra scarto e deviazione standard.
Data una rilevazione, il suo punteggio z indica quante deviazioni standard essa si discosti dalla media.

```
> (v.applicata - mean(v.applicata )) / sd(v.applicata)
> scale(v.applicata)#restituisce una matrice con z scores
> as.vector(scale(v.applicata)) #per ottenere nuovamente
  un vettore
```

$$z_i = \frac{x_i - \bar{x}}{s}$$

Analisi Bivariata – due variabili caregoriali e tabelle di contingenza

Riporta le frequenze congiunte delle due variabili in analisi.
Quando spesso ogni modalità di una variabile ricorre con ogni possibile modalità dell'altra variabile in esame.

```
> occhi <- sample(c("c","s"), 20, replace = T)
> capelli <- sample(c("c","b", "r"), 20, replace = T)
```

In R, frequenze assolute possono essere ottenute con **table()** :

```
> table(occhi, capelli)
```

Questo comando sostanzialmente permette un'analisi bivariata ottenendo una tabelle di contingenza.

- `prop.table()` può essere usata per calcolare frequenze relative:

```
> prop.table(table(occhi, capelli)) # default margin = NULL
> prop.table(table(occhi, capelli), margin = 1) # riga
> prop.table(table(occhi, capelli), margin = 2) # colonna
```

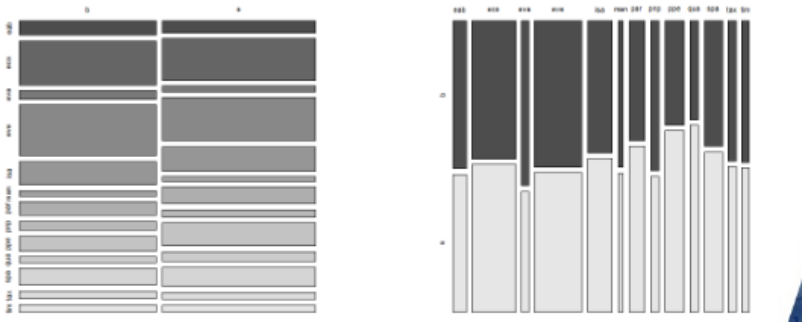
```
> prop.table(table(occhi, capelli))*100 #frequenze percentuali
> prop.table(table(occhi, capelli), margin = 1)*100 # se uso margin = 1 ottengo che quello che sommano a 1 sono le righe
> prop.table(table(occhi, capelli), margin = 2)*100 # se uso margin = 2 ottengo il contrario, ho il rapporto proporzionale rispetto al marginale della colonna
```

Variabile Dipendente e Indipendente:

- **Variabile Dipendente:** variabile di cui dobbiamo spiegare il comportamento
- **Variabile Indipendente** (aka Fattore): la variabile che influenza la distribuzione della variabile dipendente. Il tempo di reazione rispetto a una parola stimolo può essere in parte spiegato considerando la frequenza della parola.

Mosaic Plot – composizione bivariata con 2 variabili categoriali

```
> mosaicplot(FeatureTypeCoarse ~ Group, data = blind.prod,
  color = T, main = "")
> mosaicplot(Group ~ FeatureTypeCoarse, data = blind.prod,
  color = T, main = "")
```



Si può fare anche con il bar-plot.

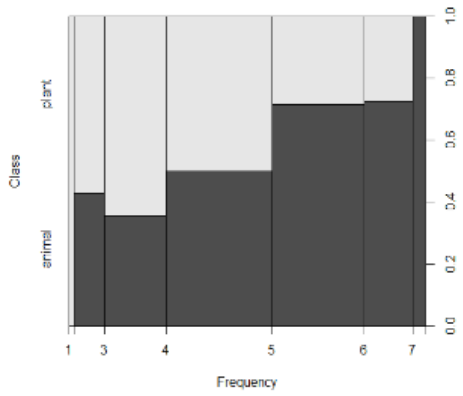
Analisi Bivariata – Variabile Categoriale – Variabile Numerica (Spine Plot):

- Carichiamo il dataframe "ratings " dalla libreria "languageR" (Baayen, 2008),

```
> install.packages("languageR"); library(languageR); data(ratings)
```

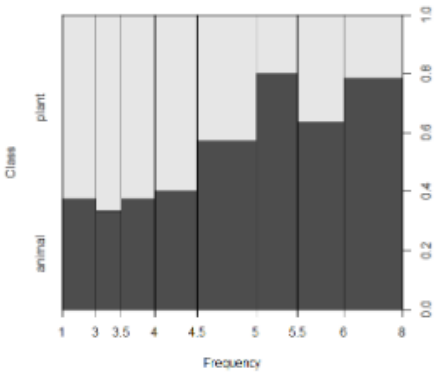
- Rappresenta graficamente la relazione tra 1 variabile **categoriale** ed una variabile **almeno ordinale**:

```
> spineplot(Class ~ Frequency, data = ratings) #log-frequency
```



Rappresenta graficamente la relazione tra 1 variabile dipendente qualitativa e una variabile indipendente almeno ordinale:

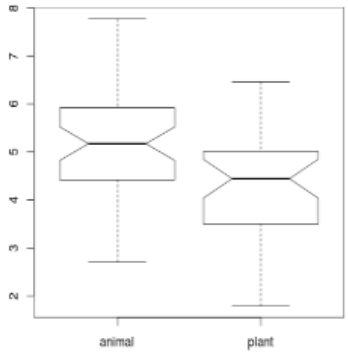
```
> spineplot(Class ~ Frequency, data = ratings,
  breaks = c(1, 3, 3.5, 4, 4.5, 5, 5.5, 6, 8))
```



Analisi Bivariata – Variabile Numerica – Variabile Categoriale (BoxPlot):

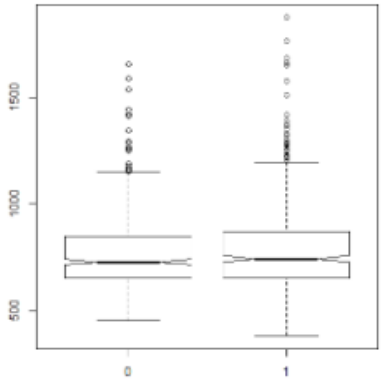
Possiamo usare un (notched) boxplot per esaminare la tendenza centrale di ogni singola modalità della variabile indipendente:

```
> boxplot(Frequency ~ Class, data = ratings, notch = T)
```



Possiamo usare un (notched) boxplot per esaminare la tendenza centrale di ogni singola modalità della variabile indipendente:

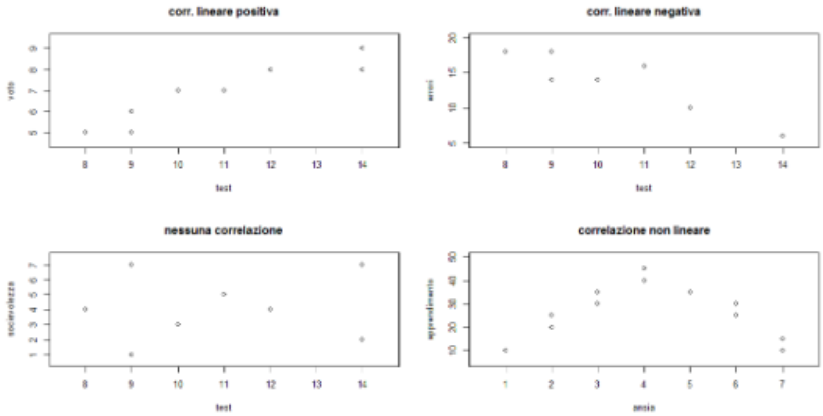
```
> boxplot(RT ~ CondSem, data = PW.prod, notch = T)
```



Notch = T → attributo che **anticipa** la nozione di quanto i dati sono statisticamente significativi; se i due notch **non si sovrappongono** significa che le differenze tra le mediane sono **statisticamente significative**, quindi **non è un dato casuale.**

Analisi Bivariata – Variabile Numerica – Variabile Numerica (Correlazione)

Date due variabili, la misura di quante queste covariano (ossia della loro tendenza a variare assieme)



Misura di quanto queste due variabili **covarino,** ho a che fare con l'analisi di **regressione** e gli **indici di correlazione:**

- vedo **quanto** e **come** queste variabili variano insieme
  - **1° caso**: correlazione di tipo **lineare positiva** (linea retta)
  - **2° caso**: correlazione di tipo **lineare negativa**
  - **3° caso**: assenza di correlazione
  - **4° caso:** correlazione non lineare, si identifica un pattern ma non è una linea retta
- Problema della correlazione lineare: **quanto** i dati fittano con questa retta (quanto sono vicini a una retta)

```
- > plot(corpora$COLFIS, corpora$REP)
- > model <- lm(REP ~ COLFIS, data=corpora)
- > abline(model, col="red", lwd=2) //noi abbiamo usato anche retta di regr.
```

Tipi di coefficienti di correlazione:

Metodi statistici utilizzati per misurare il grado di relazione tra due variabili.

```
> cor(corpora$COLFIS, corpora$REP)
```

Pearson:

- **Variabili quantitative** con scala di misura ad **intervalli** o **rapporti**;
- **Distribuzione** approssimativamente **normale;**
- **Relazione lineare tra due variabili continue;**
- Esempi: altezza-peso, temperatura-consumo di energia, etc.

P di Spearman:

- **Variabili ordinali** o **non necessariamente distribuite normalmente**;
- **Correlazione monotona (non necessariamente lineare) tra due variabili non ordinate;**
- **Esempi:** ranking prodotti di un sondaggio – vendite, classificazione atleti in gara – tempo impiegato.

T di Kendall:

- Variabili ordinali;
- Correlazione tra le classificazioni di due variabili ordinate;
- Esempi: ranking diprodotti di un sondaggio – ranking fatto da un altro gruppo, ordine delle posizioni di classifica di diversi giudici.

Correlazione punto-biseriale:

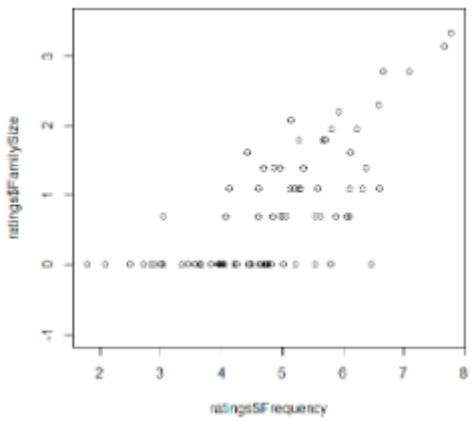
- Una variabile continua e l'altra dicotomica (binaria);
- Misura la correlazione tra una variabile continua e una dicotomica;
- Esempi: tempo di studio (variabile continua) – superamento di un esame, età (continua) – rischio di malattia

Come si usano:

Pearson:

- Plottiamo le variabili "Frequency" e "FamilySize" nel dataframe `ratings`:
- Carichiamo il dataframe "ratings " dalla libreria "languageR" (Baayen, 2008),

```
> install.packages("languageR"); library(languageR); data(ratings)
> plot(ratings$Frequency, ratings$FamilySize)
```



Comando:

- **`cor(ratings$Frequency, ratings$FamilySize, method = "pearson")`**

Definibile come:

- media aritmetica del prodotto dei valori standardizzati:

```
> sum(scale(ratings$Frequency) * scale(ratings$FamilySize)) /
  (length(ratings$FamilySize) - 1)
```

- rapporto tra la varianza condivisa dalle due variabili (covarianza) e il prodotto delle loro deviazioni standard:

```
> cov(ratings$Frequency, ratings$FamilySize) /
  (sd(ratings$Frequency) * sd(ratings$FamilySize))
```

Risultati:

Il coefficiente di correlazione *r di Bravais-Pearson* può assumere valori compresi tra [-1,1], con il seguente significato:

Coefficiente	Intensità	Tipo di Correlazione
$0.7 < r \leq 1$	Estremamente alta	Positiva
$0.5 < r \leq 0.7$	Alta	
$0.2 < r \leq 0.5$	Intermedia (tendenza)	
$0 < r \leq 0.2$	Bassa (trascurabile)	
$r \approx 0$	Correlazione Assente	
$0 > r \geq -0.2$	Bassa (trascurabile)	Negativa
$-0.2 > r \geq -0.5$	Intermedia (tendenza)	
$-0.5 > r \geq -0.7$	Alta	
$-0.7 > r \geq -1$	Estremamente alta	

Retta di regressione

Studia **quanto i valori assunti dalla variabile dipendente siano determinati da quelli della variabile indipendente**. Tecnica statistica utilizzata per comprendere la relazione tra due o più variabili.

Come funziona:

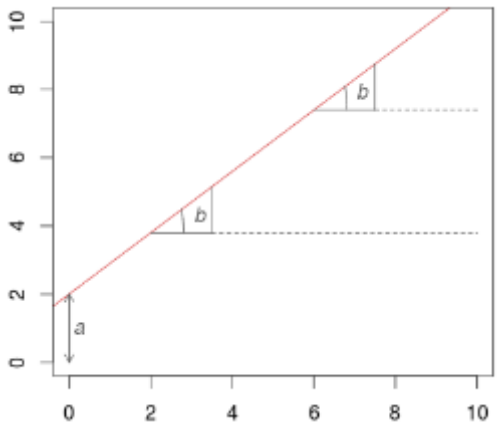
- Se la correlazione tra le nostre variabili è **alta,** possiamo prevedere i valori della variabile dipendente a partire da quelli della variabile indipendente.
- Se la correlazione tra le nostre variabili **è bassa** (o è inesistente) la stima migliore per i valori della variabile dipendente è la sua media.

N.B. Dato che trattiamo relazioni lineari, prevedere i valori della variabile dipendente equivale a caratterizzare l'equazione della retta che meglio approssima i nostri dati.

Equazione della retta:

Data l'equazione di una retta $y = ax + b$, i parametri da stimare sono:

- **Intercetta;**
- **Coefficiente di regressione.**



Possiamo stimare i parametri dell'equazione di regressione usando il comando **lm():**

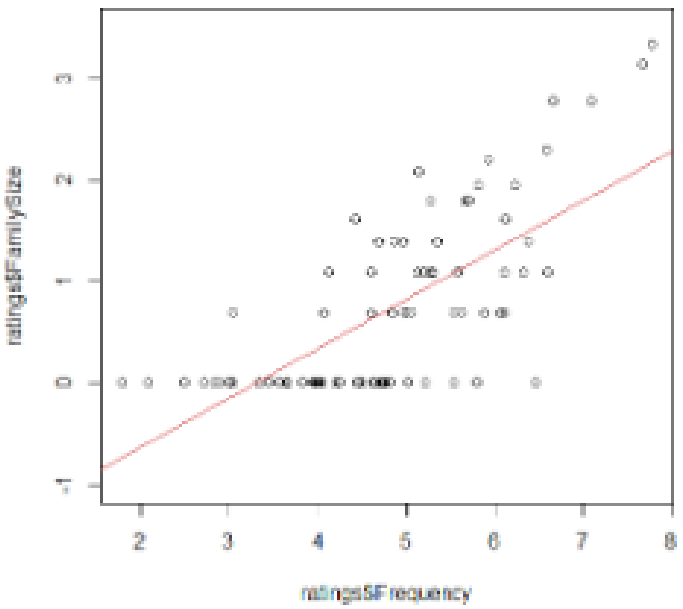
```
> model <- lm(ratings$FamilySize ~ ratings$Frequency)
> model

  Call:
  lm(formula = ratings$FamilySize ~ ratings$Frequency)

  Coefficients:
        (Intercept)  ratings$Frequency
            -1.5933              0.483
```

A questo punto possiamo tracciare la retta di regressione nel grafico:

```
> plot(ratings$Frequency, ratings$FamilySize, ylim = c(-
  1,3))
> abline(model, col="firebrick3")
```



Il coefficiente di correlazione è bi-direzionale (invertendo le due variabili non cambia).

La regressione è una relazione uni-direzionale (invertendo la variabile dipendente e quella indipendente otteniamo rette diverse)

Come valutare la bontà di un modello lineare?

Valutare la bontà di un modello lineare significa determinare quanto bene il modello rappresenti i dati osservati. In altre parole, si tratta di valutare quanto il modello si adatti ai dati e quanto sia accurata la sua capacità predittiva.

Coefficiente di determinazione R^2

Percentuale di variazione della variabile dipendente (Y) spiegata dalla variabile indipendente (X). Più il valore è alto, migliore è l'adattamento del modello ai dati.

Risultato: valore compreso tra 0 e 1

Comando:

Fare il coefficiente di correlazione al quadrato: cor^2

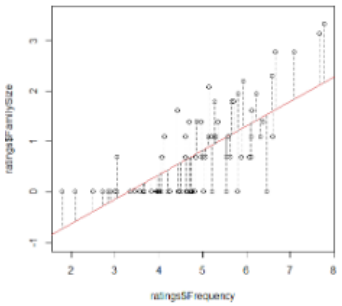
Varianza residua: 1-R^2

Porzione della varianza della variabile dipendente NON “spiegata” dalla variabile dipendente.

Residui:

Distanze tra i valori osservati e i valori predetti dalla retta di regressione

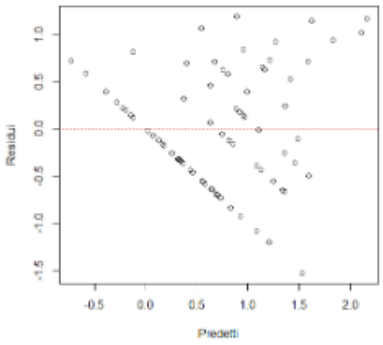
```
> residuals(model)
```



Analisi dei residui:

Rappresentiamo graficamente la distribuzione dei residui contro i valori predetti:

```
> plot(fitted(model), residuals(model), ylab="Residui",
  xlab="Predetti"); abline(h=0, lty=3, col="firebrick3")
```



I punti dovrebbero collocarsi in maniera omogenea (a “nuvola”) intorno ad un retta orizzontale con intercetta 0.

Pattern differenti sono da interpretare come indicazione di una relazione NON lineare.

Variabile Casuale

Una variabile Casuale (detta anche aleatoria, stocastica) è una variabile che può assumere diversi valori in maniera non deterministica.

Al ripetersi di un evento non siamo in grado di predire con assoluta certezza quale valore assumerà la nostra variabile.

Due tipologie:

- **Discreta:** può assumere un numero finito (eg. Lancio dei dadi) oppure un numero infinito ma contabile di modalità (eg. Frequenza di una parola)
- **Continua:** le sue modalità appartengono ad un intervallo di numeri reali limitato o illimitato (es. altezza, velocità)

Probabilità

Possibilità che un evento possa verificarsi. L'evento è la modalità di una variabile.

Definizione frequentista: la probabilità di un evento è uguale alla sua frequenza relativa, dato un numero di prove sufficientemente grande eseguite nelle medesime condizioni.

- La probabilità di un evento è sempre compresa tra 0 e 1.
- 0<= p(x) <= 1, dove 1 = evento impossibile, 1 = evento certo
- La somma delle probabilità di tutti gli eventi possibili è 1.

Distribuzione di probabilità

Data una variabile casuale, è la funzione che associa una valore di probabilità ad ogni sua possibile modalità (valore).

- È il **modello matematico** analogo alla distribuzione di frequenza (costruita invece su rilevazioni campionarie)
- Caratterizzata da una serie di parametri che sono stimati sulla base delle rilevazioni del campione

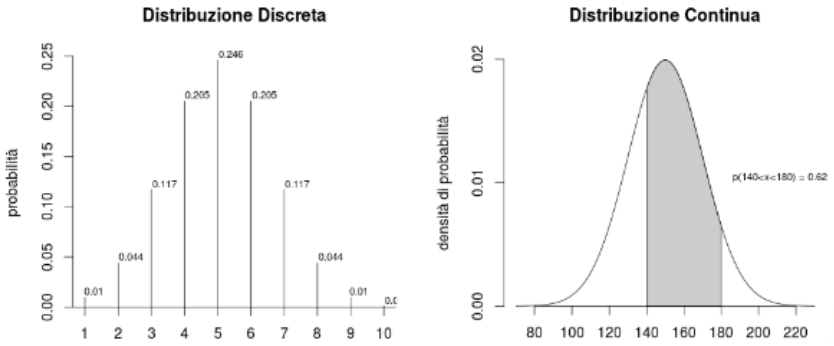
Varie distribuzioni:

```
> ?distribution
```

Molte altre sono in altri pacchetti:

```
http://cran.r-project.org/web/views/Distributions.html
```

Una prima classificazione delle distribuzioni di probabilità è basata sulla natura della scala di misura della variabile di interesse.



Distribuzione Discreta: ad ogni possibile valore della variabile di interesse è associata una probabilità (probability mass function).

La somma delle probabilità delle varie modalità è 1.

Distribuzione continua: descritte da funzioni che hanno come valore la densità di probabilità (probability density function), di cui ci interessa l'integrale.

La probabilità che la variabile assuma un valore appartenente ad un intervallo è pari all'area sottesa alla porzione di curva compresa tra i due estremi dell'intervallo.

Distribuzione Normale

Descrive molti fenomeni fisici e biologici. Usata dai modelli analitici.

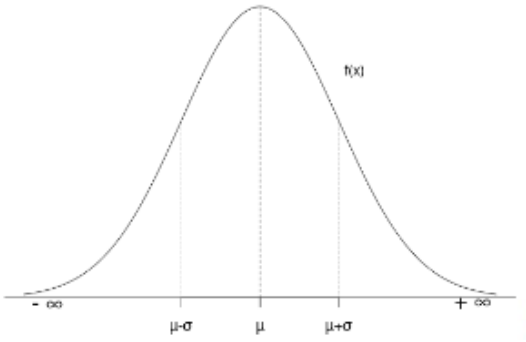
È definita da due parametri: **media** (u) e **deviazione standard** (o). la variabile casuale può assumere valori continui compresi tra [-inf, +inf].

Forma a campana:

- moda, mediana e media coincidono in μ
- simmetrica intorno alla media
- punti di flesso (punti in cui curva passa da concava a convessa) in $\mu \pm \sigma$

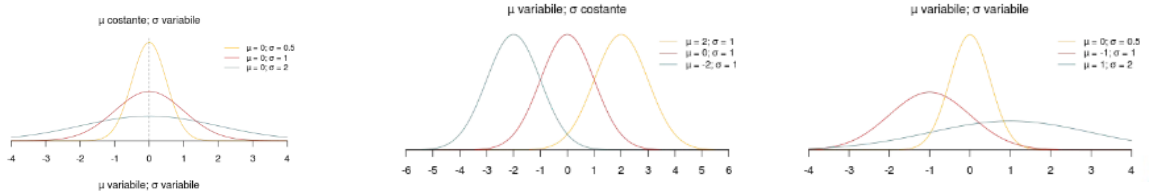
Distribuzione Normale Standard:

- $\mu = 0; \sigma = 1$



Qualsiasi normale può essere standardizzata trasformando i suoi dati in z-scores (sottraendo u e dividendo per o).

N.B. variando media e d.s. si possono ottenere infinite distribuzioni normali:

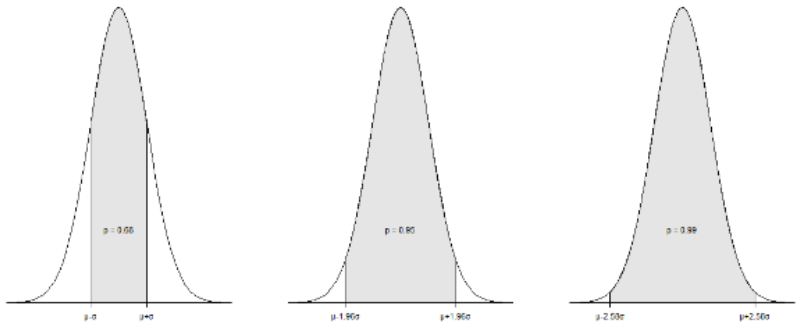


Distribuzione normale in R

Si usa la funzione dnorm():

```
> z <- seq(-4, 4, by = 0.01)
> plot(z, dnorm(z), type = "l", ylab = "density(z)") #
  produce la distribuzione normale standardizzata
> plot(z, dnorm(z, mean = 0.5, sd= 0.8), type = "l", ylab
  = "density(z)") #produce la distribuzione normale con
  media 0.5 e deviazione standard = 0.8
```

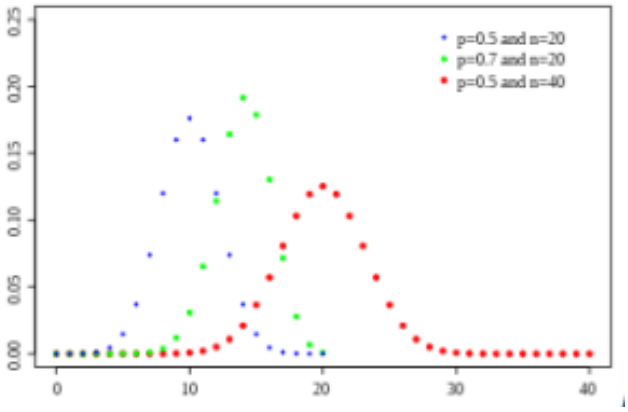
Le aree sottese della curva (probabilità) possono essere ricavate dalla distanza (espressa in deviazione standard) dalla media.



Distribuzione Binomiale

Un modello probabilistico descrive il numero di successi in una serie di prove indipendenti, ciascuna con due possibili esiti (successo o insuccesso). È particolarmente utile quando si analizzano situazioni in cui si eseguono esperimenti ripetuti con condizioni identiche.

Es. lancio di una moneta, estrazione di una parola da un corpus, (successo: la parola estratta è X; fallimento: la parola estratta non è X).



In R:

```
> a <- seq(1:100) #sequenza discreta di 100 elementi,
corrsipondenti ai numeri di successi

> plot(a, dbinom(a,100,0.5), ylab = p(z)") #
distribuzione di probabilità binomiale del numero di
successi su 100 esperimenti con probabilità di successo
0.5
```

Distribuzione X^2

Distribuzione della probabilità della somma dei quadrati di v valori indipendenti standardizzati di una variabile normalmente distribuita con media u e deviazione standard o.

$$X_\nu^2 = \sum_{i=1}^{\nu} \frac{(x_i - \mu)^2}{\sigma^2} = \sum_{i=1}^{\nu} z_i^2$$

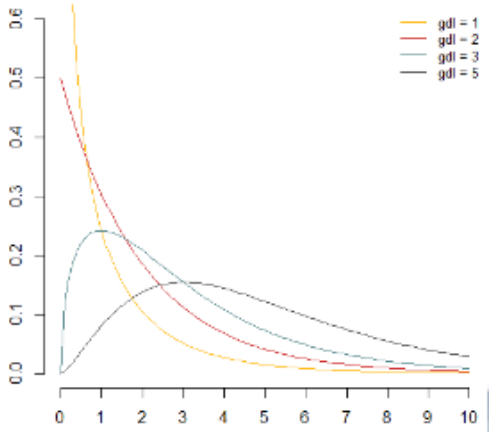
Definita su valori positivi $[0, +\infty]$

Asimmetria positiva

La sua forma dipende dal numero ν di valori dipendenti che generano la distribuzione (i suoi gradi di libertà)

Approssima una distribuzione normale all'aumentare dei gradi di libertà

Utile per dati categorici



Studio quantitativo

1- **Ipotesi nulla (H0):** l'ipotesi di cui si stima la validità. **Tutte le differenze osservate nell'esperimento sono dovute al caso**.
2- **Ipotesi alternativa (H1):** ipotesi che viene accettata quando si può ragionevolmente dubitare della validità dell'ipotesi nulla.

è per accettare l'ipotesi alternativa che si decide di fare un esperimento.

Verifica delle ipotesi

1) Si cerca di prevedere la distribuzione dei dati assumendo che l'ipotesi nulla sia vera (ovvero: se l'ipotesi nulla fosse vera, quale distribuzione dei dati osserveremmo?)
2) Se i dati effettivamente osservati nel nostro esperimento (campione) sono molto distanti da quelli previsti, questa differenza non può essere imputata al caso, per cui l'ipotesi nulla è rifiutata e l'ipotesi alternativa accettata.
3) Se i dati osservati non sono sufficientemente distanti da quelli previsti, questa differenza può essere imputata al caso, motivo per cui non possiamo scartare l'ipotesi nulla.

L'ipotesi nulla non è mai accettata dal fatto che i dati non siano sufficientemente distanti da quelli previsti, possiamo concludere solo che essi non sono sufficiente per scartare l'ipotesi nulla.

	Non Rifiuto H_0	**Rifiuto H_0**
H_0 Vera	vero positivo	Errore di I tipo (falso positivo)
H_0 Falsa	Errore di II tipo (falso negativo)	vero negativo

Errore di I tipo: pensiamo che i dati supportino l'ipotesi di ricerca H1 quando in realtà è falso.

Errore di II tipo: pensiamo che i dati non supportino l'ipotesi di ricerca H1 quando in realtà è vero.

Un risultato sperimentale è detto statisticamente significativa se ci consente di rifiutare l'ipotesi nulla H0.

Livello di significativà

Regola decisionale che ci serve 'per decidere se rifiutare o meno l'ipotesi nulla: **tanto è più basso alpha, tanto è più improbabile che l'ipotesi nulla sia vera.**

Corrisponde alla probabilità di rifiutare erroneamente H0 e quindi accettare H1 quando H0 è vera (= avere un falso positivo)

Per convenzione sono assunte le seguenti soglie di significatività e generalmente alpha viene stabilito prima di raccogliere i dati sul campione:

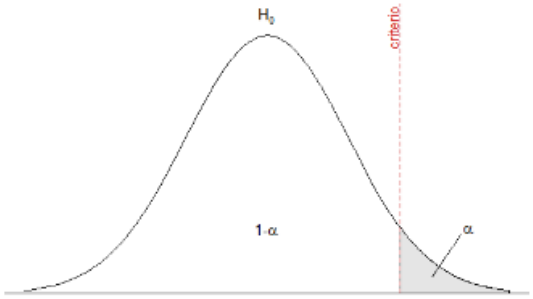
- $\alpha = 0.05$: il rischio di sbagliare rifiutando H_0 è pari al 5%
- $\alpha = 0.01$: il rischio di sbagliare rifiutando H_0 è pari al 1%
- $\alpha = 0.001$: il rischio di sbagliare rifiutando H_0 è pari allo 0.1%

	Non Rifiuto H_0	**Rifiuto H_0**
H_0 Vera	$1 - \alpha$	Livello di Significatività: α
H_0 Falsa	β	Potenza: $1 - \beta$

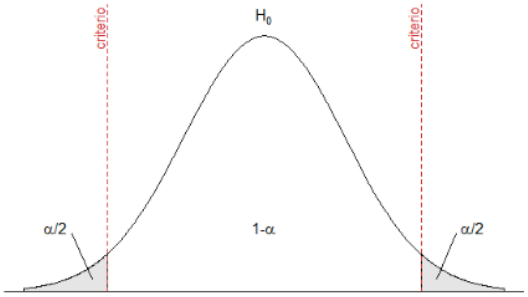
Per il teorema del limite centrale le medie di campioni di una popolazione hanno una distribuzione normale.

- La regione di rifiuto stabilisce quanto lontani, per poter smentire l'ipotesi H0, i dati del nostro campione devono essere da quelli di campioni di una popolazione in cui fosse vera H0
- Se i dati del nostro campione (es. la media) cadono nella regione di rifiuto, allora possiamo rifiutare H0, con probabilità di errore inferiore ad alpha.

Regione di rifiuto di H0 (H1 monodirezionale):



Regione di rifiuto di H0 (H1 bidirezionale):



p-value

Valore ricavato dall'applicazione di un test statistico ad un campione sperimentale.
Alla luce del nostro campione, ci indica la probabilità di errore associata al rifiuto di H0 sulla base dei nostri dati.

Se il p-value è minore del livello di significatività prescelto (0.05, 0.01, 0,001) il risultato sperimentale (es. la differenza tra i tempi di reazione tra gruppi di soggetti) **si dice statisticamente significativo e si accetta H1**.

Test statistici

- **Parametrici:**
  - Richiedono il verificarsi di alcune condizioni di applicabilità: distribuzione dei dati, omogeneità delle varianze, numerosità del campione
  - Maggior potere statistico (maggior probabilità di scartare H0 quando questa è falsa)
- **Non Parametrici:**
  - Maggiore flessibilità
    - Prescindono dai parametri della distribuzione
    - Applicabili anche a campioni di numerosità più ridotta
    - Applicabili anche a variabile non numeriche (ma ordinabili)
  - Minore potere statistico (perdita che diminuisce all'aumentare della popolazione campionaria).

Test di Shapiro-Wilks

Test statistico utilizzato per verificare se un campione di dati segue una distribuzione normale.

Test di Shapiro-Wilks

```
> shapiro.test(tOntLength)
```

Testiamo l'assunzione che il nostro campione sia estratto da una distribuzione normale (H_0)

(nel nostro caso, il *p-value* inferiore a 0.05 ci fa rigettare l'ipotesi di normalità)

```
Shapiro-Wilk normality test

data:  tOntLength
W = 0.9248, p-value = 2.145e-05
```

Interpretazione dei Risultati

- **Ipotesi Nulla (H0)**: I dati provengono da una distribuzione normale.
- **Ipotesi Alternativa (H1)**: I dati non provengono da una distribuzione normale.

Se il valore p è inferiore a un livello di significatività predefinito (ad esempio 0.05), si rifiuta l'ipotesi nulla, indicando che i dati non seguono una distribuzione normale.

Spiegazione nel compito:

Per verificare l'esistenza di una distribuzione normale posso usare il test di Shapiro-Wilk, considerando come ipotesi nulla (H_0) che la distribuzione del campione è normale.

Quindi, considerando un certo livello di significatività (pari a 0.05), se il p-value è minore posso respingere l'ipotesi nulla di normalità e accettare l'ipotesi alternativa (H_1), secondo cui la distribuzione seguita non è normale; altrimenti, se il p-value è maggiore, non avrò dati sufficienti per scartare l'ipotesi nulla.

Prima di applicare Shapiro-Wilks si può creare un grafico ad istogramma con la linea tracciata per vedere se indicativamente si può avere una distribuzione normale per i dati:

```
hist(df2$REACTTIME, freq = F)
lines(density(df2$REACTTIME), col = "red")
```

Se la linea che si viene a creare approssima una distribuzione normale si procede con il livello di significativà.

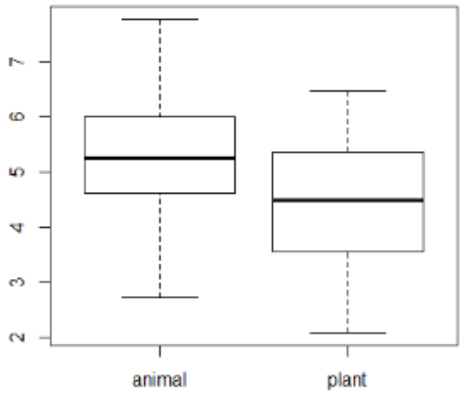
Test per due campioni indipendenti

Dati: rating di parole morfologicamente semplici

```
> simplex <-
  ratings[ratings$Complex ==
  "simplex",]
```

Confrontiamo la distribuzione delle frequenze delle diverse classi:

```
> boxplot(Frequency ~ Class,
  data = simplex)
```



Si deve controllare se la distribuzione è normale:

Se i dati sono **distribuiti normalmente**, possiamo rispondere a questa domanda tramite un **test t**

NB: per verificare se i dati sono distribuiti normalmente applicare prima il test Shapiro – Wilks!!!

```
> animalsFreqs <-
  simplex[simplex$Class ==
  "animal",]$Frequency

> plantsFreqs <-
  simplex[simplex$Class ==
  "plant",]$Frequency

> t.test(animalsFreqs,
  plantsFreqs)
```

```
Welch Two Sample t-test

data:  animalsFreqs and
plantsFreqs
t = 2.674, df = 57.545, p-value
= 0.009739
alternative hypothesis: true
difference in means is not
equal to 0
95 percent confidence interval:
 0.193183 1.344315
sample estimates:
mean of x mean of y
 5.208494  4.439745
```

N.B. il t-test è parametrico, quindi per dati non parametrici si deve usare altro:

Il test t non può essere usato per distribuzioni non normali (è un test parametrico). In questi casi, si deve optare un test non parametrico:

Test U di Wilcoxon - Mann-Whitney

```
> wilcox.test(tOntLength,
  nOntLength)
```

```
Wilcoxon rank sum test with
continuity correction

data:  tOntLength and
nOntLength
W = 3744.5, p-value = 0.0005335
alternative hypothesis: true
location shift is not equal to
0
```

Interpretazione risultati:

Importante: quando si da una spiegazione sui test effettuati è buona norma citare sempre il tipo di test usato, il valore calcolato, il p-value e i df (degrees of freedom) se presenti.

Test X^2

Viene spesso utilizzato per determinare se esiste una relazione tra due variabili categoriali (**test di indipendenza**) oppure usato per determinare se le frequenze osservate differiscono da quelle attese (**test di bontà di adattamento**).

Ci permette di verificare se la distribuzione dei valori di una variabile X è dipendente dai valori di un'altra variabile Y: misura il grado di associazione tra due variabili categoriali.

Procedura:

1. **Formulazione delle ipotesi**:
    - **H0**: Le due variabili sono indipendenti.
    - **H1**: Le due variabili non sono indipendenti.
2. **Applicazione del teorema:**

Test X^2

```
> chisq.test(ct)
```

(nel nostro caso, il *p-value* inferiore a 0.001 ci fa rigettare l'ipotesi di indipendenza)

```
> summary(ct)
```

```
Pearson's Chi-squared test

data:  ct
X-squared = 414.8204, df = 12, p-value < 2.2e-16
```

Quali celle della tabella di contingenza contribuiscono maggiormente al valore del X^2?

Possiamo rispondere a questa domanda analizzando i residui di Pearson:

N.B. questo test viene utilizzato per sapere se esiste un'associazione tra due variabili categoriali

In questo caso devono essere passate le bivariate

```
> chisq.test(ct)$residuals
```

- Interpretazione: residui il cui valore assoluto è maggiore di 4 ($\alpha \cong 0.0001$) sono considerati altamente significativi, mentre residui il cui valore assoluto è compreso tra 2 ($\alpha \cong 0.05$) e 4 sono da interpretarsi come mediamente significativi.

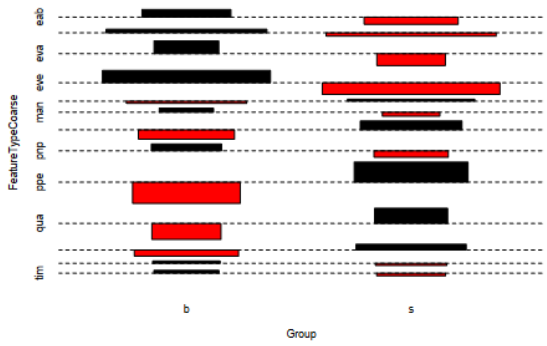
Questa informazione può essere rappresentata in un mosaic plot:

```
> mosaic(ct, shade = T, rot_labels = c(top=90, left=0),
  offset_varnames = c(top=1.5, left=1)) # richiede "vcd"
```

Association Plot

Mostra la significatività dei residui di Pearson.

```
> assocplot(ct)
```



Mosaic Plot:

Questa informazione può essere anche rappresentata in un mosaic plot:

```
> mosaic(ct, shade = T, rot_labels = c(top=90, left=0),
  offset_varnames = c(top=1.5, left=1)) # richiede "vcd"
```
