Laboratorio di Statistica e Analisi Dati: Lezione 1

Tommaso C. & Marco G.

26 - 28 Ottobre 2016

Laboratorio di Statistica

- 9 settimane + 1 extra (preparazione del primo scritto di gennaio)
- 2 effettive la settimana, secondo il seguente schema:

	Mercoledì	**Venerdì**
Orario	15:30 -17:30	15:30 -17:30
Aula	309	307

Le lezioni del venerdì sono una replica di quelle del mercoledì.

- Sito del corso: http://genuzio.di.unimi.it/laboratoriostatistica.html
- Trovate:
  - *Slide e materiale didattico di ogni lezione*
  - *Collegamenti a manuali R*

- *Eventuali avvisi*

Argomenti del corso

1. PRIMA PARTE:
    - *Introduzione al linguaggio R*
        - Strutture dati R: vettori, matrici, liste, data frame
        - Istruzioni di controllo del flusso
        - Funzioni e script
        - Operazioni di I/O
        - Grafica
2. SECONDA PARTE:
    - *esercitazioni di statistica e calcolo delle probabilità*

Lo scopo è fornire un'introduzione a R e sfruttarlo come strumento per l'analisi dati.

Ambiente R

- Software gratuito disponile al repository CRAN(Comprehensive R Archive Network)
  - *Oltre all'ambiente R, fornisce documentazione e package R aggiornati*
  - *Disponibile per piattaforme Linux, Windows and MacOS*
- Linguaggio ad alto livello
- Disponibili numerosi package che implementano algoritmi in diversi ambiti: statistica, machine learning, bioinformatica, economia, medicina, etc. (è un linguaggio diffuso e richiesto nel mondo del lavoro)

Ambiente R

Linguaggio interpretato:

- I comandi vengono eseguiti direttamente dall'interprete
- Il codice non deve essere compilato

Ottimo per l'elaborazione di dati statistici:

- Praticamente è tutto già implementato, nelle librerie base di R o in package installabili semplicemente con un comando (quasi sempre)

Ambiente R

Installare R:

- Pagina per Linux
- Pagina per il download .pkg OSX
- Pagina per il download per Windows

Per avviare R:

- Linux: digita ‘R’ dalla linea di comando
- OS X : digata ‘R’ in Spotlight search
- Windows: apri R dalla GUI (Graphical User Interface)

Ambiente R

Uscire da R:

```
q()
```

Chiedere aiuto ad R

```
help() # chiede l'help generale
help(nomecomando) # chiede l'help del comando
?nomecomando # chiede l'help del comando
help.start() # lancia l'help in un browser
```

Ambiente R

Ulteriori comandi base

```
ls() # mostra il contenuto del workspace
rm(i) # rimuove dal workspace la variabile i
rm(list=ls()) # rimuove tutto (dati, funzioni) dal workspace
save(a, b ,file="prova.rda") # salva le variabili a e b nel file  'prova.rda'
load("prova.rda") # ricarica nel workspace i dati salvati in precedenza
setwd(nome_directory) # sposta il controllo di R nella directory specificata
getwd() # visualizza la directory corrente
```

Vettori

- Nei linguaggi di programmazione ad alto livello non si ha accesso diretto alla memoria fisica, ma ad una sua astrazione
- I vettori rappresentano sequenze di elementi omogenei accessibili direttamente mediante un indice
- Un vettore v è rappresentabile tramite una struttura unidimensionale:

```
y <- c(5,7,8,NA,1,5,8)
```

```
y
## [1]  5  7  8 NA  1  5  8
```

- Gli indici partono da 1. Il valore in posizione 3 è 8.

```
y[3]
## [1] 8
```

Creazione vettori

```r
c(1,4,5) # crea un vettore di interi
c("A","B","C") # crea un vettore di caratteri
c("gatto", "topo", "12") # crea un vettore di stringhe
```

La funzione c(arg1, arg2, arg3, …) concatena i suoi argomenti.

NB:

```r
c("gatto", "topo",12)
## [1] "gatto" "topo"  "12"
```

Cosa è successo al 12?

Assegnamenti

```
X <- c(1, 4, 5)
X
## [1] 1 4 5
```

La variabile X rappresenta ora il vettore <1 4 5>: si può pensare come un "contenitore" del vettore stesso. Un nuovo assegnamento cancella il contenuto precedente

```
X <- c(5,8,5)
X
## [1] 5 8 5
Y <- 100 # vettore formato da 1 solo elemento
Y
## [1] 100
```

Operazioni con vettori aritmetici

- Le operazioni usuali dell' aritmetica vengono eseguite sui vettori elemento per elemento:

```
x <- c(1,5,8,3)
y <- c(4,5,6,3)
z <- x + y
z
## [1]  5 10 14  6
d<- x - y
d
## [1] -3  0  2  0
p <- x * y
p
## [1]  4 25 48  9
q <- y / x
q
## [1] 4.00 1.00 0.75 1.00
```

Esempi di funzioni applicate a vettori numerici

```
x
## [1] 1 5 8 3
max(x)
## [1] 8
min(x)
## [1] 1
range(x)
## [1] 1 8
sum(x)
## [1] 17
prod(x)
## [1] 120
sort(x) #ordina il vettore
## [1] 1 3 5 8
order(x) # indici corrispondenti agli elementi ordinati
## [1] 1 4 2 3
```

NB sulla funzione sort:

```
x
## [1] 1 5 8 3
sort(x) #restituisce l'ordine ma x rimane invariato
## [1] 1 3 5 8
x<-sort(x) # restituisce l'ordine e lo assegna a x
x
## [1] 1 3 5 8
```

Generazione di sequenze regolari

- R dispone di diversi comandi per generare automaticamente sequenze di numeri:

```
c(1:10)
##  [1]  1  2  3  4  5  6  7  8  9 10
c (5:1)
## [1] 5 4 3 2 1
seq (1,10)
##  [1]  1  2  3  4  5  6  7  8  9 10
seq (from=1, to=4, by=0.5)
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0
```

La funzione seq() può avere 5 argomenti (si veda l' help).

Generazione di sequenze regolari (cont.)

- Un'altra funzione per generare repliche di vettori è rep():

```
rep( c(1,2), times=4)
## [1] 1 2 1 2 1 2 1 2
```

Vettori ed operatori logici

Sono vettori i cui elementi possono assumere valore TRUE o FALSE

- Si ottengono dall'applicazione di operatori logici <, <=, >, >=, == (uguaglianza), != (disuguaglianza)
- Possibile trasformarli in valori numerici mediante il comando as.numeric()

```
as.numeric(TRUE)
## [1] 1
as.numeric(FALSE)
## [1] 0
```

Vettori ed operatori logici (esempi)

```
x
## [1] 1 5 8 3
x > 5
## [1] FALSE FALSE  TRUE FALSE
x <= 5
## [1]  TRUE  TRUE FALSE  TRUE
x == 5
## [1] FALSE  TRUE FALSE FALSE
x != 5
## [1]  TRUE FALSE  TRUE  TRUE
x == c(5,6)  # vale la "regola del riciclo"!
## [1] FALSE FALSE FALSE FALSE
```

NB regola del riciclo

```
x
## [1] 1 5 8 3
length(x)
## [1] 4
length(c(5,6))
## [1] 2
x == c(5,6) # == (x == c(5,6,5,6))
## [1] FALSE FALSE FALSE FALSE
x == c(3,6,8) # == (x == c(3,6,8,3))
## Warning in x == c(3, 6, 8): longer object length is not a multiple of
## shorter object length
## [1] FALSE FALSE  TRUE  TRUE
```

Dati mancanti

- In molte situazioni reali i componenti di un vettore possono essere "non noti" o comunque non disponibili.
- In questi casi R riserva il valore speciale NA (Not Available)
- In generale qualunque operazione che coinvolga valori NA ha come risultato NA.

```
x <- c(1:4, NA)
x + 2
## [1]  3  4  5  6 NA
```

Dati mancanti

Si noti che NA non è un valore ma un marcatore di una quantità non disponibile:

```
x
## [1]  1  2  3  4 NA
x == NA
## [1] NA NA NA NA NA
```

- Per individuare quali elementi siano effettivamente NA in un vettore si deve usare la funzione is.na():

```
is.na(x)
## [1] FALSE FALSE FALSE FALSE  TRUE
```

Vettori: selezione e accesso a sottoinsiemi di elementi

Esistono diverse modalità di accesso a singoli elementi o a sottoinsiemi di elementi di un vettore. In generale la selezione e l'accesso avviene tramite l'operatore [] (parentesi quadre):

- sottoinsiemi di elementi di un vettore sono selezionati collegando al nome del vettore un vettore di indici in parentesi quadre.

Esistono 4 diverse modalità di selezione/accesso:

1. Vettori di indici interi positivi
2. Vettore di indici interi negativi
3. Vettore di indici logici

4. Vettori di indici a caratteri

Selezione ed accesso tramite vettori di indici interi positivi

- Gli elementi di un vettore x sono selezionati tramite un vettore y di indici positivi racchiuso fra parentesi quadre: x[y]
- I corrispondenti elementi sono selezionati e concatenati.

```
x <- 5:10
x[1] # selezione di un singolo elemento
## [1] 5
x[5]
## [1] 9
length(x) # numero elementi nel vettore
## [1] 6
x[7] # accesso fuori range
## [1] NA
x[2:4]
## [1] 6 7 8
x[c(1,3,5)]
## [1] 5 7 9
```

Selezione ed accesso tramite vettori di indici interi negativi

Sono selezionati gli elementi di un vettore x che devono essere esclusi tramite un vettore y di indici negativi racchiuso fra parentesi quadre: x[y]

```
y <- rep(c("G","A","T","T"), times=3)
y
##  [1] "G" "A" "T" "T" "G" "A" "T" "T" "G" "A" "T" "T"
z <- y[-(1:5)] # selezionati tutti gli elementi di y eccetto primi 5
z
## [1] "A" "T" "T" "G" "A" "T" "T"
z <- z[-length(z)] # cancellazione dell'ultimo elemento di z
z
## [1] "A" "T" "T" "G" "A" "T"
```

Selezione ed accesso tramite vettori indice logici (filtro)

- Il vettore degli indici deve essere della stessa lunghezza del vettore i cui elementi devono essere selezionati. Sono selezionati gli elementi corrispondenti a TRUE ed omessi gli altri.

```
x <- c(1:5,NA,NA)
x
## [1]  1  2  3  4  5 NA NA
i <- c(rep(TRUE,times=3),rep(FALSE,times=4))
i
## [1]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
x[i]
## [1] 1 2 3
x[!is.na(x)] # selezione elementiche non sono NA
## [1] 1 2 3 4 5
x[!is.na(x) & x > 2]
## [1] 3 4 5
```

Selezione ed accesso tramite vettori indice logici (cont.)

```
x<- 0:10
x
##  [1]  0  1  2  3  4  5  6  7  8  9 10
x[ x < 2 | x > 6 ]
## [1]  0  1  7  8  9 10
```

"Scoppattiamo" questo esempio

```
minori<- x < 2
minori
##  [1]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
maggiori<- x > 6
maggiori
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
x[ minori | maggiori]
## [1]  0  1  7  8  9 10
```

Selezione ed accesso tramite vettori di indici a caratteri

Applicabile quando è stato impostato l'attributo names del vettore

- Associa una stringa ad ogni elemento
- Un sottovettore di names seleziona le componenti corrispondenti

```
campione <- c(45, 180, 70, 2007, 3)
names(campione) <-c("Eta","Altezza","Peso","AnnoIns","Ricoveri")
campione
##      Eta  Altezza     Peso  AnnoIns Ricoveri
##       45      180       70     2007        3
Eta.Anno <- campione[c("Eta","AnnoIns")]
Eta.Anno
##     Eta AnnoIns
##      45    2007
```

Selezione ed accesso tramite vettori di indici a caratteri

Casi particolari delle funzione names

```
names(campione) <-c("Eta","Altezza","Peso","AnnoIns")
campione
##     Eta Altezza    Peso AnnoIns    <NA>
##      45     180      70    2007       3
#il comando seguente non funziona
# names(campione) <-c("Eta","Altezza","Peso","AnnoIns", "Ricoveri", "Parite")
```

Altri comandi utili

- sample(x, n, replace = FALSE) prende un sample da x di una dimensione n con o senza reinserimento

```
sample(1:6, 3)
## [1] 1 3 4
# sample(1:6, 7) errore!!!
# cerco di prendere un sample di 7 elementi partendo da un
# vettore di 6 elementi senza ripetizioni
sample(1:6, 7, replace=TRUE)
## [1] 5 1 2 4 5 5 5
```

Altri comandi utili (cont.)

- floor(x), ceiling(x), trunc(x), round(x, digits=3)

```
x<- c (1.4,2.9, 6.453463,9.58761)
floor(x)
## [1] 1 2 6 9
ceiling(x)
## [1]  2  3  7 10
trunc(x)
## [1] 1 2 6 9
round(x, digits=3)
## [1] 1.400 2.900 6.453 9.588
```

N.B. floor vs trunc

```
x <- c(-3.2, -1.8, 2.3, 2.9)
floor(x)
## [1] -4 -2  2  2
trunc(x)
## [1] -3 -1  2  2
```

Altri comandi utili (cont.)

- paste("stringa1", "stringa", sep="-") concatenazione di stringhe

```
paste ("Gianni", "Pinotto", sep= " e ")
## [1] "Gianni e Pinotto"
```

- setdiff(vettore1, vettore2) differenza insiemistica, elementi in vettore1 non in vettore2

```
y <- c ("Marco", " e ", "Pinotto")
z <- c ("Gianni", " e ","Pinotto")
setdiff(y,z)
## [1] "Marco"
```

E ora?

…. esercizi

Esercizio (1)

Inserire i seguenti valori numerici in tre distinti vettori:

- 2.5, 2, 5, 1, 11, -0.1, 10, 9
- 10, 9.5, 9, 8.5, 8, 7.5, 7, 6.5
- 1, 2, 3, 4, 5, 6, 7, 8

Quindi:

1. Calcolare minimo e massimo di ogni vettore
2. Calcolare la somma degli elementi di ciascun vettore
3. Calcolare la somma dei primi due vettori
4. Calcolare la differenza dei il secondo e il terzo vettore

5. Calcolare il prodotto degli elementi del secondo vettore
6. Concatenare i tre vettori ed estrarre un campione casuale pari al 30% di tutti gli elementi

Esercizio (2)

I seguenti dati rappresentano il numero di giorni in cui ciascuno di 20 lavoratori si è assentato per malattia nelle ultime sei settimane:

2, 2, 0, 0, 5, 8, 3, 4, 1, 0, 0, 7, 1, 7, 1, 5, 4, 0, 4, 0

- I codici dei lavoratori sono "Lav0", "Lav1", …, "Lav19" Quindi:
  1. Creare un vettore contenente i giorni di malattia e nel campo names i corrispondenti codici
  2. Selezionare il numero di giorni di malattia del lavoratore "Lav6""
  3. Determinare quanti lavoratori hanno fatto 0 giorni di malattia e stamparne i codici
  4. Estrarre il sottovettore contenente i giorni di malattia di tutti i lavoratori eccetto "Lav2"" e "Lav11"

5. Estrarre uno dei lavoratori con il minor numero di giorni di malattia
6. Estrarre il lavoratore con il maggior numero di giorni di malattia