

---

Francesca Mazzia  
Dipartimento di Matematica  
Università di Bari

## R: lezione introduttiva

---

R: PSE sviluppato dall R development Core Team. È reperibile gratuitamente dal sito: <http://www.r-project.org/>.

È dotato di un particolare linguaggio di programmazione di tipo interpretativo che consente di sviluppare le proprie applicazioni. È nato come software per l'analisi statistica dei dati, viene molto utilizzato anche per il calcolo scientifico.

R può essere associato con una interfaccia grafica. Per windows suggerisco l'editore **TINN-R** che può essere scaricato da <http://sourceforge.net/projects/tinn-r> e <http://www.sciviews.org/Tinn-R>. Per linux suggerisco RKWard che può essere scaricato da <http://rkwad.sourceforge.net/>.

Lavorando usando la console, al prompt `>` ogni istruzione viene interpretata ed eseguita immediatamente.

`>`

il che significa che il calcolatore è pronto a ricevere le istruzioni R ed ad eseguirle. Se vogliamo eseguire la somma  $3+2$  basta scrivere:

```
> 3 + 2
```

```
[1] 5
```

e premere il tasto invio. Otteniamo immediatamente la risposta.  
e il calcolatore è di nuovo pronto a ricevere un nuovo comando. Le variabili nei linguaggi di programmazione sono dei contenitori di valori. Se vogliamo assegnare il risultato dell'operazione alla variabile di nome `var` basta scrivere:

```
> var <- 3 + 2
```

oppure

```
> var = 3 + 2
```

Questa istruzione si chiama *istruzione di assegnazione*. Per visualizzare il risultato dell'istruzione eseguita si digita il nome della variabile

```
> var
```

```
[1] 5
```

Possiamo assegnare a una variabile il valore di una espressione più complicata. Per far questo dobbiamo sapere a quali simboli corrispondono le principali operazioni aritmetico-logiche. In R questi sono:

+	addizione
-	sottrazione
*	moltiplicazione
/	divisione
^	elevamento a potenza
&	and logico
	or logico
!	not logico
==	uguale
!=	diverso
<	minore
<=	minore uguale
>	maggiore
>=	maggiore uguale

Le espressioni aritmetiche e logiche seguono le classiche regole di precedenza, che è possibile modificare utilizzando le parentesi tonde.

Per avere informazioni su tutte le operazioni che possono essere eseguite in R si può digitare sul pulsante `help.start()`. Compare la finestra di con l'elenco di tutte le informazioni su R divise per capitoli.

Possiamo ora eseguire:

```
> var = 5 * 3/(3 + 2)^2
> var
```

```
[1] 0.6
```

È anche possibile calcolare i valori delle principali funzioni. Ad esempio sono predefinite le funzioni in una variabile:

abs(x) valore assoluto di x,  
log(x) logaritmo naturale,  
exp(x) esponenziale con base e,  
sin(x) seno,  
cos(x) coseno,  
tan(x) tangente.

Anche per queste funzioni è possibile usare il comando help. Si provi ad eseguire:

```
> ?cos
> ?tan
> ?log
```

È possibile quindi eseguire espressioni del tipo:

```
> var = cos((5 * 3/(3 + cos(5)))^log(2) * 100)
> var
```

```
[1] -0.748199
```

Se eseguiamo il comando

```
> options(digits = 4)
```

e rieseguiamo l'operazione

```
> var = cos((5 * 3/(3 + cos(5)))^log(2) * 100)
> var
```

```
[1] -0.7482
```

La variabile `var` viene visualizzata usando 4 cifre dopo la virgola.

Oltre alle variabili, sono di particolare utilità anche le *costanti*. Le costanti sono dei contenitori di valori che non cambiano nel corso dell'esecuzione di un algoritmo. In R vi sono delle costanti predefinite, come, ad esempio, la costante `pi` che contiene il valore di  $\pi$ :

```
> pi
[1] 3.142
```

Le variabili possono essere scalari oppure vettori o matrici. I vettori e le matrici sono variabili che contengono un certo numero di componenti. Ad esempio dopo l'istruzione:

```
> A <- c(3, 5, 6)
> A
[1] 3 5 6
```

la variabile `A` sarà un vettore di tre elementi con `A[1]=3`, `A[2]=5`, `A[3]=6`. Gli elementi di un vettore si indirizzano specificando l'indice tra parentesi quadre. Se vogliamo sapere la somma degli elementi di `A` possiamo eseguire:

```
> A[1] + A[2] + A[3]
[1] 14
oppure
> sum(A)
[1] 14
```

Tutte le variabili già definite, dette variabili globali, possono essere elencate con il comando:

```
> ls()
[1] "A" "var"
```

oppure:

```
> ls.str()
A : num [1:3] 3 5 6
var : num -0.748
```

Per chiudere la sessione di lavoro digitiamo `q()`.