

PICCOLA INTRODUZIONE ALLA PROGRAMMAZIONE IN R

Console: permette di scrivere dei comandi e ricevere un output. Le istruzioni che scriviamo però non rimangono in memoria. Sono solo temporanee.

Script: file in cui possiamo scrivere diverse righe di codice comprendendo variabili, funzioni e altro affinché possa poi riutilizzarlo e modificarlo.

Variabili: sono dei contenitori in cui possiamo mettere numeri, parole, lettere per poi utilizzarli in seguito. Le variabili vengono salvate, quindi, nell'ambiente di lavoro. Usare le variabili ci aiuta a velocizzare le istruzioni che poi andremo ad usare.

Ci sono molti tipi di variabili come:

Ci sono molti tipi di variabili come:																											
variabile numerica/carattere	Singolo valore all'interno della cella (num interi, decimali, lettere, parole, TRUE/FALSE)	<table><tr><td>3</td></tr></table>	3																								
3																											
stringa	insieme di caratteri	<table><tr><td>CIAO</td></tr></table>	CIAO																								
CIAO																											
vettori	insieme di valori dello stesso tipo	<table><tr><td>3</td><td>5</td><td>4</td><td>9</td><td>0</td></tr></table>	3	5	4	9	0																				
3	5	4	9	0																							
matrici	vettore bidimensionale, ovvero una tabella, composta da valori dello stesso tipo	<table><tr><td>3</td><td>5</td><td>4</td><td>9</td><td>0</td></tr><tr><td>2</td><td>1</td><td>1</td><td>6</td><td>8</td></tr><tr><td>1</td><td>2</td><td>2</td><td>0</td><td>4</td></tr><tr><td>5</td><td>5</td><td>7</td><td>9</td><td>0</td></tr><tr><td>3</td><td>8</td><td>4</td><td>7</td><td>3</td></tr></table>	3	5	4	9	0	2	1	1	6	8	1	2	2	0	4	5	5	7	9	0	3	8	4	7	3
3	5	4	9	0																							
2	1	1	6	8																							
1	2	2	0	4																							
5	5	7	9	0																							
3	8	4	7	3																							
lista	insieme di valori anche di tipo diverso compresi anche i vettori (ogni cella è una variabile di tipo vettore)	<table><tr><td>3</td><td>F</td><td>0.1</td><td>9</td><td>SI</td></tr></table>	3	F	0.1	9	SI																				
3	F	0.1	9	SI																							
dataframe	lista bidimensionale, ovvero una tabella, composta da valori che possono essere di diverso tipo es. → 1ª colonna: numeri interi, 2ª colonna: F/M, 3ª colonna : numeri decimali	<table><tr><td>3</td><td>F</td><td>0.1</td><td>9</td><td>SI</td></tr><tr><td>2</td><td>M</td><td>0.4</td><td>6</td><td>SI</td></tr><tr><td>1</td><td>M</td><td>0.5</td><td>0</td><td>NO</td></tr><tr><td>5</td><td>F</td><td>0.9</td><td>9</td><td>NO</td></tr><tr><td>3</td><td>F</td><td>0.3</td><td>7</td><td>SI</td></tr></table>	3	F	0.1	9	SI	2	M	0.4	6	SI	1	M	0.5	0	NO	5	F	0.9	9	NO	3	F	0.3	7	SI
3	F	0.1	9	SI																							
2	M	0.4	6	SI																							
1	M	0.5	0	NO																							
5	F	0.9	9	NO																							
3	F	0.3	7	SI																							

Funzioni: è un insieme di istruzioni attraverso il quale inseriamo degli input per avere un output.

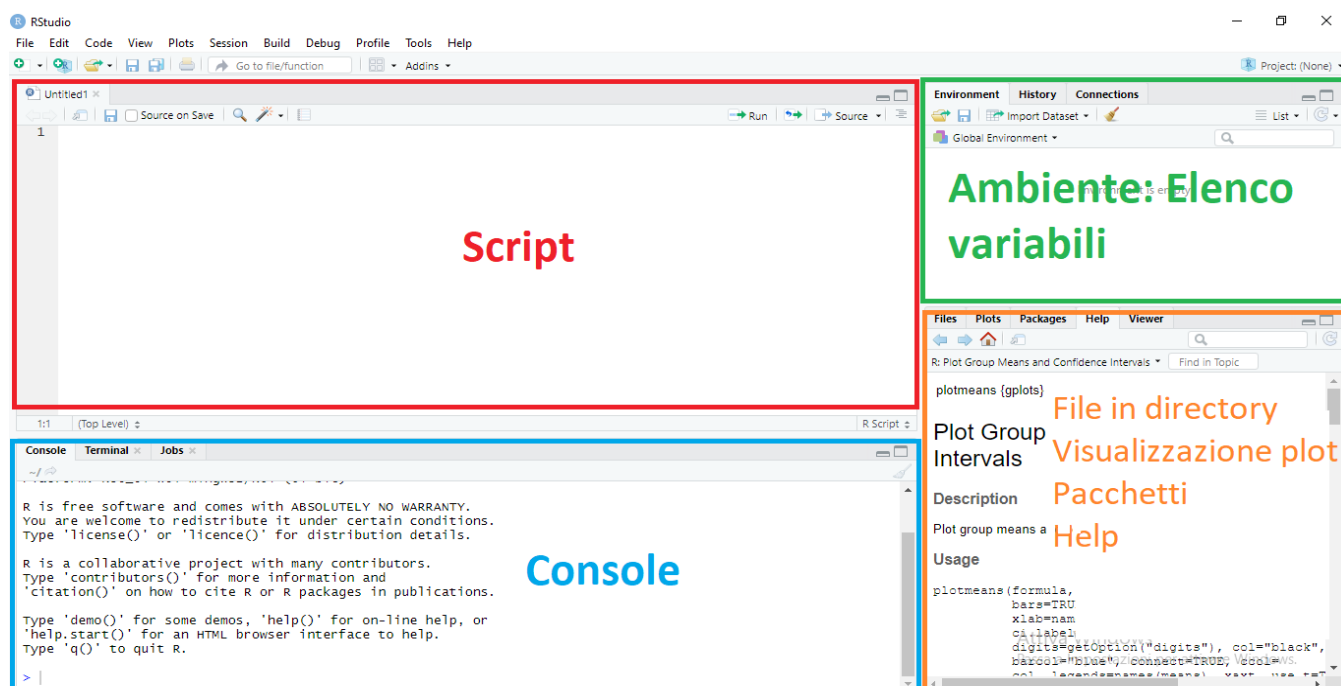
Ogni funzione ha un suo nome(e tra parentesi degli argomenti tra cui input da inserire).

Gli argomenti cambiano in base alla funzione.

Usare ?nomefunzione nella console per vedere l'help e gli argomenti

es. mean(var) → inserendo come input var avremo la media dei valori di var

AMBIENTE R



CONSIGLIO: per imparare meglio e ricordarsi durante le lezioni utilizzare i commenti

Per ogni esercizio scrivere il codice in uno script assieme a dei commenti, spiegando in ogni riga cosa significa l'istruzione. Questo è utile quando si vuole rivedere alcune cose in futuro.

Per fare i commenti basta scrivere # e tutto ciò che viene scritto dopo comparirà verde. Vale solo per la riga in cui si scrive. Appena si va a capo non sarà più un commento

esempio

```
num1 <- 3    #inserisco 3 nella variabile num1  
num2 <- 5    #inserisco 5 nella variabile num2  
somma <- num1 + num2    #la variabile somma contiene la somma di num1 e num2
```

VARIABILI:

Assegnare valore: nome_variabile<-valore

Visualizzare variabile: nome_variabile

Rimuovere contenuto: rm(nome_variabile)

VETTORI:

Creare vettore: nome_variabile ← c(elenco valori) es. var ← c("1","4","7")

Estrarre elementi vettore: nome_variabile[numero]

```
vett[2:4]      # restituisce gli elementi dalla posizione 2 alla 4
vett[c(1,3)]   # restituisce valori posizione 1 e 3
vett[-c(2,3)]  # restituisce tutti i valori meno che valori posizione 2 e 3
```

Alcune funzioni vettore:

- length(vett) #lunghezza array
- sum(vett) # somma di tutti gli elementi
- cumsum(vett) # la somma cumulata
- sum(vett)/length(vett) oppure mean(vett) #media

Altre info: <https://www.datamentor.io/r-programming/vector/>

MATRICI:

Creare matrice:

```
> matrix(1:9, nrow = 3, ncol = 3)
[,1] [,2] [,3]
[1,]  1  4  7
[2,]  2  5  8
[3,]  3  6  9
> # same result is obtained by providing only one dimension
> matrix(1:9, nrow = 3)
[,1] [,2] [,3]
[1,]  1  4  7
[2,]  2  5  8
[3,]  3  6  9
```

Creare matrice con valori per riga:

We can see that the matrix is filled column-wise. This can be reversed to row-wise filling by passing `TRUE` to the argument `byrow`.

```
> matrix(1:9, nrow=3, byrow=TRUE) # fill matrix row-wise
[,1] [,2] [,3]
[1,]  1  2  3
[2,]  4  5  6
[3,]  7  8  9
```

Rinominare Righe e Colonne:

It is possible to name the rows and columns of matrix during creation by passing a 2 element list to the argument `dimnames`.

```
> x <- matrix(1:9, nrow = 3, dimnames = list(c("X","Y","Z"), c("A","B","C")))
> x
  A B C
X 1 4 7
Y 2 5 8
Z 3 6 9
```

Aggiungere righe o colonne:

Another way of creating a matrix is by using functions `cbind()` and `rbind()` as in column bind and row bind.

```
> cbind(c(1,2,3),c(4,5,6))
[,1] [,2]
[1,]  1   4
[2,]  2   5
[3,]  3   6
> rbind(c(1,2,3),c(4,5,6))
[,1] [,2] [,3]
[1,]  1   2   3
[2,]  4   5   6
```

Accedere ai valori: `var[riga,colonna]`

es. `var[1,3]` # restituisce il valore alla riga 1 e colonna 3

`var[1:4,3:6]` # restituisce valore dalle righe 1 a 4 e colonne 3 a 6

Altre funzioni: <https://www.datamentor.io/r-programming/matrix/>

LISTE:

Creare liste: `x <- list("a" = 2.5, "b" = TRUE, "c" = 1:3)`

#avremo così una lista con “a” come numero decimale, “b” come TRUE/FALSE e “c” come numero intero

Vedere struttura: `str(var)`

Come accedere: `var$nome_categoria`

es. `x$a` # restituisce 2.5

`x$b` # restituisce TRUE

`x$c` # restituisce 1,2,3

Altre info: <https://www.datamentor.io/r-programming/list/>

DATASET:

Creare dataset: `> x <- data.frame("SN" = 1:2, "Age" = c(21,15), "Name" = c("John","Dora"))`

aggiungere `stringAsFactors=FALSE` per evitare che crei fattori anziché come vettore

Come accedere: We can use either `[]`, `[[` or `$` operator to access columns of data frame.

```
> x["Name"]
Name
1 John
2 Dora
> x$Name
[1] "John" "Dora"
> x[["Name"]]
[1] "John" "Dora"
> x[[3]]
[1] "John" "Dora"
```

Aggiungere valori:

```
> rbind(x,list(1,16,"Paul"))
  SN Age Name
1   1  20 John
2   2  15 Dora
3   1  16 Paul
```

```
> cbind(x,State=c("NY","FL"))
  SN Age Name State
1   1  20 John   NY
2   2  15 Dora   FL
```

Altre info: <https://www.datamentor.io/r-programming/data-frame/>