

Introduzione alla chemiometria e disegno sperimentale

Modulo 2: Il Software R (matrici e dataframe)

Docente: Dr. Sabina Licen (slicen@units.it)

Materiale consultabile e siti web su R



<https://cran.r-project.org/>

Manuali principali,
anche in italiano



<https://rstudio.com/resources/cheatsheets/>

Documenti "riassuntivi" di
categorie di funzioni per R



www.statmethods.net

Molti esempi di
utilizzo di funzioni e
preparazione grafici



<https://rseek.org/>

Serve ad individuare in a
che package appartiene
una determinata funzione

Cookbook for R

<http://www.cookbook-r.com/>

Tutorial su funzioni si base
e grafici

The R Graph Gallery

<https://www.r-graph-gallery.com/>

Tanti esempi di grafici e
script già pronti!

Materiale consultabile e siti web su R (2)



<http://www.r-tutor.com/r-introduction>

Tutorial di molte funzioni e grafici



<http://www.sthda.com/english/>

Tutorial di molte funzioni e grafici



<https://www.datanovia.com/en/>

Tutorial *free* + corsi a pagamento



<https://www.datamentor.io/>

Tutorial *free* + corsi a pagamento



www.datacamp.com

Alcuni video-tutorial *free* + altri a pagamento



<https://www.r-bloggers.com/>

Domande/risposte, alcuni tutorial e link a molte fonti di informazioni su R



<https://stackoverflow.com/questions/1744861/how-to-learn-r-as-a-programming-language>

Domande/risposte

Per fare pratica... Set di dati reperibili su web

- Istruzioni per visualizzare elenco dataset presenti nella versione base di R:
<https://stackoverflow.com/questions/33797666/how-do-i-get-a-list-of-built-in-data-sets-in-r>

- Elenco dataset disponibili nei diversi package di R scaricabili come csv
<https://vincentarelbundock.github.io/Rdatasets/datasets.html>

- Alcuni dataset divisi secondo i possibili utilizzi (regressione, DoE, analisi multivariata, etc..)
<http://openmv.net/>

- Molti dataset divisi secondo i possibili utilizzi (regressione, DoE, analisi multivariata, etc..)
<https://dasl.datadescription.com/datafiles/>

- Data in Brief: rivista *open access* con dataset disponibili e alcune elaborazioni
<https://www.journals.elsevier.com/data-in-brief>

I vettori - riassunto

Insieme lineare di elementi omogenei per tipologia (tutti numeri, tutti caratteri, tutti logici)

Per generare un vettore esistono molti modi, il più comune dei quali è la funzione
c(elemento1, elemento2, ecc..)
che combina i valori presenti nel suo elenco in un vettore.

Altri modi per creare un vettore:

- **c(n₁:n₂)** genera un vettore con elementi a partire dal numero n₁ fino al numero n₂ a passo 1;
- **rep(n₁, n₂)** genera un vettore con n₂ elementi di valore n₁;
- **seq(n₁, n₂, n₃)** genera un vettore con numeri che vanno da n₁ a n₂ a passo n₃.

Attributi di un vettore:

length(nomeoggetto) indica il numero di elementi del vettore

mode(nomeoggetto) o **class(nomeoggetto)** indica il tipo di contenuto del vettore
(numeri, testo, ecc...)

Aritmetica con i vettori

I vettori possono essere utilizzati in qualunque espressione aritmetica. In questo caso l'operazione è applicata a tutti gli elementi del vettore.

```
> #Vedere il "contenuto" dell'oggetto
> d
[1] 10.6  1.0  2.0  5.0  7.0
>
> #Moltiplicare per un numero
> d*2
[1] 21.2  2.0  4.0 10.0 14.0
>
> #Moltiplicare per una variabile
> d*a
[1] 112.36  10.60  21.20  53.00  74.20
>
> #Moltiplicare per un altro vettore (di stessa lunghezza)
> d*AA[1:5]
[1] 190.8  17.0  32.0  75.0  98.0
>
> #elevamento al quadrato
> d^2
[1] 112.36  1.00  4.00  25.00  49.00
>
> |
```

SEGUE 

➤ Ai vettori possono essere applicate anche funzioni statistiche:

```
min(nomeoggetto)    median(nomeoggetto)
max(nomeoggetto)    quantile(nomeoggetto)
sum(nomeoggetto)    var(nomeoggetto)
mean(nomeoggetto)   sd(nomeoggetto)
```

```
> #Vedere il "contenuto" dell'oggetto
> d
[1] 10.6  1.0  2.0  5.0  7.0
>
> #valore minimo
> min(d)
[1] 1
>
> #valore massimo
> max(d)
[1] 10.6
>
> #somma degli elementi
> sum(d)
[1] 25.6
>
> #media
> mean(d)
[1] 5.12
>
```

```
> #mediana
> median(d)
[1] 5
>
> quantile(d)
 0%  25%  50%  75% 100%
1.0  2.0  5.0  7.0 10.6
>
> # 1° quartile
> quantile(d, probs = 0.25)
25%
 2
>
> # 3° quartile
> quantile(d, probs = 0.75)
75%
 7
>
> # quartili e mediana
> quantile(d, probs = c(0.25,0.5,0.75))
25% 50% 75%
 2  5  7
>
> # Calcolare la varianza
> var(d)
[1] 15.072
>
>
> # Calcolare la deviazione standard
> sd(d)
[1] 3.882267
> |
```

Individuare elementi nei vettori

```
> #Vedere il "contenuto" dell'oggetto
```

```
> d  
[1] 10.6 1.0 2.0 5.0 7.0
```

```
> #Individuare un elemento
```

```
> d[5]  
[1] 7
```

```
> #Individuare tutti gli elementi tranne quello indicato
```

```
> d[-5]  
[1] 10.6 1.0 2.0 5.0
```

```
> #Individuare più di un elemento
```

```
> d[c(1,4)]  
[1] 10.6 5.0
```

```
> #Individuare tutti gli elementi tranne quelli indicati
```

```
> d[-c(1,4)]  
[1] 1 2 7
```

```
> #Individuare un sottoinsieme di elementi
```

```
> d[3:5]  
[1] 2 5 7
```

```
> #Individuare un sottoinsieme di elementi esclusi quelli indicati
```

```
> d[-(3:5)]  
[1] 10.6 1.0
```

```
> #Individuare tutti gli elementi "maggiori di"
```

```
> d[d>3]  
[1] 10.6 5.0 7.0
```

```
> #Individuare tutti gli elementi "maggiori di" e "minori di"
```

```
> d[d>3&d<7]  
[1] 5
```

```
> #Individuare tutti gli elementi "minori di" o "maggiori di"
```

```
> d[d<3|d>7]  
[1] 10.6 1.0 2.0
```

Si usano dei criteri all'interno di parentesi quadre **nomeoggetto[criterio]**

In taluni casi nella scrittura del criterio viene "citato" il nome dell'oggetto

Individuare la posizione di elementi nei vettori

```
> #Vedere il "contenuto" dell'oggetto
> d
[1] 10.6  1.0  2.0  5.0  7.0
>
> #Individuare l'indice di POSIZIONE degli elementi "uguali a" o
> #"maggiori di" o "minori di", ecc...
>
> which(d==2)
[1] 3
> which(d!=1)
[1] 1 3 4 5
> which(d>3)
[1] 1 4 5
> which(d<5)
[1] 2 3
> which(d>3&d<7)
[1] 4
> which(d<3|d>7)
[1] 1 2 3
> |
```

Attenzione!!!

Qui si usano parentesi tonde

which è una funzione che serve ad individuare l'indice di posizione di elementi di un vettore che corrispondono ad un determinato criterio

Gli operatori logici utilizzabili in R sono:

< minore

<= minore o uguale

> maggiore

>= maggiore o uguale

== uguale

!= diverso

& è l'intersezione ("e")

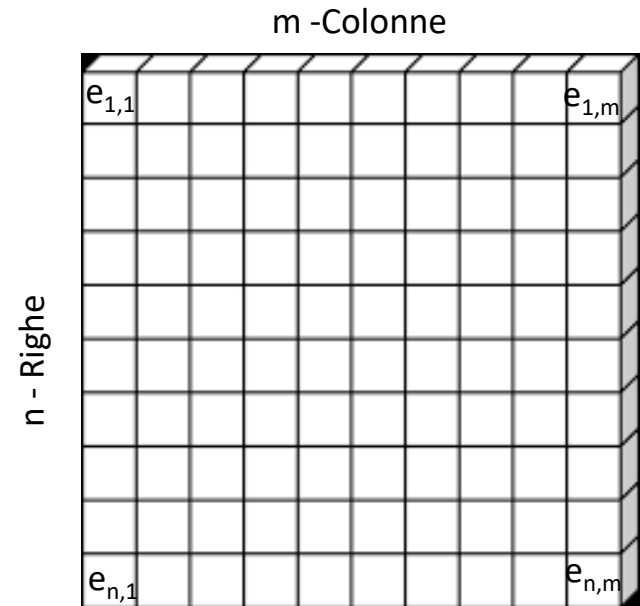
| è l'unione ("o")

! è la negazione ("non")

Le matrici

Le matrici sono oggetti contenenti **esclusivamente un tipo di valori (di solito numeri)**
disposti in righe e colonne

```
> #Per generare una matrice è possibile "appaiare" dei vettori
> #come se fossero colonne (i vettori devono avere stessa lunghezza):
>
> Mtx<-cbind(d[1:5],Cx[16:20],Dx[4:8])
>
> #Vedere il "contenuto" dell'oggetto
> Mtx
      [,1] [,2] [,3]
[1,] 10.6  5.0  3.0
[2,]  1.0  5.5  2.5
[3,]  2.0  6.0 -1.0
[4,]  5.0  6.5 -0.8
[5,]  7.0  7.0 -0.6
> |
```



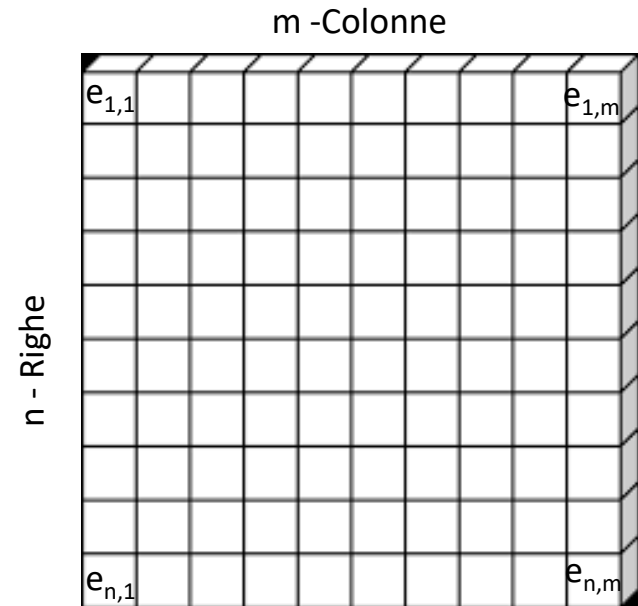
cbind(ogg1,ogg2,...) serve a raggruppare vettori di uguale lunghezza in una matrice

SEGUE

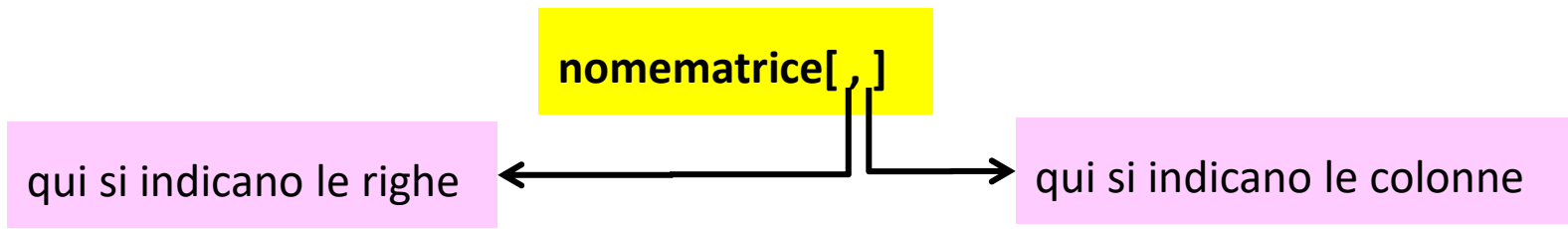
Le matrici

`matrix(vettore, nrow=...,ncol=...)` serve ad ottenere una matrice "forzando" un vettore a dividersi in **nrow** righe e **ncol** colonne

```
> matrix(c(1:20), 5, 4)
      [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20
> |
```



Individuare elementi nelle matrici



```
> #Individuare un elemento
> Mtx[4,3]
[1] -0.8
>
> #Individuare una riga
> Mtx[3,]
Var1 Var2 Var3
  2    6   -1
>
> #Individuare una colonna
> Mtx[,2]
  c1  c2  c3  c4  c5
5.0 5.5 6.0 6.5 7.0
>
> #Individuare più di una riga
> Mtx[c(1,4),]
  Var1 Var2 Var3
c1 10.6  5.0  3.0
c4  5.0  6.5 -0.8
> Mtx[4:5,]
  Var1 Var2 Var3
c4    5  6.5 -0.8
c5    7  7.0 -0.6
> |
```

```
> #Individuare più di una colonna
> Mtx[,c(1,3)]
  Var1 Var3
c1 10.6  3.0
c2  1.0  2.5
c3  2.0 -1.0
c4  5.0 -0.8
c5  7.0 -0.6
> Mtx[,1:2]
  Var1 Var2
c1 10.6  5.0
c2  1.0  5.5
c3  2.0  6.0
c4  5.0  6.5
c5  7.0  7.0
>
> #Individuare un sottoinsieme di elementi
> Mtx[2:4,2:3]
  Var2 Var3
c2  5.5  2.5
c3  6.0 -1.0
c4  6.5 -0.8
> |
```

I dataframe

I dataframe sono oggetti con elementi disposti in righe e colonne, che possono contenere sia numeri che caratteri o logici purché ogni colonna contenga esclusivamente una classe di valori

```
> #Per generare un dataframe è possibile "appaiare" dei vettori
> #come se fossero colonne (i vettori devono avere stessa lunghezza
> # e possono contenere sia numeri che testo):
>
> Dtf<-data.frame(vex,b,AA[3:6],Dx[6:9])
>
> #Vedere il "contenuto" dell'oggetto
> Dtf
      vex b AA.3.6. Dx.6.9.
1   pera 4      6    -2.0
2 banana 1      5    -1.8
3   mela 6      4    -1.6
4  pesca 9      3    -1.4
> |
```

data.frame(ogg1,ogg2,...) serve a raggruppare vettori di uguale lunghezza in un dataframe

Attributi dei dataframe

```
> #Numero di colonne
> ncol(Dtf)
[1] 4
>
> #Numero di righe
> nrow(Dtf)
[1] 4
>
> #Nomi delle intestazioni di colonna
> colnames(Dtf)
[1] "vex"      "b"        "AA.3.6." "Dx.6.9."
>
> #Assegnare dei nomi alle intestazioni di colonna
> colnames(Dtf) <- c("Campione", "Var1", "Var2", "Var3")
>
> #Vedere il "contenuto" dell'oggetto
> Dtf
  Campione Var1 Var2 Var3
1     pera   4    6 -2.0
2  banana   1    5 -1.8
3     mela   6    4 -1.6
4     pesca   9    3 -1.4
> |
```

oppure...

```
> dim(Dtf)
[1] 4 4
>
```

```
> class(Dtf)
[1] "data.frame"
> mode(Dtf)
[1] "list"
>
```

Generare un dataframe con intestazioni di colonna predefinite

```
> #Cancellare il dataframe
>
> rm(Dtf)
>
> #Creare il nuovo dataframe
> Dtf<-data.frame(Campione=vex,Var1=b,Var2=AA[3:6],Var3=Dx[6:9])
>
> #Vedere il "contenuto" dell'oggetto
> Dtf
  Campione Var1 Var2 Var3
1     pera   4    6 -2.0
2  banana   1    5 -1.8
3     mela   6    4 -1.6
4    pesca   9    3 -1.4
> |
```

data.frame(nomecolonna1=ogg1,nomecolonna2=ogg2,...)

Individuare elementi nei dataframe

`nomedataframe[,]`

qui si indicano le righe

qui si indicano le colonne

```
> #Individuare un elemento
```

```
> Dtf[2,4]
```

```
[1] -1.8
```

```
>
```

```
> #Individuare una riga
```

```
> Dtf[4,]
```

```
 Campione Var1 Var2 Var3
```

```
4   pesca    9    3 -1.4
```

```
>
```

```
> #Individuare una colonna
```

```
> Dtf[,3]
```

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

```
> #Individuare più di una riga
```

```
> Dtf[c(1,4),]
```

```
 Campione Var1 Var2 Var3
```

```
1   pera    4    6 -2.0
```

```
4   pesca    9    3 -1.4
```

```
> Dtf[2:3,]
```

```
 Campione Var1 Var2 Var3
```

```
2   banana    1    5 -1.8
```

```
3    mela    6    4 -1.6
```

```
> |
```

```
> #Individuare più di una colonna
```

```
> Dtf[,c(1,3)]
```

```
 Campione Var2
```

```
1     pera    6
```

```
2   banana    5
```

```
3     mela    4
```

```
4   pesca    3
```

```
> Dtf[,1:2]
```

```
 Campione Var1
```

```
1     pera    4
```

```
2   banana    1
```

```
3     mela    6
```

```
4   pesca    9
```

```
>
```

```
> #Individuare un sottoinsieme di elementi
```

```
> Dtf[2:4,2:3]
```

```
 Var1 Var2
```

```
2     1    5
```

```
3     6    4
```

```
4     9    3
```

```
> |
```

SEGUE

Individuare elementi nei dataframe

```
> #ALTRO MODO di richiamare COLONNE, molto IMPORTANTE!!!  
> Dtf$Campione  
[1] pera   banana mela   pesca  
Levels: banana mela pera pesca  
>  
> Dtf$Var3  
[1] -2.0 -1.8 -1.6 -1.4  
> |
```

nomeoggetto\$nomecolonna1

(con le matrici non funziona!)

Osservare come viene interpretata "concettualmente" questa colonna del dataframe:

Variabili "factor"



Vettori che contengono dei "livelli"

Individuare elementi di un dataframe secondo criteri

```
nomedataframe[which(nomedataframe$nomecolonna "condizione/i"), ]
```

```
> #Righe complete che soddisfano la condizione per una colonna:
```

```
> Dtf[which(Dtf$Campione=="pera"), ]
```

```
  Campione Var1 Var2 Var3
1      pera   4   6   0
```

```
>
```

```
> Dtf[which(Dtf$Var3==0), ]
```

```
  Campione Var1 Var2 Var3
1      pera   4   6   0
3      mela   6   4   0
```

```
>
```

```
> #Righe complete che soddisfano più di una condizione per colonne:
```

```
> Dtf[which(Dtf$Var1>1 & Dtf$Var3==0), ]
```

```
  Campione Var1 Var2 Var3
1      pera   4   6   0
3      mela   6   4   0
```

```
>
```

```
> #Righe complete che soddisfano condizioni alternative per colonne:
```

```
> Dtf[which(Dtf$Var1<2 | Dtf$Var1>6), ]
```

```
  Campione Var1 Var2 Var3
2  banana   1   5   3
4   pesca   9   3   3
```

```
> |
```

Menu modifica - Editor dei dati

The image shows the RGui (64-bit) interface. The 'Modifica' menu is open, showing options like 'Copia', 'Incolla', 'Incolla solo i comandi', 'Copia e incolla', 'Seleziona tutti', 'Pulisci console', 'Editor dei dati...', and 'Preferenze interfaccia...'. A yellow callout (1) points to the 'Editor dei dati...' option. A yellow callout (2) points to the 'Question' dialog box, which asks for the name of the data frame or matrix, with 'A' entered in the text field. A yellow callout (3) points to the 'Editor dei dati' window, which displays a table with columns 'col1', 'col2', 'col3', 'col4', 'var5', and 'var6'. The first row of the table is highlighted in red. A yellow callout (4) points to the close button (X) in the top right corner of the 'Editor dei dati' window. The console window shows the following code and output:

```
> matrix(c(1:20),5,4)
     [,1] [,2] [,3] [,4]
[1,]  1   6  11  16
[2,]  2   7  12  17
[3,]  3   8  13  18
[4,]  4   9  14  19
[5,]  5  10  15  20
> A<-matrix(c(1:20),5,4)
> |
```

La funzione **fix()** consente di effettuare la stessa operazione