

Manuale Base R

Antonio Calcagnì

Claudio Zandonella Callegher

October 9, 2019

Contents

1	Scrittura di espressioni	2
2	Vettori	4
2.1	Creazione di vettori	4
2.2	Selezione elementi di un vettore	4
2.3	Funzioni ed operazioni tra vettori	5
3	Matrici	6
3.1	Creazione di matrici	6
3.2	Selezione di elementi di una matrice	6
3.3	Funzioni ed operazioni tra matrici	7
4	DataFrames	9
4.1	Creazione di DataFrames	9
4.2	Selezione di elementi di un DataFrame	10
4.3	Funzioni con DataFrames	11
5	Liste	13
5.1	Creazione di Liste	13
5.2	Selezione di elementi di una lista	13
6	Tipi di vettori	15
6.1	Vettori numerici	15
6.2	Vettori logici	15
6.3	Vettori di caratteri	16
6.4	Fattori	16

1 Scrittura di espressioni

R è un'ottima calcolatrice.

Table 1: Principali funzioni matematiche in R

<code>x + y</code>	Addizione	<code>> 5 + 3</code> [1] 8
<code>x - y</code>	Sottrazione	<code>> 7 - 2</code> [1] 5
<code>x * y</code>	Moltiplicazione	<code>> 4 * 3</code> [1] 12
<code>x / y</code>	Divisione	<code>> 8 / 3</code> [1] 2.666667
<code>x %% y</code>	Resto della divisione	<code>> 7 %% 5</code> [1] 2
<code>x %/% y</code>	Divisione intera	<code>> 7 %/% 5</code> [1] 1
<code>abs (x)</code>	Valore assoluto	<code>> abs(3-5^2)</code> [1] 22
<code>sign (x)</code>	Segno di un'espressione	<code>> sign(-8)</code> [1] -1
<code>sqrt (x)</code>	Radice quadrata	<code>> sqrt(25)</code> [1] 5
<code>log (x)</code>	Logaritmo naturale	<code>> log(10)</code> [1] 2.302585
<code>exp (x)</code>	Esponenziale	<code>> exp(1)</code> [1] 2.718282
<code>sin (x)</code> <code>cos (x)</code> <code>tan (x)</code> <code>asin (x)</code> <code>acos (x)</code> <code>atan (x)</code>	Funzioni trigonometriche	<code>> sin(pi/2)</code> [1] 1 <code>> cos(pi/2)</code> [1] 6.123234e-17
<code>factorial (x)</code>	Fattoriale	<code>> factorial(6)</code> [1] 720
<code>choose (n, k)</code>	Coefficiente combinatorio	<code>> choose(5,3)</code> [1] 10

Esercizi

Calcola il risultato delle seguenti espressioni utilizzando R.

$$1) \frac{(45 + 21)^3 + \frac{3}{4}}{\sqrt{32 - \frac{12}{17}}}$$

$$2) \frac{\sqrt{7 - \pi}}{3 (45 - 34)}$$

$$3) \sqrt[3]{12 - e^2} + \ln(10\pi)$$

$$4) \frac{\sin(\frac{3}{4}\pi)^2 + \cos(\frac{3}{2}\pi)}{\log_7 e^{\frac{3}{2}}}$$

$$5) \frac{\sum_{n=1}^{10} n}{10}$$

Note:

- In R la radice quadrata si ottiene con la funzione `sqrt()` mentre per radici di indici diversi si utilizza la notazione esponenziale ($\sqrt[3]{x}$ è dato da `x^(1/3)`).
- Il valore di π si ottiene con `pi`.
- Il valore di e si ottiene con `exp(1)`.
- In R per i logaritmi si usa la funzione `log(x, base=a)`, di base viene considerato il logaritmo naturale.

2 Vettori

2.1 Creazione di vettori

In R per definire un vettore si utilizza il comando `nome_vettore <- c(oggetti)`. Ricorda che gli elementi devono essere separati da una virgola.

Esercizi

1. Crea il vettore `x` contenente i numeri 4, 6, 12, 34, 8.
2. Crea il vettore `y` contenente tutti i numeri pari compresi tra 1 e 25 (`?seq()`).
3. Crea il vettore `z` contenente tutti i primi 10 multipli di 7 partendo da 13 (`?seq()`).
4. Crea il vettore `s` in cui le lettere "A", "B" e "C" vengono ripetute nel medesimo ordine 4 volte (`?rep()`).
5. Crea il vettore `t` in cui le letter "A", "B" e "C" vengono ripetute ognuna 4 volte (`?rep()`).

2.2 Selezione elementi di un vettore

In R per selezionare gli elementi di un vettore si deve indicare all'interno delle parentesi quadre la **posizione degli elementi** da selezionare, non il valore dell'elemento stesso

`nome_vettore[indice_posizione]`

In alternativa si può definire la condizione logica che gli elementi che si vogliono selezionare devono rispettare.

Table 2: Operatori logici in R

<code>x == y</code>	Uguale	<code>> 5==3</code> [1] FALSE
<code>x != y</code>	Diverso	<code>> 5!=3</code> [1] TRUE
<code>x >= y</code>	Maggiore o uguale	<code>> 5>=3</code> [1] TRUE
<code>x > y</code>	Maggiore	<code>> 13>7</code> [1] TRUE
<code>x <= y</code>	Minore o uguale	<code>> 5<=3</code> [1] FALSE
<code>x < y</code>	Minore	<code>> 13<7</code> [1] FALSE

Per concatenare più operazioni logiche si possono usare la congiunzione logica “e” (`&`) o la disgiunzione inclusiva “o” (`|`). Per **eliminare degli elementi** da un vettore si utilizza all'interno

delle parentesi quadre l'operatore “-” insieme agli indici di posizione degli elementi da eliminare (esempio: `x[c(-2,-4)]` oppure `x[-c(2,4)]`).

Esercizi

1. Del vettore `x` seleziona il 2°, 3° e 5° elemento.
2. Del vettore `y` seleziona tutti i valori minori di 13 o maggiori di 19.
3. Del vettore `z` seleziona tutti i valori compresi tra 24 e 50.
4. Elimina dal vettore `z` i valori 28 e 42.
5. Del vettore `s` seleziona tutti gli elementi uguali ad “A”.
6. Del vettore `t` seleziona tutti gli elementi diversi da “B”.

2.3 Funzioni ed operazioni tra vettori

Per compiere operazioni tra vettori è necessario che essi abbiano identica lunghezza.

Table 3: Operazioni con vettori

<code>nuovo_vettore <- c(vettore1,vettore2)</code>	Per unire più vettori in un unico vettore
<code>length(nome_vettore)</code>	Per valutare il numero di elementi contenuti in un vettore
<code>vettore1 + vettore2</code> <code>vettore1 - vettore2</code>	Somma di due vettori Differenza tra due vettori
<code>vettore1 * vettore2</code> <code>vettore1 / vettore2</code>	Prodotto tra due vettori Rapporto tra due vettori

Nota: In R il prodotto e rapporto tra vettori sono eseguiti elemento per elemento (al contrario di molti altri software).

Esercizi

1. Crea il vettore `j` unendo i vettori `x` ed `z`.
2. Elimina gli ultimi tre elementi del vettore `j` e controlla che i vettori `j` e `y` abbiano la stessa lunghezza.
3. Calcola la somma tra i vettori `j` e `y`.
4. Moltiplica il vettore `z` per una costante `k=3`.
5. Calcola il prodotto tra i primi 10 elementi del vettore `y` ed il vettore `z`.

3 Matrici

3.1 Creazione di matrici

In R per definire una matrice di n righe e s colonne si utilizza il comando

```
nome_matrice <- matrix(data, nrow=n, ncol=s, byrow=FALSE)
```

Nota: Di default R riempie la matrice per colonne, impostando `byrow = TRUE` si riempie per righe.

Esercizi

1. Crea la matrice A così definita:

2	34	12	7
46	93	27	99
23	38	7	04

2. Crea la matrice B contenente tutti i primi 12 numeri dispari disposti su 4 righe e 3 colonne.
3. Crea la matrice C contenente i primi 12 multipli di 9 disposti su 3 righe e 4 colonne.
4. Crea la matrice D formata da 3 colonne in cui le lettere "A", "B" e "C" vengano ripetute 4 volte ciascuna rispettivamente nella prima, seconda e terza colonna.
5. Crea la matrice E formata da 3 righe in cui le lettere "A", "B" e "C" vengano ripetute 4 volte ciascuna rispettivamente nella prima, seconda e terza riga.

3.2 Selezione di elementi di una matrice

In R per selezionare gli elementi di matrice si deve indicare all'interno delle parentesi quadre l'indice di riga e l'indice di colonna (**separati da virgola**) degli elementi da selezionare oppure la condizione logica che devono rispettare.

```
nome_matrice[indice_riga , indice_colonna]
```

Nota: per selezionare tutti gli elementi di una data riga o di una data colonna basta lasciare vuoto rispettivamente l'indice di riga o l'indice di colonna.

Esercizi

1. Utilizzando gli indici di riga e di colonna seleziona il numero 27 della matrice A
2. Seleziona gli elementi compresi tra la seconda e quarta riga, seconda e terza colonna della matrice B.
3. Seleziona solo gli elementi pari della matrice A (Nota: utilizza l'operazione resto %%).
4. Elimina dalla matrice C la terza riga e la terza colonna.
5. Seleziona tutti gli elementi della seconda e terza riga della matrice B.
6. Seleziona tutti gli elementi diversi da "B" appartenenti alla matrice D.

3.3 Funzioni ed operazioni tra matrici

Table 4: Operazioni con matrici

<code>nuova_matrice <- cbind(matrice1,matrice2)</code>	Per unire due matrici creando nuove colonne (le matrici devono avere lo stesso numero di righe)
<code>nuova_matrice <- rbind(matrice1,matrice2)</code>	Per unire due matrici creando nuove righe (le matrici devono avere lo stesso numero di colonne)
<code>nrow(nome_matrice)</code>	Per valutare il numero di righe della matrice
<code>ncol(nome_matrice)</code>	Per valutare il numero di colonne della matrice
<code>dim(nome_matrice)</code>	Per valutare la dimensione della matrice (righe e colonne)
<code>t(nome_matrice)</code>	Per ottenere la trasposta della matrice
<code>diag(nome_matrice)</code>	Ottenere un vettore con gli elementi della diagonale della matrice
<code>det(nome_matrice)</code>	Ottenere il determinante della matrice (la matrice deve essere quadrata)
<code>solve(nome_matrice)</code>	Ottenere l'inversa della matrice (la matrice deve essere quadrata)
<code>colnames(nome_matrice)</code> <code>rownames(nome_matrice)</code>	Nomi delle colonne della matrice Nomi delle righe della matrice
<code>matrice1 + matrice2</code> <code>matrice1 - matrice2</code>	Somma elemento per elemento di due matrici Differenza elemento per elemento tra due matrici
<code>matrice1 * matrice2</code> <code>matrice1 / matrice2</code>	Prodotto elemento per elemento tra due matrici Rapporto elemento per elemento tra due matrici
<code>matrice1 %*% matrice2</code>	Prodotto matriciale

Note:

- Per il significato di determinante di una matrice considera: <https://it.wikipedia.org/wiki/Determinante>.

- Per il significato di matrice inversa considera: https://it.wikipedia.org/wiki/Matrice_invertibile.
- Per compiere operazioni elemento per elemento tra due matrici, esse devono avere la stessa dimensione.
- Per compiere il prodotto matriciale il numero di colonne della prima matrice deve essere uguale al numero di righe della seconda matrice (vedi https://it.wikipedia.org/wiki/Moltiplicazione_di_matrici).
- E' possibile assegnare nomi alle colonne e righe di una matrice rispettivamente attraverso i comandi:

```
colnames(nome_matrice)<-c("nome_1",...,"nome_s")
rownames(nome_matrice)<-c("nome_1",...,"nome_n")
```

Esercizi

1. Crea la matrice G unendo alla matrice A le prime due colonne della matrice C.
2. Crea la matrice H unendo alla matrice C le prime due righe della matrice trasposta di B.
3. Ridefinisci la matrice A eliminando la seconda colonna. Ridefinisci la matrice B eliminando la prima riga. Verifica che le matrici così ottenute abbiano la stessa dimensione.
4. Commenta i differenti risultati che otteniamo nelle operazioni $A*B$, $B*A$, $A\%*B$ e $B\%*A$.
5. Assegna i seguenti nomi alle colonne e alle righe della matrice C: "col_1", "col_2", "col_3", "col_4", "row_1", "row_2", "row_3".

4 DataFrames

4.1 Creazione di DataFrames

Uno degli oggetti più utilizzati in R sono i DataFrames. I DataFrames permettono di raccogliere all'interno di uno stesso oggetto vettori di diverso tipo (i.e., vettori numerici, logici, fattori o stringhe di caratteri). Per questo motivo, i DataFrames sono utili per riportare tutti i dati riguardanti le diverse variabili misurate in un esperimento.

In genere ogni riga di un DataFrames rappresenta una singola osservazione e nelle colonne sono riportate i vari valori delle variabili misurate.

Esistono due formati principali di DataFrames:

- **Wide:** ogni singola riga rappresenta un soggetto e ogni sua risposta o variabile misurata sarà riportata in una diversa colonna.

	Id	age	sex	item_1	item_2	item_3
1	subj_1	21	F	2	0	2
2	subj_2	23	M	1	2	0
3	subj_3	19	F	1	1	1

- **Long:** ogni singola riga rappresenta una singola osservazione. Quindi i dati di ogni soggetto saranno riportati su più righe e le variabili che non cambiano tra le osservazioni saranno ripetute.

	Id	age	sex	item	response
1	subj_1	21	F	1	2
2	subj_1	21	F	2	1
3	subj_1	21	F	3	1
4	subj_2	23	M	1	0
5	subj_2	23	M	2	2
6	subj_2	23	M	3	1
7	subj_3	19	F	1	2
8	subj_3	19	F	2	0
9	subj_3	19	F	3	1

In R per definire un DataFrame si utilizza il comando:

```
nome_DataFrame <- data.frame(variab_1=c(...), ..., variab_s=c(...))
```

All'interno vanno riportate le variabili che si vogliono inserire separate da virgole. Ogni variabile deve avere la **stessa lunghezza**.

Nota: di default R considera una variabile stringa all'interno di un DataFrame come una variabile categoriale. E' possibile cambiare questa opzione specificando `stringsAsFactors=FALSE`.

Esercizi

Esercizi:

1. Crea il dataframe `data_wide` riportato precedentemente.
2. Crea il dataframe `data_long` riportato precedentemente.

4.2 Selezione di elementi di un DataFrame

In R per selezionare gli elementi di un DataFrame si può, analogamente alle matrici, indicare all'interno delle parentesi quadre l'indice di riga e l'indice di colonna (**separati da virgola**).

```
nome_DataFrame[indice_riga , indice_colonna]
```

Per accedere ad una specifica variabile del DataFrame è possibile utilizzare l'operatore "\$":

```
nome_DataFrame$nome_variabile
```

Per quanto riguarda l'indice di riga è possibile definire una condizione logica rispetto ad una variabile, mentre per l'indice di colonna si può indicare il nome delle variabili:

```
nome_DataFrame[condizione_logica , c("variabile_1", ..., "variabile_s")]
```

Nota: per selezionare tutti gli elementi di una data riga basta lasciare vuoto l'indice di colonna.

Esempio: `data_wide[data_wide$sex=="F", c("Id","age")]`

Esercizi

1. Utilizzando gli **indici numerici** di riga e di colonna seleziona i dati del soggetto `subj_2` riguardanti le variabili `item` e `response` dal DataFrame `data_long`.
2. Compila la stessa selezione dell'esercizio precedente usando però questa volta una condizione logica per gli indici di riga e indicando direttamente il nome delle variabili per gli indici di colonna.
3. Considerando il DataFrame `data_wide` seleziona le variabili `Id` e `sex` dei soggetti che hanno risposto 1 alla variabile `item_1`.
4. Considerando il DataFrame `data_long` seleziona solamente i dati riguardanti le ragazze con età superiore ai 20 anni.
5. Elimina dal DataFrame `data_long` le osservazioni riguardanti il soggetto `subj_2` e la variabile `"sex"`.

4.3 Funzioni con DataFrames

Table 5: Operazioni con DataFrames

<code>nome_DataFrame <- cbind(nome_DataFrame, nuova_variabile) nome_DataFrame\$nome_variabile <- dati</code>	Per aggiungere una nuova variabile al DataFrame (deve avere lo stesso numero di righe)
<code>nome_DataFrame <- rbind(nome_DataFrame, nuovi_dati)</code>	Per aggiungere delle osservazioni (i nuovi dati devono essere coerenti con la struttura del DataFrame)
<code>nrow(nome_DataFrame)</code>	Per valutare il numero di osservazioni del DataFrame
<code>ncol(nome_DataFrame)</code>	Per valutare il numero di variabili del DataFrame
<code>colnames(nome_DataFrame) names(nome_DataFrame)</code>	Nomi delle colonne del DataFrame
<code>rownames(nome_DataFrame)</code>	Nomi delle righe del DataFrame

Nota: E' possibile assegnare nomi alle colonne e righe di un DataFrame allo stesso modo delle matrici, attraverso i comandi

```
colnames(nome_DataFrame)<-c("nome_1",...,"nome_s")  
names(nome_DataFrame)<-c("nome_1",...,"nome_s")  
rownames(nome_DataFrame)<-c("nome_1",...,"nome_n")
```

Esercizi

1. Aggiungi sia al DataFrame `data_wide` che `data_long` la variabile numerica `"memory_pre"`.

```
      Id memory_pre  
1 subj_1          3  
2 subj_2          2  
3 subj_3          1
```

2. Aggiungi sia al DataFrame `data_wide` che `data_long` la variabile categoriale `"gruppo"`.

```
      Id      gruppo  
1 subj_1 trattamento  
2 subj_2 trattamento  
3 subj_3 controllo
```

3. Aggiungi al DataFrame `data_wide` i dati del soggetto `subj_4` e `subj_5`.

```
      Id age sex item_1 item_2 item_3 memory_pre      gruppo  
1 subj_4  25  F      1      0      2          1 trattamento  
2 subj_5  22  M      1      1      0          3 controllo
```

4. Considerando il DataFrame `data_wide` calcola la variabile "memory_post" data dalla somma degli item.
5. Considerando il DataFrame `data_wide` cambia i nomi delle variabili `item_1`, `item_2` e `item_3` rispettivamente in `problem_1`, `problem_2` e `problem_3`.

5 Liste

5.1 Creazione di Liste

Le liste sono degli speciali oggi in R che permettono di contenere al loro interno altri oggetti indipendentemente dalla loro tipologia. Possiamo quindi avere nella stessa lista sia vettori, sia matrici sia DataFrames.

In R per definire una lista si utilizza il comando:

```
nome_lista <- list(nome_oggetto_1 = oggetto_1, ..., nome_oggetto_n = oggetto_n)
```

All'interno si possono riportare vari oggetti che si vogliono inserire con i relativi nomi, separati da virgole.

Esercizi

1. Crea la lista `esperimento_1` contenente:

- il DataFrame `data_wide`
- la matrice `A`
- il vettore `x`
- la variabile `info = "Hello world!"`

2. Crea la lista `esperimento_2` contenente:

- il DataFrame `data_long`
- la matrice `C`
- il vettore `y`
- la variabile `info = "Prima raccolta dati"`

5.2 Selezione di elementi di una lista

In R per selezionare gli elementi di una lista si possono usare le doppie parentesi quadre indicando l'indice della posizione dell'oggetto che si vuole selezionare:

```
nome_lista[[indice_posizione]]
```

In alternativa, se i nomi degli oggetti sono stati specificati, è possibile utilizzare l'operatore `"$"` e il nome dell'oggetto da selezionare all'interno della lista:

`nome_lista$nome_oggetto` In seguito per accedere a specifici elementi all'interno degli oggetti si utilizzano le stesse norme precedentemente presentate a seconda del tipo di oggetto.

Esempio: `esperimento_1[[2]][,2]`

```
esperimento_1$data_wide$age
```

Nota: per definire o cambiare i nomi degli oggetti contenuti in una lista è possibile utilizzare la funzione:

```
names(nome_lista) <- c(nome_oggetto_1, ..., nome_oggetto_n)
```

Esercizi

1. Utilizzando gli **indici numerici** di posizione seleziona i dati dei soggetti `subj_1` e `subj_4` riguardanti le variabili `age`, `sex` e `gruppo` dal DataFrame `data_wide` contenuto nella lista `esperimento_1`.
2. Compi la stessa selezione dell'esercizio precedente usando però questa volta il nome dell'oggetto per selezionare il DataFrame dalla lista.
3. Considerando la lista `esperimento_2` seleziona gli oggetti `data_long`, `y` e `info`.
4. Cambia i nomi degli oggetti contenuti nella lista `esperimento_2` rispettivamente in `"dati_esperimento"`, `"matrice_VCV"`, `"codici_Id"` e `"note"`

6 Tipi di vettori

In R ci sono 4 tipi differenti di vettori: numerici, logici, caratteri e fattori.

6.1 Vettori numerici

I vettori numerici sono utilizzati per compiere operazioni aritmetiche, in R sono indicati come `num`. In R ci sono è possibil e specificare se i numeri contenuti nel vettore sono numeri interi, avremmo quindi un vettore di valori interi (indicato in R come `int`). Per fare ciò è possibile aggiungere `L` ad ogni valore numerico nel definire il vettore oppure usare la funzione `as.integer()` per trasformare un vettore numerico in un vettore intero.

Esempio:

```
> x <- c(4L, 6L, 12L, 34L, 8L)
>
> x <- as.integer(c(4, 6, 12, 34, 8))
```

Nota: per trasformare un vettore intero in un vettore numerico è possibile usare la funzione `as.numeric()`.

6.2 Vettori logici

I vettori logici sono formati dai valori `TRUE` e `FALSE`, che possono essere abbreviati rispettivamente in `T` e `F`. In R i vettori logici sono indicati come `logi`. In genere, i vettori logici sono il risultato delle operazioni in cui viene chiesto ad R di valutare la condizione logica di una proposizione.

```
> x>10
[1] FALSE FALSE  TRUE  TRUE FALSE
```

Nota: in R, come in molti altri software di programmazione, `TRUE` assume il valore numerico 1 e `FALSE` assume il valore 0.

```
> sum(x>10)
[1] 2
```

E' possibile trasformare un vettore numerico in un vettor logico attraverso la funzione `as.logical()`, gli 0 assumeranno il valore `FALSE` mentre qualsiasi altro numero assumerà il valore `TRUE`.

```
> as.logical(c(1,0,.034,-1,0,8))
[1] TRUE FALSE  TRUE  TRUE FALSE  TRUE
```

6.3 Vettori di caratteri

I vettori di caratteri contengono stringhe di caratteri e sono indicati in R con `chr`. Non è possibile eseguire operazioni aritmetiche con vettori di caratteri ma solo valutare se due stringhe sono uguali o differenti.

```
> j<-c("Hello","World","hello","world")
> j=="hello"

[1] FALSE FALSE  TRUE FALSE
```

Per trasformare un vettore qualsiasi in una vettore di caratteri è possibile usare la funzione `as.character()`.

```
> as.character(x)

[1] "4"  "6"  "12" "34" "8"

> as.character(x>10)

[1] "FALSE" "FALSE" "TRUE"  "TRUE"  "FALSE"
```

6.4 Fattori

I fattori sono utilizzati per definire delle variabili categoriali, sono indicati in R con `Factor`. Per creare una variabile categoriale in R si utilizza la funzione:

```
nome_variabile<-factor(c(..., data, ...), levels=c(...))
```

L'opzione `levels=c(...)` è usata per specificare quali sono i possibili livelli della variabile categoriale. E' possibile modificare o aggiungere nuovi livelli della variabile anche in un secondo momento utilizzando la funzione:

```
levels(nome_fattore)<- c(..., nuovi_livelli, ...)
```

Nota: nel creare un fattore R associa ad ogni livello un valore in ordine crescente e assegna agli elementi del vettore il loro valore numerico a seconda del proprio livello. Pertanto se un fattore è trasformato in un vettore numerico vengono restituiti tali valori numerici e non i livelli anche nel caso fossero dei numeri. Prendiamo per esempio la variabile `anni_istruzione`

```
> anni_istruzione<-factor(c(11,8,4,8,11,4,11,8))
> anni_istruzione

[1] 11 8  4  8 11 4 11 8
Levels: 4 8 11

> as.numeric(anni_istruzione)

[1] 3 2 1 2 3 1 3 2
```

Per riottenere gli estti valori numerici è necessario eseguire


```
> as.numeric(as.character(anni_istruzione))  
[1] 11  8  4  8 11  4 11  8
```

Esercizi

1. Crea la variabile categoriale **sex** così definita:

```
[1] M F M F M F F F M  
Levels: F M
```

2. Rinomina i livelli della variabile **sex** rispettivamente in "donne" e "uomini".

3. Crea la variabile categoriale **intervento** così definita:

```
[1] CBT          Psicanalisi CBT          Psicanalisi CBT          Psicanalisi  
[7] Controllo    Controllo    CBT  
Levels: CBT Controllo Psicanalisi
```

4. Correggi nella variabile **intervento** la 7° e 8° osservazione con la voce **Farmaci**.
5. Aggiungi alla variabile **intervento** le seguenti nuove osservazioni:

```
[1] "Farmaci"    "Controllo" "Farmaci"
```