Prima introduzione a R

M. Romito

5 ottobre 2017

MASTER COPY
Tel. 050 831 21 26
Cell. 388 9837745

Generazione e visualizzazione di un vettore

```
x = c(1,2,4)
x
```

Creo un vettore c contenente dei numeri

visualizzo il vettore x

Lunghezza di un vettore

```
x = c(1,2,4)
length(x)
```

Visualizzazione di parti o funzioni di un vettore

```
x[3]
```

x[3] → mi da il numero in posizione 3

```
x[3:4]
```

x[3:4] → mi da tutti i numeri compresi tra il 3° e il 4°

x[-3] → (per escludere si mette il −) mi da tutti i numeri in x tranne quello in posizione 3

```
z=1/x
z
x*z
```

quando si applica una funzione ad un vettore, me lo applica ad ogni singolo elemento del vettore

```
y=c(1,3)
x*y
```

* in questo caso i due vettori, x e y, hanno lunghezza ≠, poiché x ha lunghezza 3, mentre y ha lunghezza 2, quindi il software esegue cmq l'operazione, ma "solitamente" esegue alcune operazioni

L'operazione è stata eseguita lo stesso, ma il software ha restituito un messaggio di avviso. Le operazioni che esso ha eseguito sono le seguenti

- `y1=c(y,y)`: concatenazione di due copie del vettore più corto, (y1 = (1,3,1,3) ora y1 ha lunghezza 4)
- `y2=y[1:3]`: adattamento della lunghezza,
- `x*y2`: operazione aritmetica coerente.

ATT! Il fatto che il programma da un risultato, (eventualmente senza segnalare l'errore) non significa che ha fatto quello che volevamo!

* Infatti, in questo caso, il programma ha prolungato un vettore (duplicando quello + corto), in questo caso y, fino ad arrivare ad una lunghezza almeno pari a quella dell'altro, e poi, se la concatenazione ha generato un vettore + lungo, lo adatterà poi alla lunghezza del + corto.

```
r=c(1,4,5,6)
y
r
r*y
y
y^2
```

il prodotto tra vettori deve sempre avvenire tra due vettori di uguale lunghezza

Alcune operazioni con i vettori

```
z
abs(z)
u = c(1,-2,0.53567)
```

```
abs(u)
round(u)
length(u)
min(u)
max(u)
max(abs(u))
sum(u)
sort(u)
rev(u)
rev(sort(u))
```

abs(u) valore assoluto

round(u) round (u,3) → arrotondamento a 3 cifre → viene fatto all'intero inferiore cioè per difetto

min(u) il minimo valore presente nel vettore

max(u) il massimo valore presente nel vettore

sum(u) somma di tutti gli elementi del vettore

sort(u) riordinare un vettore (in ordine crescente)

rev(u) prende il vettore e lo inverte → invece fare sort(rev(x)) NON ha senso! è inutile! Posso fare direttamente sort(x)

rev(sort(u)) ordina il vettore e poi lo inverte xché per convenzione viene eseguito il comando più interno e poi quello più esterno

↓ in ordine crescente

↓ in ogni caso ha eseguito l'ordine decrescente

Primi comandi statistici

```
mean(x)
mean(u)
sd(u)
```

mean(x) media di un vettore : + lunga è la nostra sequenza di dati, migliore sarà l'approssimazione della media

sd(u) deviazione standard (standard deviation)

Generazione di sequenze

```
q = 10:10
q
q = 10:1
q
w = seq(from=0, to=1, by=0.01)
w
e = seq(0,1,0.01)
e
```

q = 10:10 → da ___ a

ATTENZIONE al punto!

w = seq(from=0, to=1, by=0.01) voglio i valori da 0 a 1 (ad intervalli di) con un incremento di (1 centesimo)

|| è lo stesso comando ||

e = seq(0,1,0.01) ← sequenze

posso omettere le "from", "to", "by"

Oggetti in memoria

```
ls()
rm(z)
rm()
```

rm(z) → sta per REMOVE (cancella gli elementi in memoria)

rm() → cancella TUTTO

Variabili Booleane e dati mancanti

Fortemente sconsigliato definire oggetti con i nomi `T`, `F`, `NA`, perché sono variabili interne di `R` predefinite.

T sta per TRUE

definite a priori

MASTER COPY Tel. 050 8312126 Cell. 388 9837745

F sta per FALSE

NA sta per NON DEFINITO (VUOTO)

Manipolazione di vettori

```
q = 1:100
q
```
(creo un vettore con numeri da 1 a 100)

```
q[99:100] = c(199,200)
q
```
estraggo il 99° e il 100° elemento e gli assegno due nuovi valori.

] estraggo un sottovettore da q e gli assegno un nuovo valore

```
q[99:100]=5
q
```
estraggo il 99° e il 100° elemento e assegno ad entrambi uno stesso valore

```
q[98:100]=c(1,2,3)
q
```
il nuovo vettore da assegnare è troppo lungo (contiene 3 elementi) rispetto al sottovettore che devo estrarre (che ne contiene 2) quindi l'ultimo elemento lo butta via.

```
q=1:100
q[98]=2
q[100]=500
q
```

```
q[100]=NA
q
```
numero mancante in posizione 100

```
q[-(1:10)]
```

```
o = 1:10
o[11:100]=11:100
o
```

```
o=1:10
o[20:30]=20:30
o
```

Commenti in R

```
#ciao
```

Generazione di matrici

```
z = 1:24
A = matrix(z,4,6)
A
```
matrice contenente il vettore z
n° colonne
n° righe

```
B = matrix(z,6,4)
t(B)
```
→ è una matrice 6×4
trasposta di B → è una matrice 4×6

Il riempimento avviene per colonne. Se si vuole ottenere il riempimento per righe si può operare per trasposizione, oppure usare il comando seguente:

per convenzione

ATT! Se non esplicito i valori da mettere all'interno della matrice, il software usa il valore standard NA.

Inoltre se ad es faccio `x = 1:7` `A = matrix(5,5)` → lo fa ma
- se è un multiplo ESATTO non R lavenza
- se non è un multiplo esatto R lavenza (lo riempie provvisoriamente)

se `x = 1:30` `A = matrix(x,5,5)` `A` → se sono di + alcuni li butta via

master copy COPISTERIA 050/8312126 3987745 MASTER COPY Tel. 050 8312126 Cell. 333 3937745

```
A = matrix(z,4,6,byrow=T)
A
```

riempimento per righe : se il riempimento per colonne non ci va bene lo devo specificare!

Dimensione di una matrice

```
dim(A)
dim(A)[2]
nrow(A)
ncol(A)
```

dimensione della matrice

?

mi dice il n° di righe

mi dice il n° di colonne

Visualizzazione di parti di una matrice

```
A[,2]
A[3,]
A[3,1:3]
A[1:2,4:6]
```

mi da tutti gli elementi della 2° colonna (estraggo tutta la colonna)

mi da tutti gli elementi della 3° riga (estraggo tutta la riga)

estrae tutta la 3° riga e la 1° 2° 3° colonna

Estrae la 1° e la 2° riga e la 3° 4° 5° 6° colonna

```
B=A
C=c(A,B)
C

A
c(A)

B=t(A)
z=c(B)
z

A1 = matrix(1:24,6,4)
A2 = matrix(1,6,4)
A2

B=matrix(2,4,2)
A1+A2

A1-A2

A1*A2
```

Prodotti matriciali

il prodotto tra matrici è : %*%

```
dim(A1)
dim(B)
A1 %*% B

y=1:4
A1 %*% y
```

master COPY COPISTERIA 050/8312126 388/9837745
MASTER COPY Tel. 050 8312126 Cell. 388 9837745

Alcune funzioni sulle matrici

La matrice diagonale con diagonale il vettore y

```
diag(y)
```

La trasposta di una matrice

```
A1
t(A1)
```

Una matrice di dimensioni date e riempita con un vettore (notare l'ordine di riempimento)

```
Q=matrix(1:25,5,5)
Q
```

Il determinante

```
det(Q)
```

```
E = rnorm(25)
Q=matrix(E,5,5)
Q
```

```
det(Q)
```

La matrice inversa

```
solve(Q)
```

```
Q %*% solve(Q)
```

```
I = Q %*% solve(Q)
round(I)
```

Ricerca di autovalori e autovettori

```
eigen(Q)
```

```
z = eigen(Q)
v = z$values
v
```

```
w = z$vectors
w
```

```
Autovalori = eigen(Q,only.values=T)
Autovalori
```

```
Autovalori$values
```

Classe di un oggetto

```
x=c(1:5)
class(x)
```

```
s=c(1,2,3.45656)
class(s)
```

master COPY COPISTERIA 050/8312126 388/9837745 MASTER COPY Tel. 050 831 2126 Cell. 388 9837745