

# R CheatSheet

---

## Statistica Descrittiva

### Introduzione

```
# caricare un file con gui
f <- file.choose()
x <- scan(f, sep = "separatore dati")

# oppure
x <- scan("path/to/file", sep = "separatore dati", dec = "separatore decimali")

# oppure per leggere csv
x <- read.csv("path/to/file", sep="separatore", header=TRUE)

# leggere sorgente R
source("path/to/file.R")

# collegare database al path di R
attach(x)

# scollegare database dal path di R
detach(x)

# installare pacchetto e usare una libreria
install.packages("name")
library("name")

# per stampare il risultato di R
print(x)

# per stampare una variabile, volendo con descrizione
cat("descrizione", x, "descrizione")

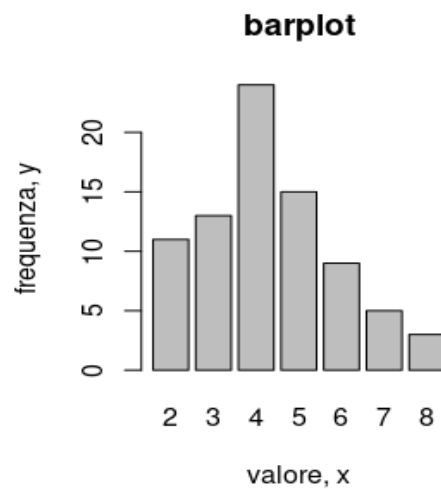
# frequenze assolute, sopra valori e sotto frequenze
# x può anche essere una sola colonna del database
table(x)

# frequenze relative
prop.table(table(x))

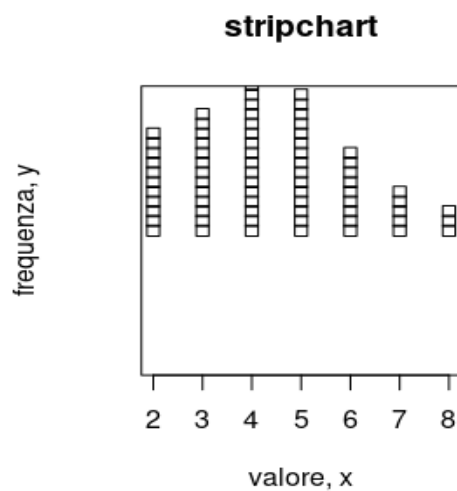
# frequenze cumulate assolute
cumsum(table(x))

# frequenze cumulate relative
cumsum(prop.table(table(x)))
```

```
# istogramma con frequenze assolute, su x i dati e su y le frequenze
barplot(table(x), xlab="etichetta x", ylab="etichetta ", main="titolo")
```



```
# grafico a stack con frequenze assolute, su x i dati e su y le frequenze
stripchart(x, method = "stack", xlab = "etichetta x", ylab = "etichetta ",
main="titolo")
```

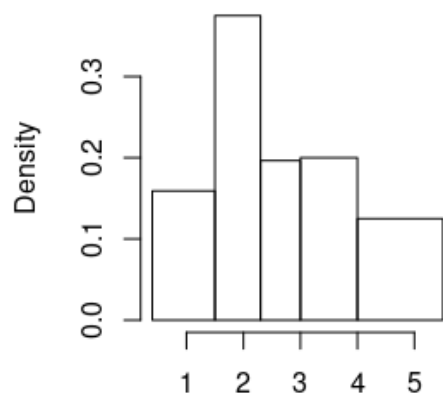


```
# minimo e massimo
min(x)
max(x)

# lista di dati
array <- c(0.4, 1.5, 2.3, 3, 4, 5.5)

# istogramma con step dato da un array
hist(x, breaks = array, xlab = "etichetta x", ylab = "etichetta ", main =
"titolo")
```

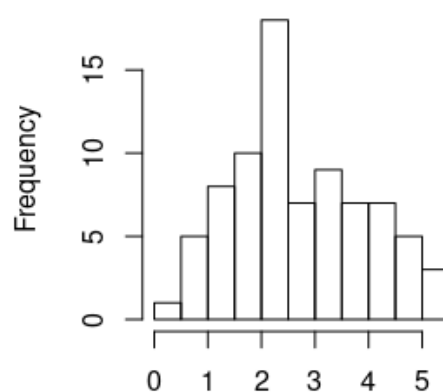
### istogramma con step



```
# istogramma con intervalli automatici
```

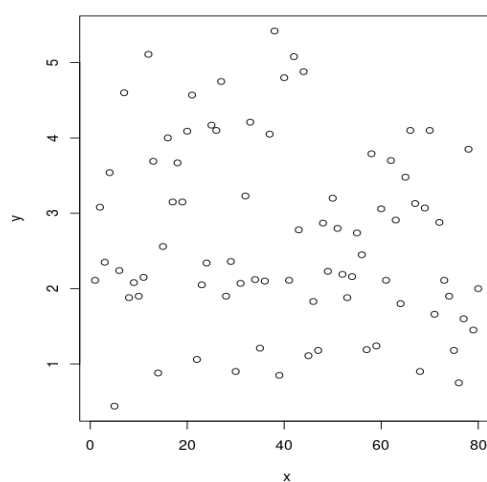
```
hist(x, xlab = "etichetta x", ylab = "etichetta ", main = "titolo")
```

### istogramma automatico



```
# scatterplot
```

```
plot(x, xlab = "x", ylab = "y", main = "titolo")
```



```
# ordinamento crescente
sort(x)

# ordinamento decrescente (decreasing = dec = true = T)
sort(x, dec = T)

# media
media <- mean(x)
print(media)
# mediana
median(x)

# moda
m <- table(x)
m[m == max(m)]

# varianza campionaria
var(x)

# deviazione standard
sd(x)
sqrt(var(x))

# quantili
quantile(x, c(0.25, 0.5, 0.75))

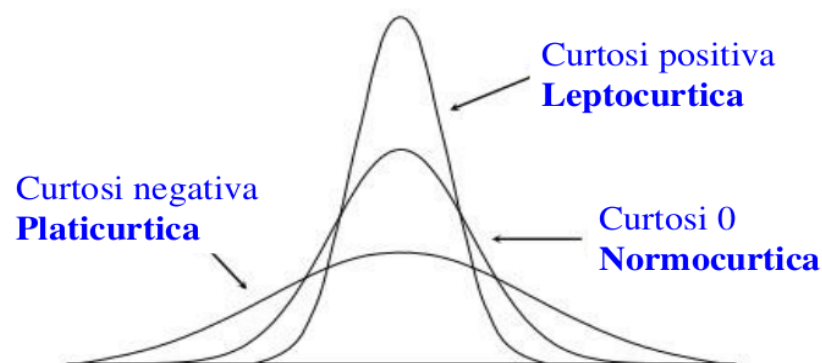
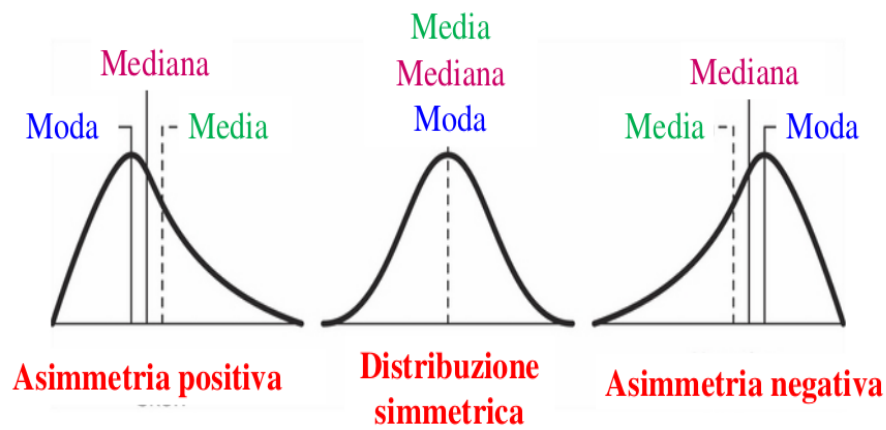
# range interquantile
IQR(x)

# range (min, max)
range(x)

# libreria indici di forma
install.packages("e1071")
library("e1071")

# asimmetria
skewness(x)
2 * sqrt(6 / length(x))

# curtosi
kurtosis(x)
4 * sqrt(6 / length(x))
```



## Caratteri Bidimensionali

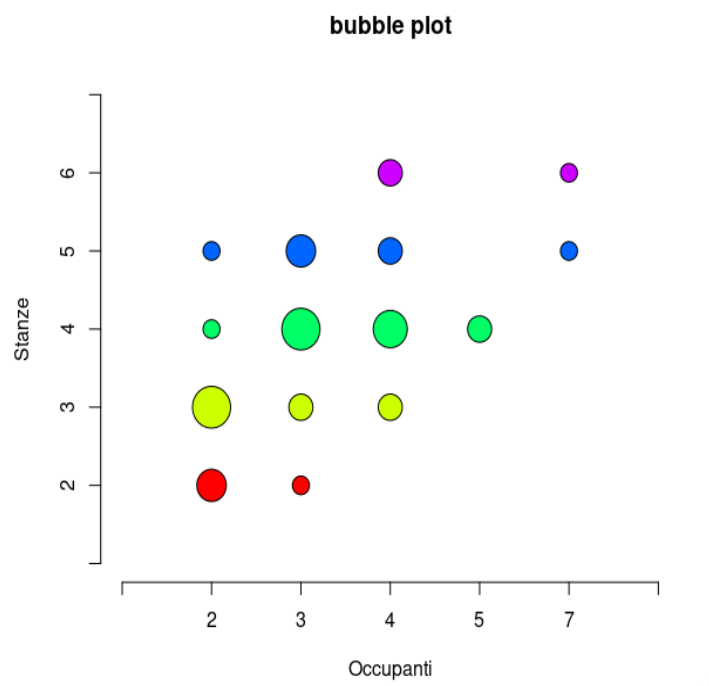
```
# frequenze assolute
tc <- table(x)

# tabella di contingenza con distribuzioni assolute marginali
tcc <- cbind(tc, margin.table(tc,1)) # marginale stanze
rbind(tcc, margin.table(tcc,2)) # marginale occupanti

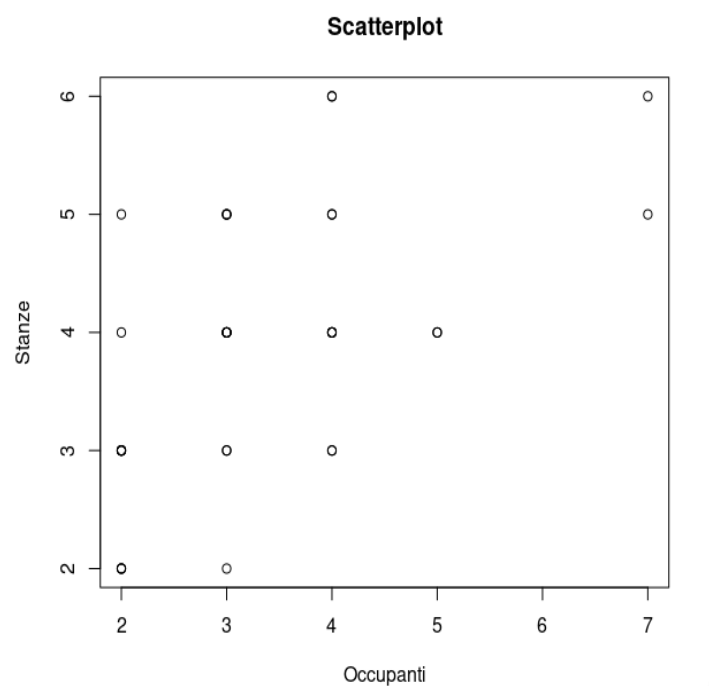
# frequenze relative
tcr <- prop.table(table(x))

# tabella di contingenza con distribuzioni relative marginali
tccr <- cbind(tcr,margin.table(tcr,1)) # marginale stanze
rbind(tccr,margin.table(tccr,2)) # marginale occupanti

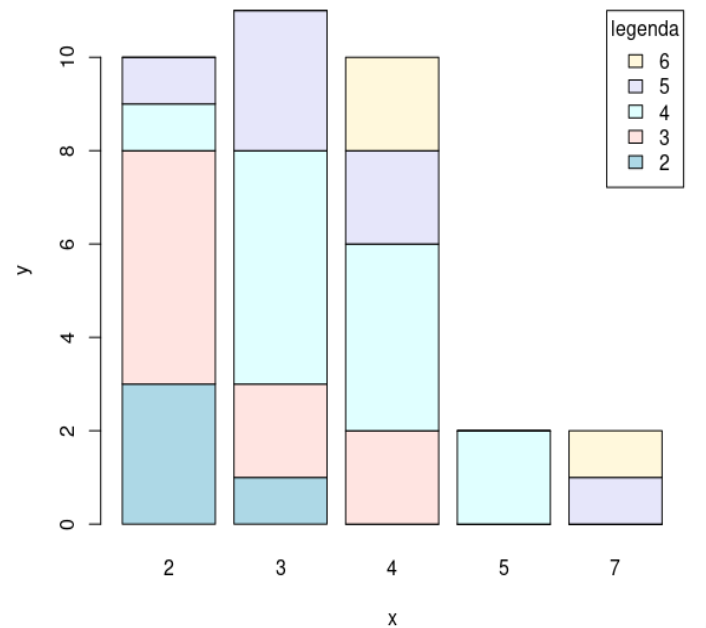
# bubbleplot (install.packages("labstatR")) prende in automatico le
etichette
require("labstatR")
bubbleplot(tc)
```



```
# scatterplot
plot(x$valorix, x$valoriY, xlab="x", ylab="y", main="Titolo")
```



```
# barplot con legenda (dotata di posizione e titolo), colori col dati da un
array di colori
barplot(table(x), legend=TRUE, col = c("lightblue", "mistyrose",
"lightcyan", "lavender", "cornsilk"), xlab="x", ylab="y", args.legend=list(x="to
pright", title="legenda"))
```



```
# covarianza
cov(x$valore1,x$valore2)

# correlazione
cor(x$valore1,x$valore2)
```

## Regressione Lineare

```
# scarto quadratico medio x
sqrt(mean((x$valore1.x. - mean(x$valore1.x.))^2))

# scarto quadratico medio y
sqrt(mean((x$valore2.y. - mean(x$valore2.y.))^2))

# covarianza
mean((x$valore1.x. - mean(x$valore1.x.)) *
      (x$valore2.y. - mean(x$valore2.y.)))

# correlazione
cov(x$valore1.x.,x$valore2.y.) /
(sd(x$valore1.x.) * sd(x$valore2.y.))

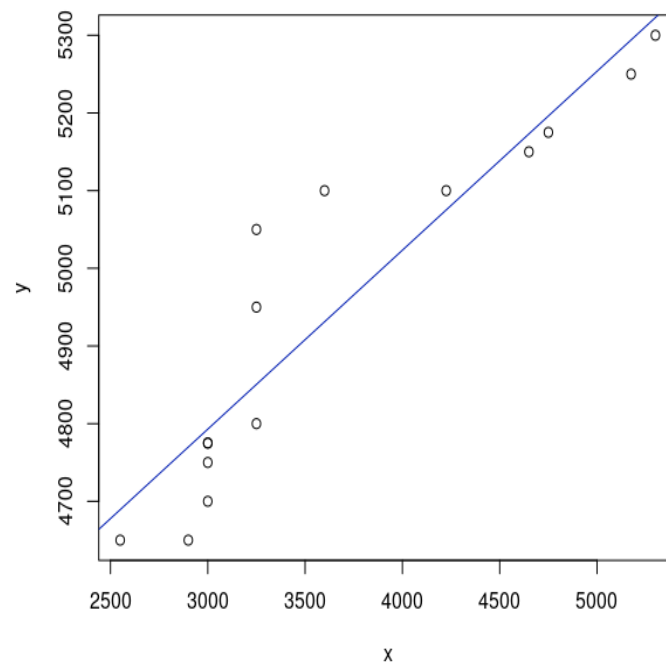
# retta di regressione (richiamare rr farà stampare i
# dati della retta), rr è una classe coi dati della regressione
rr <- lm(x$valore2.y.~ x$valore1.x.)

# per accedere direttamente ai coefficienti
coefficients(rr)

# grafico della regressione
```

```
# prima lo scatterplot (con x$valore1.x. acedo ai dati "valore"
# del dataset x, ma possono essere semplici vettori)
plot(x$valore1.x., x$valore2.y., xlab="x", ylab="y", main="titolo")

# poi la retta (lwd larghezza linea,
# col = colore, esplicito o in hex)
abline(rr, col="colore", lwd=2)
```



## Regressione non Lineare

```
# retta di regressione (richiamare rrnl farà stampare i
# dati della retta)
rrnl <- lm(x$valore2.y. ~ 1 + x$valore1.x. + I(x$valore1.x.^2))

# si può usare in alternativa
lm(x$valore2.y. ~ poly(x$valore1.x., 2, raw = TRUE))

# grafico della regressione non lineare
plot(x$valore1.x., x$valore2.y., xlab="x", ylab="y")
lines(x$valore1.x., predict(rrnl), col="green", lwd=2)
```



