

**PROVA R 1 (1° TIPO DI "OGGETTI" = SCALARI)**

```
> 2+3
[1] 5
> 2-3
[1] -1
> 2*3
[1] 6
> 2/3
[1] 0.6666667
> 2^3
[1] 8
>
```

**PROVA R 2**

```
> local({fn<-choose.files(filters=Filters[c('R','txt','All'),],index=4)
+ file.show(fn,header=fn,title=")})
> 4^2-3*2
[1] 10
> (4^2)-(3*2)
[1] 10
> 2^-3
[1] 0.125
> -2--3
[1] 1
> -2 - -3
[1] 1
> sqrt(9)
[1] 3
>
```

**SIMBOLO DI ASSEGNAZIONE**

```
> a <- 2
> a
[1] 2
>
```

**MAIUSCOLE E MINUSCOLE (differenza)**

```
> a <- 2
> A
Errore: oggetto "A" non trovato
> a
[1] 2
>
```

**VETTORI (2° TIPO DI "OGGETTI")**

```
> x <- c(2, 5, 9.5, -3)
> x
[1] 2.0 5.0 9.5 -3.0
>
```

**OPERAZIONI ARITMETICHE SUI VETTORI (elemento per elemento)**

```
> x <- c(2, 5, 9.5, -3)
> x
[1] 2.0 5.0 9.5 -3.0
> x+2
[1] 4.0 7.0 11.5 -1.0
> x/2
[1] 1.00 2.50 4.75 -1.50
>
> a <- 2
> a
[1] 2
> a*x
[1] 4 10 19 -6
>
```

**SOMMA TRA 2 VETTORI (stesso numero di elementi)**

```
> x <- c(2, 5, 9.5, -3)
> x
[1] 2.0 5.0 9.5 -3.0
> y <- c(2, 1, -2, 3)
> y
[1] 2 1 -2 3
>
> x+y
[1] 4.0 6.0 7.5 0.0
>
```

**ESTRAZIONE DI ELEMENTI DI UN VETTORE (specificando la posizione degli elementi estratti, tra parentesi quadre)**

```
> x <- c(2, 5, 9.5, -3)
> x
[1] 2.0 5.0 9.5 -3.0
>
> x[2]
[1] 5
>
> x[2]/x[1]
[1] 2.5
> x[1:3]
[1] 2.0 5.0 9.5
```

**FUNZIONI PER LA MANIPOLAZIONE DEI VETTORI**

```

> x <- c(2, 5, 9.5, -3)
> x
[1] 2.0 5.0 9.5 -3.0
> length(x)
[1] 4
> min(x)
[1] -3
> max(x)
[1] 9.5
> sum(x)
[1] 13.5
> mean(x)
[1] 3.375
>

```

**MATRICI**

```

> X <- matrix(c(
+ 2, 4,
+ 1, 2,
+ 6, 5,
+ 7, 1), ncol=2, byrow=T)
> X
      [,1] [,2]
[1,]  2   4
[2,]  1   2
[3,]  6   5
[4,]  7   1
> dim(X)
[1] 4 2
>
> X[3, ]
[1] 6 5
> X[,2]
[1] 4 2 5 1
> X[1,2]
[1] 4

```

**FATTORI (Vettori con elementi alfanumerici)**

```

> F1 <- factor( c('a', 'a', 'b', 'c', 'a') )
> F1
[1] a a b c a
Levels: a b c
> x <- c(2, 5, 9.5, -3)
> x
[1] 2.0 5.0 9.5 -3.0
> is.factor(x)

```

```
[1] FALSE
> is.factor(F1)
[1] TRUE
>
```

### DATA-FRAME (Risultati sia alfanumerici che qualitativi)

```
> cognome <- factor( c('Rossi', 'Bianchi', 'Verdi', 'Neri', 'Arancio') )
> cognome
[1] Rossi Bianchi Verdi Neri Arancio
Levels: Arancio Bianchi Neri Rossi Verdi
> voto <- c(29, 30, 24, 26, 18)
> voto
[1] 29 30 24 26 18
> genere <- factor( c('m','f','f','m','f') )
> genere
[1] m f f m f
Levels: f m
> dati <- data.frame(cognome, genere, voto)
> dati
  cognome genere voto
1 Rossi      m    29
2 Bianchi    f    30
3 Verdi      f    24
4 Neri       m    26
5 Arancio    f    18
> dati$voto
[1] 29 30 24 26 18
> mean(dati$voto)
[1] 25.4
```

### FUNZIONE class() (Tipo di classe di oggetti a cui appartiene l'oggetto)

```
> a <- 2
> a
[1] 2
> class(a)
[1] "numeric"
>
> x <- c(2, 5, 9.5, -3)
> x
[1] 2.0 5.0 9.5 -3.0
> class(x)
[1] "numeric"
>
> X <- matrix(c(
+ 2, 4,
+ 1, 2,
+ 6, 5,
+ 7, 1), ncol=2, byrow=T)
> X
```

```

      [,1] [,2]
[1,]  2   4
[2,]  1   2
[3,]  6   5
[4,]  7   1
> class(X)
[1] "matrix"
>
> F1 <- factor( c('a', 'a', 'b', 'c', 'a') )
> F1
[1] a a b c a
Levels: a b c
> class(F1)
[1] "factor"
>
> cognome <- factor( c('Rossi', 'Bianchi', 'Verdi', 'Neri', 'Arancio') )
> cognome
[1] Rossi Bianchi Verdi Neri Arancio
Levels: Arancio Bianchi Neri Rossi Verdi
> voto <- c(29, 30, 24, 26, 18)
> voto
[1] 29 30 24 26 18
> genere <- factor( c('m','f','f','m','f') )
> genere
[1] m f f m f
Levels: f m
> dati <- data.frame(cognome, genere, voto)
> dati
  cognome genere voto
1 Rossi      m    29
2 Bianchi    f    30
3 Verdi      f    24
4 Neri       m    26
5 Arancio    f    18
> class(dati)
[1] "data.frame"
>

```

## MEDIA

```

> y <- c(2, 3, 5, 1, 9)
> y
[1] 2 3 5 1 9
> mean(y)
[1] 4
> (2+3+5+1+9)/5
[1] 4
> sum(y)/5
[1] 4
> sum(y)/length(y)
[1] 4

```

**VARIANZA**

```

> y <- c(2, 3, 5, 1, 9)
> y
[1] 2 3 5 1 9
> var(y)
[1] 10
>
> mean(y)
[1] 4
> scarti <- y - mean(y)
> scarti
[1] -2 -1 1 -3 5
> scarti^2
[1] 4 1 1 9 25
> sum(scarti^2)
[1] 40
>
> sum(scarti^2) / (5-1)
[1] 10
>

```

**DEVIAZIONE STANDARD**

```

> y <- c(2, 3, 5, 1, 9)
> y
[1] 2 3 5 1 9
> mean(y)
[1] 4
> scarti <- y - mean(y)
> scarti
[1] -2 -1 1 -3 5
> scarti^2
[1] 4 1 1 9 25
> sum(scarti^2)
[1] 40
>
> sd(y)
[1] 3.162278
>
> sqrt( sum( (y-mean(y))^2)/(length(y)-1))
[1] 3.162278
>

```

**FUNZIONE summary() (Con un'unica istruzione dà 5 fondamentali statistiche descrittive)**

```

> y <- c(2, 3, 5, 1, 9)
> y
[1] 2 3 5 1 9
>
> summary(y)

```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	2	3	4	5	9

&gt;

**TEST t DI STUDENT**

&gt; y &lt;- c(2, 3, 5, 1, 9)

&gt; y

[1] 2 3 5 1 9

&gt; t.test(y, mu=3, var.equal = TRUE)

## One Sample t-test

data: y

t = 0.7071, df = 4, p-value = 0.5185

alternative hypothesis: true mean is not equal to 3

95 percent confidence interval:

0.07351368 7.92648632

sample estimates:

mean of x

4

**OPERATORE : (Genera sequenze di numeri interi)**

&gt; 1:4

[1] 1 2 3 4

&gt; y &lt;- c(2, 3, 5, 1, 9)

&gt; y

[1] 2 3 5 1 9

&gt; y[2]

[1] 3

&gt; y[2:4]

[1] 3 5 1

&gt;

**FUNZION seq() (Genera sequenze di numeri più complesse)**

&gt; seq(2, 8, by=2)

[1] 2 4 6 8

&gt; seq(0, 1, by=.1)

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

&gt; seq(1, 30, length.out=10)

[1] 1.000000 4.222222 7.444444 10.666667 13.888889 17.111111 20.333333 23.555556  
26.777778 30.000000

&gt;

**FUNZIONE rep()**

&gt; rep(c(1, 2, 3), each=3)

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

&gt; f1 &lt;- factor( rep(c('a', 'b', 'c'), each=3) )

```
> f1
[1] a a a b b b c c c
Levels: a b c
>
```

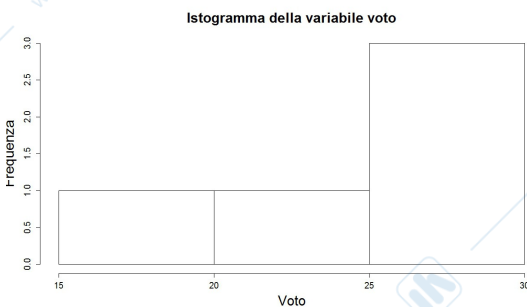
## IMPORTAZIONE DATI PACCHETTI

```
> library(car)
> data(Prestige)
> dim(Prestige)
[1] 102 6
> Prestige[1:5, ]
      education income women prestige census type
gov.administrators 13.11 12351 11.16 68.8 1113 prof
general.managers 12.26 25879 4.02 69.1 1130 prof
accountants 12.77 9271 15.70 63.4 1171 prof
purchasing.officers 11.42 8865 9.11 56.8 1175 prof
chemists 14.62 8403 11.68 73.5 2111 prof
> summary(Prestige)
      education      income      women      prestige      census      type
Min. :6.380 Min. : 611 Min. :0.000 Min. :14.80 Min. :1113 bc :44
1st Qu.: 8.445 1st Qu.: 4106 1st Qu.: 3.592 1st Qu.:35.23 1st Qu.:3120 prof:31
Median :10.540 Median : 5930 Median :13.600 Median :43.60 Median :5135 wc :
23
Mean :10.738 Mean : 6798 Mean :28.979 Mean :46.83 Mean :5402 NA's: 4
3rd Qu.:12.648 3rd Qu.: 8187 3rd Qu.:52.203 3rd Qu.:59.27 3rd Qu.:8312
Max. :15.970 Max. :25879 Max. :97.510 Max. :87.20 Max. :9517
>
```

## ISTOGRAMMA

```
> cognome <- factor( c('Rossi', 'Bianchi', 'Verdi', 'Neri', 'Arancio') )
> cognome
[1] Rossi Bianchi Verdi Neri Arancio
Levels: Arancio Bianchi Neri Rossi Verdi
> voto <- c(29, 30, 24, 26, 18)
> voto
[1] 29 30 24 26 18
> genere <- factor( c('m','f','f','m','f') )
> genere
[1] m f f m f
Levels: f m
> dati <- data.frame(cognome, genere, voto)
> dati
  cognome genere voto
1 Rossi      m    29
2 Bianchi    f    30
3 Verdi      f    24
4 Neri       m    26
5 Arancio    f    18
> dati$voto
```

```
[1] 29 30 24 26 18
> mean(dati$voto)
[1] 25.4
> hist(dati$voto)
> local({fn<-choose.files(filters=Filters[c('R','txt','All')],index=4)
+ file.show(fn,header=fn,title=")})
> hist(dati$voto,
+ main='Istogramma della variabile voto')
> hist(dati$voto,
+ main='Istogramma della variabile voto',
+ xlab='Voto', ylab='Frequenza')
> hist(dati$voto,
+ main='Istogramma della variabile voto',
+ xlab='Voto', ylab='Frequenza',
+ cex.main=2, cex.lab=2, cex.axis=1.25)
>
```



**FUNZIONE  $dnorm(x, \mu, \sigma)$  (Per calcolare la densità in  $x$ )**

```
> dnorm(0, 0, 1)
[1] 0.3989423
>
```

**FUNZIONE  $pnorm(q, \mu, \sigma)$  (Per calcolare il valore della ripartizione in  $q$ )**

```
> pnorm(-1.96, 0, 1)
[1] 0.0249979
> pnorm(1.96, 0, 1)
[1] 0.9750021
>
```

**FUNZIONE  $qnorm(p, \mu, \sigma)$  (Per calcolare il quantile di livello  $p$ )**

```
> qnorm(.975, 0, 1)
[1] 1.959964
> qnorm(.025, 0, 1)
[1] -1.959964
>
```

**FUNZIONE rnorm (n,mu,sigma) (Per generare un campione di dimensione n da una normale con media mu e deviazione standard sigma)**

```
> rnorm(10, 24, 3)
[1] 21.28975 20.42996 25.83862 19.66184 25.38103 23.33875 30.79766 27.06586
23.80436 24.59495
>
> rnorm(10, 24, 3)
[1] 25.31715 25.63835 21.55983 29.51071 19.16178 28.78043 23.71342 19.44066
21.11454 19.19383
>
```