Psicometria 1
PS 123

Laboratorio 1

Corrado Caudek

Dipartimento di Psicologia, Università di Trieste

Sommario
Iniziare e chiudere una sessione di **R**
Semplice aritmetica

Sommario

Sommario

Sommario
Iniziare e chiudere una sessione di **R**
Semplice aritmetica

Introduzione a R

- Iniziare e chiudere una sessione di **R**
- Semplice aritmetica
- Assegnazioni di valori
- Funzioni
- Gestione di vettori
- Matrici
- Data-frames

Iniziare e chiudere una sessione di R

Sommario
Iniziare e chiudere una sessione di **R**
Semplice aritmetica

- Per iniziare una sessione **R** fare un doppio click di mouse sulla icona di **R**.
- Per uscire da **R**, usa `q()`.
  - Per salvare i dati rispondere "Si", altrimenti rispondere "No".
- Per controllare cosa c'è disponibile nella directory dei dati:

```
> ls()
character(0)
```

- Per eliminare un oggetto, usa `rm()`:

```
> rm(qualcosa)
> NULL Warning message: remove:
  variable "qualcosa" was not found
```


Iniziare e chiudere una sessione di R

Sommario
Iniziare e chiudere una sessione di **R**
Semplice aritmetica

- Quando si inizia una nuova sessione di lavoro, è opportuno rimuovere tutti i vecchi oggetti che non servono.
- Un comando utile è:

```
> rm(list=ls())
```


Semplice aritmetica

Sommario
Iniziare e chiudere una sessione di **R**
Semplice aritmetica

In **R**, qualunque cosa venga scritta al prompt viene valutata:

```
> 2-3
[1] -1
> 2/3
[1] 0.6666667
> 2^3
[1] 8
> 4^2-3*2
[1] 10
> (4^2)-(3*2)
[1] 10
> 2^-3
[1] 0.125
> -2 - -3
[1] 1
```

Assegnazione di valori

Assegnazione di valori

Si può salvare un valore assegnandolo ad un oggetto mediante il simbolo "<−" :

```
> a <- 2
> a
[1] 2
> x <- c(1,2,3,4)
> x
[1] 1 2 3 4
> x/2
[1] 0.5 1.0 1.5 2.0
> a*x
[1] 2 4 6 8
```


Funzioni

Funzioni

R fornisce tutte le funzioni che si trovano su un calcolatore tascabile:

```
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    1.75    2.50    2.50    3.25    4.00
> length(x)
[1] 4
> sum(x)
[1] 10
> mean(x)
[1] 2.5
> sum(x) / length(x)
[1] 2.5
```


Funzioni

```
> vx <- var(x)
> vx
[1] 1.666667
> sum( (x-mean(x))^2 ) / (length(x)-1)
[1] 1.666667
> sd(x)
[1] 1.290994
> sqrt( var(x) )
[1] 1.290994
```


Funzioni

Ci sono ovviamente svariate funzioni dedicate alle applicazioni statistiche:

```
> x <- rnorm(20)
> x
 [1]  0.31394395  0.79604327 -1.31206944 -0.66132400
 [5]  0.05783983 -1.11801266  0.21334015 -1.63824212
 [9]  0.16165792 -0.76111798 -0.43216836  1.50071740
[13] -0.22098959  0.92038811 -1.53083690  1.43683846
[17] -0.55800446 -1.18331642 -0.08883057 -1.53580469
>
> mean(x)
[1] -0.2819974
>
> var(x)
[1] 0.932094
```

Funzioni


```
> pnorm(1.96)
[1] 0.9750021
> qnorm(.975)
[1] 1.959964
>
> qt(.975, 10)
[1] 2.228139
```


Vettori

Vettori

Per creare un vettore, si usa la funzione `c()`:

```
> y <- c(1,2,3,4)
> y
[1] 1 2 3 4
```


Gestione di vettori

Si possono usare diverse funzioni per creare vettori che sono sequenze di numeri:

```
> 1:4
[1] 1 2 3 4
> 4:1
[1] 4 3 2 1
> -1:2
[1] -1  0  1  2
> seq(1,4)
[1] 1 2 3 4
> seq(2, 8, by=2)
[1] 2 4 6 8
> seq(0, 1, by=.1)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```


Gestione di vettori

Ai vettori può essere applicata la stessa aritmetica di base che è stata applicata ai valori scalari:

```
> c(1,2,3,4)/2
[1] 0.5 1.0 1.5 2.0
```


Gestione di vettori

Alcune funzioni utili per la manipolazione di vettori:

```
> x <- 3:9
> x
[1] 3 4 5 6 7 8 9
> length(x)
[1] 7
> max(x)
[1] 9
> min(x)
[1] 3
> sum(x)
[1] 42
> mean(x)
[1] 6
```

Estrazione degli elementi da un vettore

Gli elementi di un vettore possono essere estratti usando le parentesi quadre `[]`:

```
> x <- runif(20)*100
> x
 [1] 67.051835 44.755383 40.950784 97.523297  2.493610
 [6] 89.667196 45.526138  7.022685 65.994809 69.390573
[11] 38.794398 81.969528 45.877838 37.792476 72.612885
[16] 65.628103  8.744988 80.700821 89.018100 14.954435
> x[7]
[1] 45.52614
> x[11:20]
 [1] 38.794398 81.969528 45.877838 37.792476 72.612885
 [6] 65.628103  8.744988 80.700821 89.018100 14.954435
> x[-(11:20)]
 [1] 67.051835 44.755383 40.950784 97.523297  2.493610
 [6] 89.667196 45.526138  7.022685 65.994809 69.390573
```


Matrici

R consente anche di usare le matrici:

```
> x <- matrix(c(
+      2,3,
+      5,7,
+      11,15
+        ),ncol=2, byrow=T)
>
> x
     [,1] [,2]
[1,]    2    3
[2,]    5    7
[3,]   11   15
```


NB: Bisogna specificare `nrow` o `ncol` per comunicare a **R** la dimensione della matrice; `byrow=T` significa che la matrice è riempita seguendo le righe.

Per estrarre da una matrice un elemento, bisogna specificarne le due coordinate:

```
> x[3,2]
[1] 15
```


- Se non si mette una delle coordinate, si ottiene una intera riga/colonna:

```
> x[, 1]
[1]  2  5 11
>
> x[3, ]
[1] 11 15
```

- La funzione `dim()` indica la dimensione (numero di righe e numero di colonne) della matrice:

```
> dim(x)
[1] 3 2
```


Data-frames

Data-frames

- Un *data frame* è un oggetto simile ad una matrice, ma usato per rappresentare dati sperimentali. Ogni riga rappresenta una unità statistica, ogni colonna rappresenta una variabile misurata sulle unità statistiche. Le colonne possono contenere variabili numeriche o categoriali.
- Per leggere un insieme di dati di questo tipo si usa la funzione `read.table()`, che automaticamente controlla se le variabili sono numeriche o qualitative, se le righe e/o le colonne hanno etichette.

Data-frames

Supponete che il file `Duncan.txt` sia così costituito:

```
occupation               education income women prestige census type
GOV.ADMINISTRATORS           13.11  12351 11.16     68.8   1113 prof
GENERAL.MANAGERS             12.26  25879  4.02     69.1   1130 prof
ACCOUNTANTS                  12.77   9271 15.70     63.4   1171 prof
PURCHASING.OFFICERS          11.42   8865  9.11     56.8   1175 prof
CHEMISTS                     14.62   8403 11.68     73.5   2111 prof
PHYSICISTS                   15.64  11030  5.13     77.6   2113 prof
BIOLOGISTS                   15.09   8258 25.65     72.6   2133 prof
...
```

Si noti che la prima riga contiene il nome delle variabili.

Data-frames

- Possiamo acquisirlo con il comando:

```
> duncan <- read.table("Duncan.txt", header=TRUE)
```

```
> duncan[1:5, ]
           occupation education income women prestige census type
1  GOV.ADMINISTRATORS     13.11  12351 11.16     68.8   1113 prof
2    GENERAL.MANAGERS     12.26  25879  4.02     69.1   1130 prof
3         ACCOUNTANTS     12.77   9271 15.70     63.4   1171 prof
4 PURCHASING.OFFICERS     11.42   8865  9.11     56.8   1175 prof
5            CHEMISTS     14.62   8403 11.68     73.5   2111 prof
```

- Il *data frame* è anche una matrice

```
> dim(duncan)
[1] 102   4
```

- I nomi delle variabili sono

```
> names(duncan)
[1] "education" "income"    "prestige"  "type"
```


Data-frames

Per accedere ad una variabile contenuta in un data frame usiamo la notazione "`dataframe$variabile`":

```
> duncan$income
  [1] 12351 25879  9271  8865  8403 11030  8258 14163
  [9] 11377 11023  5902  7059  8425  8049  7405  6336
 [17] 19263  6112  9593  4686 12480  5648  8034 25308
 [25] 14558 17498  4614  3485  5092 10432  5180  6197
 [33]  7562  8206  4036  3148  4348  2448  4330  4761
 [41]  3016  2901  5511  3739  3161  4741  5052  6259
 [49]  4075  7482  8780  2594   918  2370  8131  6992
 [57]  7956  8895  8891  3116  3930  7869   611  3000
 [65]  3472  3582  3643  1656  6860  4199  5134  5134
 [73]  1890  4443  3485  8043  6686  6565  6477  5811
 [81]  6573  3942  5449  2847  5795  7716  4696  8316
 [89]  7147  8880  5299  5959  4549  6928  3910 14032
 [97]  8845  5562  4224  4753  6462  3617
```


Data-frames

- Utilizziamo il comando `attach()` per comunicare ad **R** che le operazioni che faremo si riferiscono al data-frame indicato.
- In questo modo è possibile accedere direttamente alle variabili contenute nel data-frame:

```
> attach(duncan)
> mean(income)
[1] 6797.902
```


Data-frames

Per avere delle statistiche di base sulle variabili contenute in `duncan` possiamo usare la funzione `summary()`:

```
> summary(duncan)

   education          income          prestige         type
 Min.   : 6.380   Min.   :  611   Min.   :14.80   <NA>: 4
 1st Qu.: 8.445   1st Qu.: 4106   1st Qu.:35.23   bc  :44
 Median :10.540   Median : 5931   Median :43.60   prof:31
 Mean   :10.738   Mean   : 6798   Mean   :46.83   wc  :23
 3rd Qu.:12.648   3rd Qu.: 8187   3rd Qu.:59.27
 Max.   :15.970   Max.   :25879   Max.   :87.20
```

Leggere i dati nelle librerie R

Molte librerie **R** contengono files di dati. In **R** è necessario usare la funzione `data()` per leggere una data-frame contenuto in un package che è stato preliminarmente caricato. Per esempio

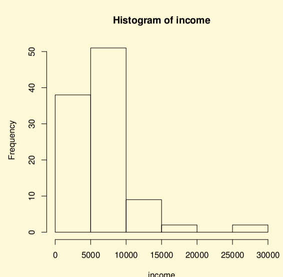
```
> library(car)
> data(Duncan)
> Duncan[1:5, ]
           type income education prestige
accountant prof     62        86       82
pilot      prof     72        76       83
architect  prof     75        92       90
author     prof     55        90       76
chemist    prof     64        86       90
```


Rappresentazioni grafiche

Rappresentazioni grafiche

Possiamo anche rappresentare graficiamente la distribuzione di una variabile ad esempio `income`, mediante un istogramma

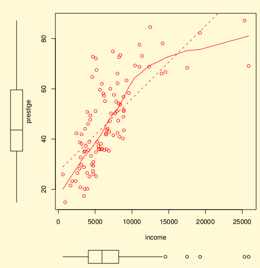
```
> hist(income)
```



Rappresentazioni grafiche

Molteplici rappresentazioni grafiche dei dati sono possibili

```
> library(car)
> scatterplot(income, prestige)
```



Funzione di help

Funzione di help

La funzione di help per i comandi e i data-frame è

```
> ?Duncan

Duncan                  package:car                  R
Documentation

Duncan's Occupational Prestige Data

Description:

     The 'Duncan' data frame has 45 rows and 4 columns.
     Data on the prestige and other characteristics of
     45 U. S. occupations in 1950.

Usage:

     data(Duncan)
```

Funzione di help

```
Format:

     This data frame contains the following columns:

     type Type of occupation. A factor with the following
          levels: 'prof', professional and managerial; 'wc',
          white-collar; 'bc', blue-collar.

     income Percent of males in occupation earning $3500 or
          more in 1950.

     education Percent of males in occupation in 1950 who
          were high-school graduates.

     prestige Percent of raters in NORC study rating
          occupation as excellent or good in prestige.
```


Distribuzioni

Distribuzioni

R consente di gestire automaticamente molte distribuzioni (per calcolare probabilità, quantili, ...). Questo permette di effettuare verifiche di ipotesi, calcolare intervalli di confidenza, ecc.

Ad esempio, consideriamo la distribuzione normale standardizzata. Esistono 4 funzioni ad essa relative:

- `dnorm(x)` calcola il valore della densità in `<x>`;
- `pnorm(q)` calcola il valore della ripartizione in `<q>`;
- `qnorm(p)` calcola il quantile di livello `<p>`;
- `rnorm(n)` genera un campione da una normale standard di dimensione `<n>`.

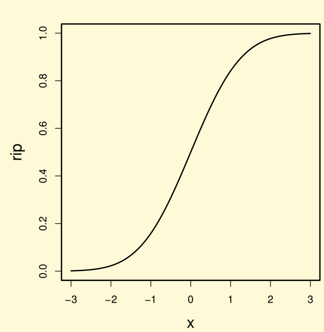
Distribuzioni

Per esempio, per vedere l'andamento della funzione di ripartizione di una normale standard:

```
> x <- seq(-3, 3, length=100)
> rip <- pnorm(x)
> plot(x, rip, type="l")
```

Alcune funzioni sono `chisq`, `f`, `norm`, `t`, `unif`.

Distribuzioni

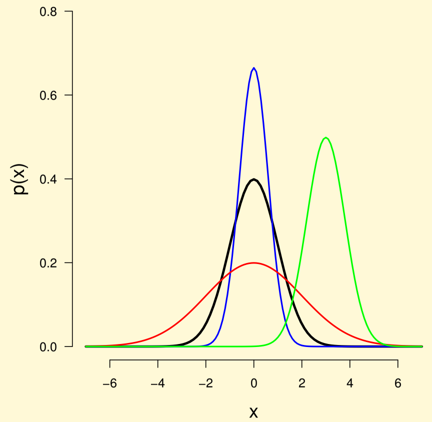


Distribuzioni

Con la funzione `dnorm(x, mean=0, sd=1)` possiamo disegnare la funzione normale con parametri `mean` e `sd`.

```
> plot(dnorm, from = -7, to = 7, ylim = c(0,0.8),
+   las = 1, col = "black", xlab="x", ylab="p(x)",
+   lwd=3, bty="n", cex=1.5)
> x <- seq(-7, 7, 0.1)
> normdensity1 <- dnorm(x, mean = 0, sd = 0.6)
> normdensity2 <- dnorm(x, mean = 0, sd = 2)
> normdensity3 <- dnorm(x, mean = 3, sd = 0.8)
> lines(x, normdensity1, col = "blue")
> lines(x, normdensity2, col = "red")
> lines(x, normdensity3, col = "green")
```

Distribuzioni

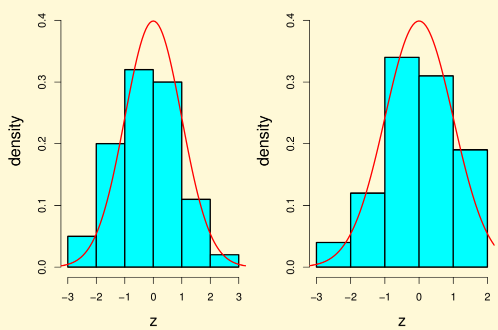


Distribuzioni

Con la funzione `rnorm(n, mean=0, sd=1)` possiamo estrarre un campione casuale di numerosità `n` da una distribuzionr normale con parametri `mean` e `sd`.

```
> library(MASS)
>
> par(mfrow=c(1,2))
> truehist(rnorm(100), ylim=c(0, .41), xlab=c("z"),
+   ylab=c("density"))
> curve(dnorm(x, 0, 1), add=TRUE, col="red")
> truehist(rnorm(100), ylim=c(0, .41), xlab=c("z"),
+   ylab=c("density"))
> curve(dnorm(x, 0, 1), add=TRUE, col="red")
```


Distribuzioni



Distribuzioni

- Con la funzione `qnorm(p, mean=0, sd=1)` troviamo il quantile della normale standardizzata tale per cui $P(q < z < \infty) = p$:

```
> qnorm(.975)
[1] 1.9600
> pnorm(1.96)
[1] 0.975
```

- Le stesse operazioni si possono eseguire per altre distribuzioni. Per la t di Student con 12 gradi di libertà, per esempio, avremo

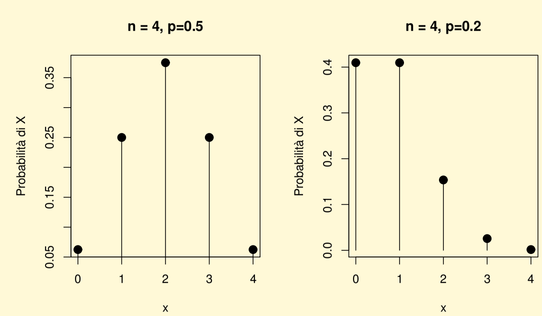
```
> qt(.975, 12)
[1] 2.1788
> pt( 2.1788, 12)
[1] 0.975
```


Distribuzioni

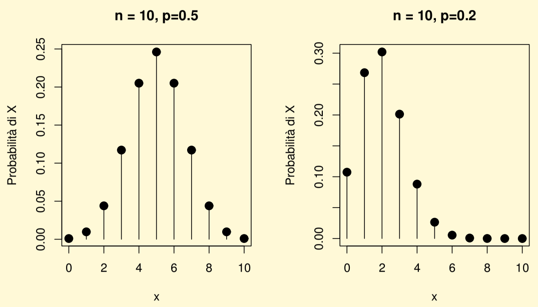
Con la funzione `dbinom()` possiamo rappresentare la funzione binomiale:

```
> par(mfrow = c(1, 2))
> x<-0:4
> plot(x,dbinom(0:4,4,.5), type="h", ylab="Probabilit{\'a} di X",
+   main="n = 4, p=0.5")
> points(x,dbinom(0:4,4,.5), pch=19, cex=1.5)
> x<-0:4
> plot(x,dbinom(0:4,4,.2), type="h", ylab="Probabilit{\'a} di X",
+   main="n = 4, p=0.2")
> points(x,dbinom(0:4,4,.2), pch=19, cex=1.5)
>
> par(mfrow = c(1, 2))
> x<-0:10
> plot(x,dbinom(0:10,10,.5), type="h", ylab="Probabilit{\'a} di X",
+   main="n = 10, p=0.5")
> points(x,dbinom(0:10,10,.5), pch=19, cex=1.5)
> x<-0:10
> plot(x,dbinom(0:10,10,.2), type="h", ylab="Probabilit{\'a} di X",
+   main="n = 10, p=0.2")
> points(x,dbinom(0:10,10,.2), pch=19, cex=1.5)
```


Distribuzioni



Distribuzioni



Distribuzioni
Adattamento ad una distribuzione

Adattamento ad una distribuzione

- Consideriamo un campione generato da una distribuzione normale standard di ampiezza 10:

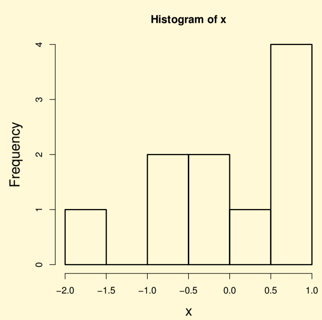
```
> x <- rnorm(10)
```

- Supponiamo di non sapere che il campione proviene da una popolazione normale. Proviamo a studiare graficamente la distribuzione dei dati per capire da che popolazione deriva.

```
> hist(x)
```

Distribuzioni
Adattamento ad una distribuzione

Adattamento ad una distribuzione

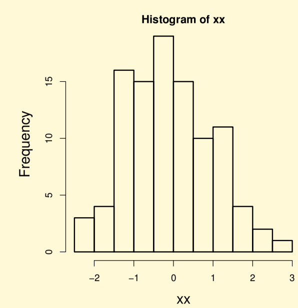


Distribuzioni
Adattamento ad una distribuzione

Adattamento ad una distribuzione

Aumentiamo la numerosità:

```
> xx <- rnorm(100)
> hist(xx)
```



Distribuzioni
Adattamento ad una distribuzione

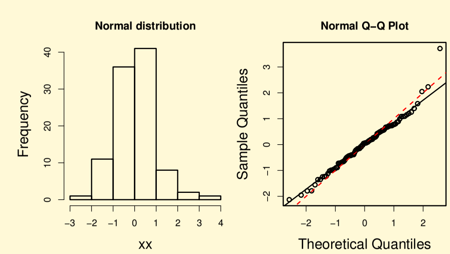
Adattamento ad una distribuzione

- Che strumenti grafici abbiamo a disposizione per vedere se proviene o meno da una distribuzione normale?
- Per verificare l'adattamento dei dati ad una distribuzione normale, può essere usato il comando `qqnorm()`.

```
> hist(xx, main="Normal distribution")
> qqnorm(xx)
> par(pty="s")
> abline(0, 1, col="red", lty=2)
> qqline(xx)
```

Distribuzioni
Adattamento ad una distribuzione

Adattamento ad una distribuzione



Distribuzioni
Adattamento ad una distribuzione

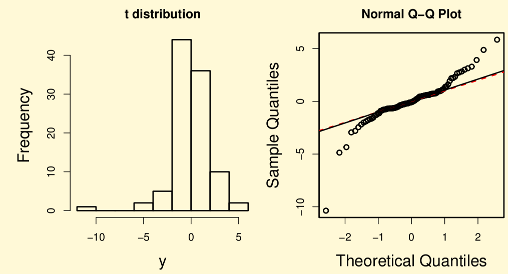
Adattamento ad una distribuzione

Distribuzioni
Adattamento ad una distribuzione

```
> par(mfrow=c(1,2))
> y <- rt(100,2)
> hist(y, main="t distribution")
> qqnorm(y)
> abline(0, 1, col="red", lty=2)
> qqline(y)
>
> par(mfrow=c(1,2))
> z <- rexp(100)
> hist(z, main="Exponential distribution")
> qqnorm(z)
> abline(0, 1, col="red", lty=2)
> qqline(z)
```

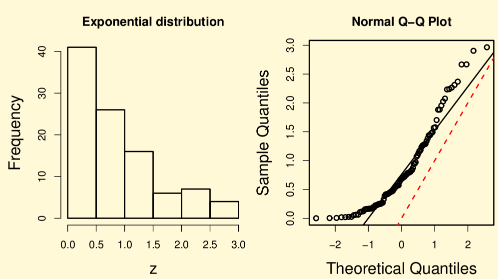

Adattamento ad una distribuzione

Distribuzioni
Adattamento ad una distribuzione



Adattamento ad una distribuzione

Distribuzioni
Adattamento ad una distribuzione



Sommatorie

Sommatorie

Sommatorie

Sommatorie

- Consideriamo la seguente distribuzione della variabile X: {3, 7, 1, 9, 2}.
- Usiamo l'indice *i* per fare riferimento alle diverse u.s. che costituiscono la distribuzione.
- Il valore assunto dalla variabile X nel caso di una generica unità statistica viene indicato con X_i .
- Quando l'indice i assume il valore 2, 3, o 5 questo significa che facciamo riferimento al valore assunto da X nel caso della seconda, terza o quinta unità statistica.
- Esempio: $X_1 = 3$; $X_4 = 9$; $X_2 = 7$.

Sommatorie

Sommatorie

- Un simbolo di cui viene fatto un grande uso nella statistica è quello di sommatoria:

$$\sum_{i=1}^{n} X_i = X_1 + X_2 + \cdots + X_n$$

- Nel caso della distribuzione precedente avremo:

$$\sum_{i=1}^{5} X_i = 3 + 7 + 1 + 9 + 2 = 22$$

- Quando non ci sono ambiguità, la notazione precedente può essere semplificata con

$$\sum_i X_i \quad \text{oppure} \quad \sum X_i$$

Sommatorie

- Un caso più complicato si ha quando le singole osservazioni fanno parte di più gruppi.
- Per esempio, assumiamo che vi siano 10 studenti in 3 aule. Per distinguere tutte queste osservazioni abbiamo bisogno di 2 indici, uno per gli studenti e uno per le aule. Sia i l'indice per gli studenti e j l'indice per le aule. Quindi l'indice i assume valori da 1 a 10 e l'indice j da 1 a 3.
- Ciascuno studente sarà dunque identificato da due indici: X_{ij}. Se usiamo la convenzione per cui l'indice i precede l'indice j, allora $X_{7,3}$ indicacherà il settimo studente nella terza aula.

Sommatorie

- Spesso accade che vogliamo sommare i punteggi di tutti gli individui in tutti i gruppi. Per fare questo usiamo la seguente notazione:

$$\sum_{j=1}^{J}\sum_{i=1}^{I} X_{ij}$$

- Nel caso dell'esempio precedente, avremo

$$\sum_{j=1}^{3}\sum_{i=1}^{10} X_{ij} = X_{1,1} + X_{1,2} + \cdots + X_{3,10}$$

Regole per la manipolazione delle sommatorie

Regola 1

- Se a è una costante, allora

$$\sum_{i=1}^{n} a = n \times a$$

- Esempio: se a = 10 e $n = 3$ allora

$$\sum_{i=1}^{3} 10 = 10 + 10 + 10 = 3 \times 10$$

Regola 2

- Se Se ciascuna delle osservazioni che entrano nella sommatoria viene moltiplicata per la costante a, allora

$$\sum_{i=1}^{n} aX_i = a\sum_{i=1}^{n} X_i$$

- Esempio. X = {3, 5, 1} e a = 2. Dunque,

$$\sum_{i=1}^{n} aX_i = 2\times3+2\times5+2\times1 = 18 = 2\times(3+5+1)$$

Regola 3

- Se dobbiamo eseguire un'operazione algebrica (quadrato, radice, logaritmo, ecc.) sulle singole osservazioni che devono essere sommate, questa operazione deve essere eseguita prima di sommare le n osservazioni.
- Esempio:

$$\sum_{i=1}^{n} X_i^2 = X_1^2 + X_2^2 + \cdots + X_n^2$$

Regola 4

- Se la sola operazione che deve essere eseguita sulle n osservazioni prima della sommatoria è essa stessa una somma (o sottrazione), allora la sommatoria può essere distribuita.
- Esempio:

$$\sum_{i=1}^{n} X_i - Y_i = \sum_{i=1}^{n} X_i - \sum_{i=1}^{n} Y_i$$

- Esempio:

$$\sum_{i=1}^{7}(X_i^2 + 4X_i - 10) = \sum_{i=1}^{7} X_i^2 + 4\sum_{i=1}^{7} X_i - 7 \times 10$$

Regola 5

- Se ciascuna osservazione viene misurata su due variabili, X_i e Y_i, allora

$$\sum_{i=1}^{n} X_i Y_i = X_1Y_1 + X_2Y_2 + \ldots X_nY_n$$

in altre parole, dobbiamo eseguire prima il prodotto tra i punteggi appaiati e poi la sommatoria.

Regola 6

$$\sum_{i=1}^{n} aX_iY_i = a\sum_{i=1}^{n} X_iY_i$$

Regola 7

$$\sum_{i=1}^{n}(aX_i + bY_i) = a\sum_{i=1}^{n} X_i + b\sum_{i=1}^{n} Y_i$$

Esempio

Esercizio

La media della variabile X è

$$\bar{X} = \sum_{i=1}^{n} X_i/n$$

Si dimostri che la somma degli scarti di ciascun valore X_i dalla media $\bar{X}$ è uguale a zero.

$$\sum_{i=1}^{n}(X_i - \bar{X}) = \sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \bar{X} = n\bar{X} - n\bar{X} = 0$$

Esercizio

Esercizio

Si semplifichi la seguente espressione:

$$\sum_{i=1}^{n}\left(X_i - \left(\sum_{i=1}^{n} X_i / n\right)\right)$$