

Generalizzazioni di vettori

- I vettori sono sequenze ordinate di elementi omogenei. Un vettore v è rappresentabile tramite una struttura Unidimensionale:

v

4	8	2	1	3	1	5	0
---	---	---	---	---	---	---	---

- Essendo strutture unidimensionali è possibile accedere o modificare un elemento di un vettore utilizzando un unico indice:

$> v[2]$
8

Matrici

- In R è possibile rappresentare estensioni bidimensionali di vettori (**matrici**)

m

3	7	6	5	3	4
6	7	8	7	5	4
1	0	7	8	9	0
9	0	8	7	4	2
9	6	7	5	1	2
5	9	8	6	4	3

Essendo strutture bidimensionali, è necessaria una coppia di indici per accedere o modificare un elemento di una matrice :

> m[1,3] # seleziona el. 1° riga e 3° colonna
6

> m[2,4] <- 0 # assegna 0 all'elemento di
posizione 2 e 4

Costruzione di matrici

- Le matrici possono essere costruite tramite la funzione **matrix** a partire da un vettore esistente

```
> x <- 1:24
> m <- matrix (x,nrow=4)#genera una
matrice con 4 righe utilizzando gli
elementi del vettore x

> m      # si noti l' assegnamento dei
valori "per colonne"
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  1   5   9  13  17  21
[2,]  2   6  10  14  18  22
[3,]  3   7  11  15  19  23
[4,]  4   8  12  16  20  24
```

```
> length (m)
[1] 24
> mode (m)
[1] "numeric"
> dim(m)
[1] 4 6
```

- Il comando *dim* mostra che la matrice *m* è un array bidimensionale 4 X 6
- **nrow** (m)
[1] 4
- **ncol** (m)
[1] 6

Costruzione di matrici

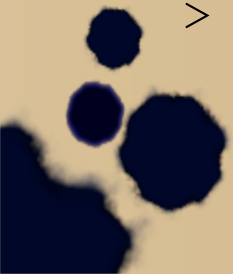


- Parametri di `matrix` (si veda l'help per maggiori dettagli):
 - **ncol**: numero di colonne che la matrice deve avere
 - **byrow**: inserimento degli elementi per righe

```
> x <- 1:12
> m <- matrix (x, ncol=4, byrow=TRUE)
> m
      [,1] [,2] [,3] [,4]
[1,]  1    2    3    4
[2,]  5    6    7    8
[3,]  9   10   11   12
> m <- matrix (x, ncol=5) # anche per le matrici si
                          applica la "regola del riciclo"
```

- Comandi utili: **as.matrix**, per trasformare oggetti R in matrici
- **rownames**, **colnames** per impostare i nomi delle righe e delle colonne :

```
> rownames (m) <-c ("nomer1", "nomer2", "nomer3")
> colnames (m) <-c ("nomec1", "nomec2", "nomec3", "nomec4")
```



Concatenazione di matrici

- Le matrici possono essere costruite anche tramite le funzioni **cbind** ed **rbind**

- *cbind* forma matrici legando insieme vettori o matrici “orizzontalmente”:

```
> x <- 1:3
> y <- 4:6
> m <- cbind(x, y)
> m
```

	x	y
[1,]	1	4
[2,]	2	5
[3,]	3	6

- *rbind* forma matrici legando insieme vettori o matrici “verticalmente”:

```
> x <- 1:3
> y <- 4:6
> m <- rbind(x, y)
> m
```

	[, 1]	[, 2]	[, 3]
x	1	2	3
y	4	5	6

Operazioni con le matrici

- Somma e prodotto con costanti

```
> x <- 1:12
> A <- matrix(x,nrow=3)
> A + 2
```

	[,1]	[,2]	[,3]	[,4]
[1,]	3	6	9	12
[2,]	4	7	10	13
[3,]	5	8	11	14

```
> B <- A * 2
> B
```

	[,1]	[,2]	[,3]	[,4]
[1,]	2	8	14	20
[2,]	4	10	16	22
[3,]	6	12	18	24

- Somma e prodotto “elemento per elemento”

```
> A+B
```

	[,1]	[,2]	[,3]	[,4]
[1,]	3	12	21	30
[2,]	6	15	24	33
[3,]	9	18	27	36

```
> A*B
```

	[,1]	[,2]	[,3]	[,4]
[1,]	2	32	98	200
[2,]	8	50	128	242
[3,]	18	72	162	288

Operazioni con le matrici

- Prodotto di matrici “righe per colonne”: operatore `%*%`
- Si ricordi che A e B sono matrici 3X4:

```
> A%*%B
```

```
Error in A %*% B : non conformable arguments
```

- Il numero delle colonne di A deve essere uguale al numero di righe di B!

```
> A%*%t(B) # t(B) trasposta di B
```

	[, 1]	[, 2]	[, 3]
[1,]	332	376	420
[2,]	376	428	480
[3,]	420	480	540

Altri operatori e funzioni per le matrici



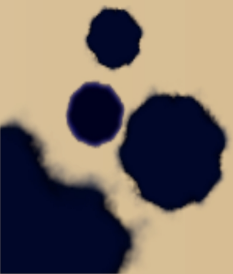
- **diag(x)** restituisce
 - › se x è uno *scalare*, ritorna una matrice identità di dimensione $\text{floor}(x)$
 - › se x è un *vettore*, ritorna la matrice diagonale con x sulla diagonale principale
 - › se x è una *matrice*, ne ritorna la diagonale principale
- **solve(A,b)**
 - › La funzione *solve* consente di invertire una matrice e può essere usata per risolvere sistemi lineari.
 - › Ad esempio, se A è una matrice quadrata non singolare e b un vettore compatibile con A , allora $x \leftarrow \text{solve}(A,b)$ risolve il sistema lineare $Ax=b$
- Esistono inoltre funzioni per il calcolo degli autovalori ed autovettori di matrici simmetriche, per il calcolo della *svd* di una matrice e molte altre



Accesso agli elementi di matrici



- Le regole di accesso per le matrici seguono quelle già viste per i vettori
- Bisogna considerare però l'esistenza di più indici e quindi la possibilità di utilizzare un vettore per ogni dimensione:
 - Vettori di indici interi positivi
 - Vettore di indici interi negativi
 - Vettore di indici logici
 - Vettori di indici a caratteri
- Si utilizza quindi un vettore di indici per ogni dimensione dell'array



Esempi di accesso agli elementi di una matrice

```
> m <- matrix (1:12,nrow=2)
> m
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  1   3   5   7   9  11
[2,]  2   4   6   8  10  12
> m[1,2]
[1] 3
> m [1,3:4]      # riga 1, col. 3 e 4
[1] 5 7
> m [1:2,4:6]   # riga 1 e 2, col. 4,5,6
      [,1] [,2] [,3]
[1,]  7   9  11
[2,]  8  10  12
> m[1,]        # tutti gli el. della riga 1
[1] 1 3 5 7 9 11
> m[,3]       # tutti gli el. Della col. 3
[1] 5 6
```

Esempi di accesso agli elementi di una matrice

```
> m[, -4] # esclusione el. nella 4° colonna
      [,1] [,2] [,3] [,4] [,5]
[1,]  1   3   5   9  11
[2,]  2   4   6  10  12

> m[m>4] # elementi > 4 (si ottiene un vettore)
[1] 5 6 7 8 9 10 11 12

> rownames(m) <- c("r1", "r2")
> colnames(m) <- paste("c", 1:6, sep=" ")

> m
      c1  c2  c3  c4  c5  c6
r1  1   3   5   7   9  11
R2  2   4   6   8  10  12

> m["r1", "c2"] # vettori indice a caratteri
[1] 3

> m["r1", ]
c1 c2 c3 c4 c5 c6
 1  3  5  7  9 11
```

Array e matrici come generalizzazioni multidimensionali di vettori

- In R si possono costruire array di dimensione arbitraria (limitatamente alle disponibilità di memoria) con il comando **array**
- Le matrici sono a tutti gli effetti array bidimensionali
- Sugli array sono applicabili le medesime operazioni di accesso e modifica (estese a più dimensioni) utilizzabili per i vettori

```
> a <- array(0, dim=c(2, 3, 2))
> a[1,2,1] ← 6
, , 1
  [,1] [,2] [,3]
[1,]  0  6  0
[2,]  0  0  0
, , 2
  [,1] [,2] [,3]
[1,]  0  0  0
[2,]  0  0  0
```

Comando *table*

- In R il comando *table* si usa per per generare distribuzioni di frequenze assolute semplici (una sola variabile) o doppie (due variabili)
 - la funzione può essere applicata a qualsiasi tipo di variabile: qualitative (fattori, li vedremo) , e quantitative discrete/continue
- Frequenze assolute semplici

```
> day_off <- c(2, 2, 0, 0, 5, 8, 3, 4, 1, 0,      0, 7, 1, 7, 1, 5, 4, 0, 4, 0,  
              1, 8, 9, 7, 0, 1, 7, 2, 5, 5,      4, 3, 3, 0, 0, 2, 5, 1, 3, 0,  
              1, 0, 2, 4, 5, 0, 5, 7, 5, 1)
```

```
> table(day_off)
```

```
day_off  
 0    1    2    3    4    5    7    8    9  
12    8    5    4    5    8    5    2    1
```

Comando *table*

- La corrispondente distribuzione di frequenze relative si ottiene applicando la funzione *prop.table* alla distribuzione di frequenze assolute:

```
> options(digits=1)
> prop.table(day_off)
```

```
[1] 0.013 0.013 0.000 0.000 0.033 0.053 0.020 0.027 0.007 0.000 0.000 0.047
[13] 0.007 0.047 0.007 0.033 0.027 0.000 0.027 0.000 0.007 0.053 0.060 0.047
[25] 0.000 0.007 0.047 0.013 0.033 0.033 0.027 0.020 0.020 0.000 0.000 0.013
[37] 0.033 0.007 0.020 0.000 0.007 0.000 0.013 0.027 0.033 0.000 0.033 0.047
[49] 0.033 0.007
```

- Mentre otteniamo la tabella delle frequenze relative con

```
> rel.freq <- table(day_off) / length(day_off)
```

Comando *table*

- Supponiamo di dividere i lavoratori in maschi e femmine

```
> sesso<-c(rep(c("M"), 20), rep(c("F"), 30))
```

- Per determinare distribuzioni di frequenza doppie, si utilizza ancora il comando `table`

```
> table(day_off, sesso)
```

```
      sesso
day_off  F  M
0         6  6
1         5  3
2         3  2
3         3  1
4         2  3
5         6  2
7         3  2
8         1  1
9         1  0
```

Installazione ed uso del package *xtable*

- Utile per generare codice il codice latex corrispondente ad una tabella

```
> install.packages("xtable")  
> library(xtable) #  
   caricamento libreria
```

```
> xtable(table(day_off))  
% latex table generated in R  
3.2.2 by xtable 1.7-4 package  
% Mon Oct 26 16:19:50 2015  
\begin{table}[ht]  
\centering  
\begin{tabular}{rr}  
  \hline  
  & day\_off \\  
  \hline  
0 & 12 \\  
 1 & 8 \\  
 2 & 5 \\  
 3 & 4 \\  
 4 & 5 \\  
 5 & 8 \\  
 7 & 5 \\  
 8 & 2 \\  
 9 & 1 \\  
  \hline  
\end{tabular}  
\end{table}
```


Installazione ed uso del package *xtable*



```
> xtable(table(day_off, sesso))
% latex table generated in R 3.2.2
by xtable 1.7-4 package
% Mon Oct 26 16:20:09 2015
\begin{table}[ht]
\centering
\begin{tabular}{rrr}
\hline
& F & M \\
\hline
0 & 6 & 6 \\
1 & 5 & 3 \\
2 & 3 & 2 \\
3 & 3 & 1 \\
4 & 2 & 3 \\
5 & 6 & 2 \\
7 & 3 & 2 \\
8 & 1 & 1 \\
9 & 1 & 0 \\
\hline
\end{tabular}
\end{table}
```



Esercizi

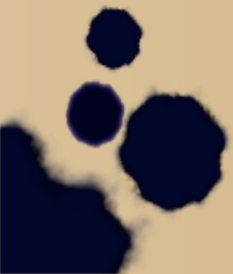


1) Siano dati i seguenti studenti con relativi punteggi di esame:

- Nomi : Luigi, Francesca, Carlo, Alberto, Nicolò, Giovanna
- Voti : 28, 18, 25, 21, 30 , 22
- # ripetizioni esame : 3, 2, 1, 1, 2, 2

- a) Inserire i dati in una matrice R , avendo sulle righe i nomi, e sulle colonne voti e numero di ripetizioni dell'esame
- b) Assegnare sia i nomi di riga che di colonna
- c) Determinare (non a mano) il numero di studenti nelle tre fasce di voto: 18-22, 23-26, 27-30
- d) Aggiungere una colonna “fascia” alla matrice, in cui inseriamo 3 per gli studenti in fascia 18-22, 2 per gli studenti in fascia 23-26 e 1 per quelli in fascia 27-30
- e) Stampare le frequenze relative delle fasce di studenti

CONTINUA....

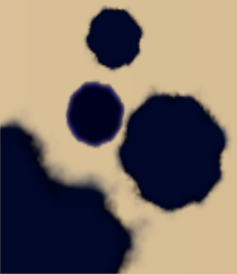


Continuazione esercizio 1



Siano dati i seguenti studenti con relativi punteggi di esame:

- Nomi : Luigi, Francesca, Carlo, Alberto, Nicolò, Giovanna
 - Voti : 28, 18, 25, 21, 30 , 22
 - # ripetizioni esame : 3, 2, 1, 1, 2, 2
- f) Determinare la media campionaria dei voti e del numero di ripetizioni d'esame
- g) Verificare che la somma degli scarti dalla media sia 0
- h) Determinare la varianza campionaria dei voti e del numero di ripetizioni
- i) Esiste una correlazione positiva tra i voti ed il numero di ripetizioni?



Esercizi

- 2) Costruire due matrici M ed N che abbiano entrambe 5 colonne. Costruire, se possibile, tramite `rbind` una matrice di 5 colonne che abbia come righe le righe di entrambe le matrici. Utilizzando M ed N , è possibile costruire una matrice tramite `cbind`?
- 3) Date due matrici quadrate A e B di dimensione 3×3 costituite da un campione di numeri estratti casualmente senza riposizionamento tra i primi 50 interi, calcolarne il prodotto elemento per elemento e “righe per colonne”. Scelto un vettore b di 3 elementi, risolvere il sistema lineare $Ax=b$ che ne deriva?