

# Matrici

Le matrici sono strutture di dati a due dimensioni, composte da un solo tipo di elementi. Per costruire una matrice, utilizziamo la funzione `matrix()`, indicando un vettore di valori, e il numero di righe e di colonne:

```
matrix(c(5, 8, 10, 34, 7, 5),  
       nrow = 2,  
       ncol = 3)
```

Possiamo anche scrivere direttamente:

```
matrix(c(5, 8, 10, 34, 7, 5), 2, 3)
```

```
      [,1] [,2] [,3]  
[1,]    5   10    7  
[2,]    8   34    5
```

Una matrice può essere composta anche di altri tipi di valori:

```
# una matrice di elementi di testo  
matrix(letters[1:8], 4, 2)
```

```
      [,1] [,2]  
[1,] "a"  "e"  
[2,] "b"  "f"  
[3,] "c"  "g"  
[4,] "d"  "h"
```

Come si è visto nei due esempi precedenti, gli elementi vengono disposti in sequenza e per colonna. Per disporli per riga, usiamo l'argomento `byrow = TRUE`:

```
matrix(c(5, 8, 10, 34, 7, 5), 2, 3,  
       byrow = TRUE)
```

```
      [,1] [,2] [,3]  
[1,]    5    8   10  
[2,]   34    7    5
```

Costruiamo la matrice M:

```
M <- matrix(c(5, 8, 10, 34, 7, 5), 2, 3)  
# attributi della matrice  
attributes(M)
```

```
$dim  
[1] 2 3
```

## rownames e colnames

Le matrici hanno nomi di riga (`rownames`), e nomi di colonna (`colnames`). Attribuiamo un nome alle colonne e alle righe della matrice:

```
colnames(M) <- c("C1", "C2", "C3")  
rownames(M) <- c("R1", "R2")  
M
```

```
   C1 C2 C3  
R1  5 10  7  
R2  8 34  5
```

Controlliamo ora gli attributi della matrice:

```
attributes(M)
```

```
$dim  
[1] 2 3  
  
$dimnames  
$dimnames[[1]]  
[1] "R1" "R2"  
  
$dimnames[[2]]  
[1] "C1" "C2" "C3"
```

## Operazioni con matrici

Le operazioni sulle matrici vengono eseguite come per i vettori:

```
# operazioni sulle matrici  
M + 2
```

```
   C1 C2 C3  
R1  7 12  9  
R2 10 36  7
```

Alle matrici si applicano però anche gli operatori del calcolo matriciale:

```
# trasposta
t(M)
```

```
  R1 R2
C1  5  8
C2 10 34
C3  7  5
```

## Dataframe e matrici

Un dataframe composto di elementi dello stesso tipo (tipicamente, numerici), può facilmente essere trasformato in una matrice:

```
as.matrix(cars)
```

```
      speed dist
[1,]     4    2
[2,]     4   10
[3,]     7    4
[4,]     7   22
[5,]     8   16
[6,]     9   10
[7,]    10   18
[8,]    10   26
[9,]    10   34
[10,]   11   17
... 
```

I nomi di colonna rimangono.

In molti casi, questa trasformazione non è neanche necessaria, in quanto molti metodi si applicano tanto alle matrici, quanto ai dataframe, in funzione del tipo di vettori di cui sono composti:

```
# trasposta
t(cars)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
[ ,13]
speed    4    4    7    7    8    9   10   10   10   11   11   12
12
dist     2   10    4   22   16   10   18   26   34   17   28   14
20
... 
```

# Funzioni

Molte funzioni si applicano dunque tanto alle matrici quanto ai dataframe:

Tab. 1: Matrici:funzioni

<code>matrix()</code>	
<code>is.matrix()</code> , <code>as.matrix()</code>	
<code>rownames()</code> , <code>colnames()</code>	nomi di righe e colonne
<code>rbind()</code> , <code>cbind()</code> ( <a href="#">vedi</a> )	unire righe e colonne
<code>rowSums()</code> , <code>colSums()</code>	totali di riga e colonna
<code>rowMeans()</code> , <code>colMeans()</code>	medie di riga e colonna

## Script di esempio

E' possibile scaricare ed eseguire lo script dell'esempio:

[es\\_matrici.R](#)

```
# matrice
M <- matrix(c(5, 8, 10, 34, 7, 5), 2, 3)

# nomi di riga e di colonna
colnames(M) <- c("C1", "C2", "C3")
rownames(M) <- c("R1", "R2")
M
attributes(M)

# operazioni sulle matrici
M + 2
# trasposta
t(M)

# dataframe come matrice
as.matrix(cars)

t(cars)
```

[Concetti di base](#), [Oggetti](#), [Gestione dei dati](#)

From:

<https://www.agnesevardanega.eu/wiki/> - **Ricerca Sociale con R**

Permanent link:

[https://www.agnesevardanega.eu/wiki/r/concetti\\_di\\_base/matrici](https://www.agnesevardanega.eu/wiki/r/concetti_di_base/matrici)

Last update: **15/10/2025 08:03**

