

LAB – LEZ. 1 – STATISTICA DESCRITTIVA CON R

LABORATORIO DI PROBABILITA' E STATISTICA

Docente: Bruno Gobbi

1 - STATISTICA DESCRITTIVA CON "R"

IL SOFTWARE "R"

- ▶ SOFTWARE PER ANALISI STATISTICHE
- ▶ LIBERO (LICENZA GNU GPL)
- ▶ LINGUAGGIO DI PROGRAMMAZIONE SEMPLICE E EFFICIENTE
- ▶ INTERFACCIA A RIGA DI COMANDO ...
- ▶ ... MA POSSIBILITA' DI SCARICARE MOLTI PACCHETTI AGGIUNTIVI DAL SITO:

www.r-project.org

RStudio

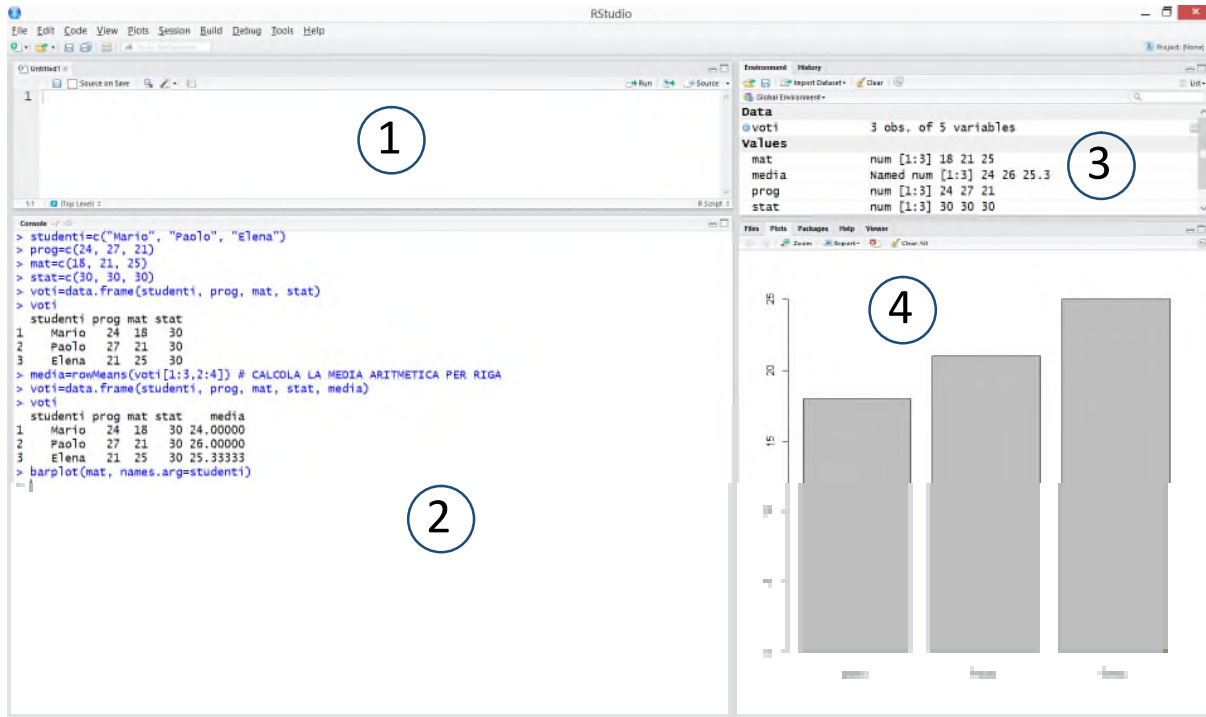
- ▶ IDE (INTEGRATED DEVELOPMENT ENVIRONMENT) PER R
- ▶ GRATUITO PER FINI NON COMMERCIALI

www.rstudio.com

CLICCARE SU:

APPLICAZIONI → PROGRAMMAZIONE → RStudio

L'AMBIENTE DI SVILUPPO DI RStudio



1 – FINESTRA PER CREARE E SALVARE FILE .R

2 – CONSOLE DEI COMANDI

3 – VARIABILI PRESENTI NELLA MEMORIA DEL PROGRAMMA E CRONOLOGIA DEI COMANDI

4 – ESPLORA RISORSE, GRAFICI, SCARICAMENTO PACCHETTI AGGIUNTIVI, HELP, VIEWER

Il codice presentato nelle pagine seguenti va scritto nella finestra 2, quella della console.

ALCUNE OPERAZIONI DI BASE CON R

NOTE SULLE MODALITA' DI PRESENTAZIONE DEL CODICE:

- NEL PRESENTE DOCUMENTO I COMANDI DA INSERIRE NELLA CONSOLE DI R SARANNO PRECEDUTI DAL SIMBOLO ">"
- I **COMMENTI** SUI VARI PASSAGGI SARANNO INVECE PRECEDUTI DA "#" E PER COMODITA' **COLORATI IN VERDE**

IN R E' POSSIBILE ESEGUIRE SUBITO OPERAZIONI MATEMATICHE DIGITANDO DIRETTAMENTE I VALORI

```
> 1+2
```

```
[1] 3
```

```
> 5*3
```

```
[1] 15
```

```
> 12/4
```

```
[1] 3
```

```
> 5^2
```

```
[1] 25
```

```
> sqrt(4) # SQUARED ROOT
```

```
[1] 2
```

```
> abs(-5) # VALORE ASSOLUTO
```

```
[1] 5
```

```
> log(1)
```

```
[1] 0
```

```
> log(5)
```

```
[1] 1.609438
```

```
> 3 < 5 # DISUGUAGLIANZE, RESTITUISCE TRUE SE VERO, FALSE ALTRIMENTI
```

```
[1] TRUE
```

```
> 3 > 5
```

```
[1] FALSE
```

PER PORRE IL SEGNO DI UGUAGLIANZA NEL CONFRONTO FRA DUE VALORI SI USA ==

```
> 3==3
```

```
[1] TRUE
```

PER IL CASO DIVERSO

```
> 3!=4
```

```
[1] TRUE
```

PER CONSULTARE LA GUIDA IN LINEA SU UNA DETERMINATA FUNZIONE, SI ANTEPONE “?” AL NOME

```
> ? plot
```

COME IN QUASI TUTTI I LINGUAGGI DI PROGRAMMAZIONE, ANCHE IN R SI UTILIZZANO GLI OGGETTI. QUESTI POSSONO ESSERE SINGOLI NUMERI, VETTORI O ALTRO. PER CREARE UN OGGETTO È SUFFICIENTE INDICARE IL NOME E A COSA È =

```
> pippo=2
```

```
> pippo
```

```
[1] 2
```

UN ALTRO MODO PER ASSEGNARE UN VALORE AD UN OGGETTO E' IL SEGUENTE

```
> pippo<-2
```

```
> pippo
```

```
[1] 2
```

PER CREARE UN VETTORE DI VALORI, SI USA LA FUNZIONE “c” (CONCATENATE) SEGUITA DAGLI ELEMENTI INCLUSI FRA PARENTESI

```
> pippo=c(1, 3, 8)
```

```
> pippo
```

```
[1] 1 3 8
```

```
> pippo=2*2
```

```
> pippo
```

```
[1] 4
```

NOTA BENE: R E' CaSe SeNsItIvE

“pippo” è diverso da “Pippo” che è diverso da “PIPPO”

CREAZIONE DI UNA TABELLA CON I VOTI DEGLI STUDENTI

CREIAMO UN ELENCO DI STUDENTI; NEL CASO DI STRINGHE DI TESTO RICORDARSI DI USARE LE ""

```
> studenti=c("A", "B", "C")
```

SE VOGLIAMO DARE UN ALTRO NOME AGLI STUDENTI, POSSIAMO RICREARE L'OGGETTO "STUDENTI" RISCRIVENDOLO:

```
> studenti=c("Mario", "Paolo", "Elena")
```

CREIAMO UN VETTORE DEI VOTI IN PROGRAMMAZIONE

```
> prog=c(24, 27, 21)
```

```
> prog
```

```
[1] 24 27 21
```

ORA CREIAMO UN'UNICA TABELLA CHE RIPORTI IL VOTO DI OGNI STUDENTE

```
> voti=data.frame(studenti, prog)
```

```
> voti
```

```
  studenti prog
```

```
1  Mario  24
```

```
2  Paolo  27
```

```
3  Elena  21
```

AGGIUNGIAMO I VOTI DI MATEMATICA

```
mat=c(18, 21, 25)
```

```
> mat
```

```
[1] 18 21 25
```

```
> voti=data.frame(studenti, prog, mat)
```

```
> voti
```

```
  studenti prog mat
```

```
1  Mario  24 18
```

```
2  Paolo  27 21
```

```
3  Elena  21 25
```

AGGIUNGIAMO I VOTI DI STATISTICA

```
> stat=c(30, 30, 30)
```

```
[1] 30 30 30
```

```
voti=data.frame(studenti, prog, mat, stat)
```

```
> voti
```

```
  studenti prog mat stat
```

```
1  Mario  24 18 30
```

```
2  Paolo  27 21 30
```

```
3  Elena  21 25 30
```

#CALCOLIAMO LA MEDIA DEI VOTI PER STUDENTE; PER FARE CIO', NEL CASO DI DATI PRESENTATI SOTTO FORMA DI MATRICE/TABELLA, DOBBIAMO USARE LA FUNZIONE rowMeans E INDICARE LE RIGHE E LE COLONNE DELLA TABELLA SULLE QUALI INTENDIAMO FARE IL CALCOLO; NEL NOSTRO ESEMPIO VOGLIAMO CALCOLARE LA MEDIA DELLE RIGHE DA 1 A 3 DELLA TABELLA "voti" E SUI VALORI PRESENTI NELLE COLONNE DA 2 A 4 (PERCHE' LA PRIMA CONTIENE I NOMI DEGLI STUDENTI)

```
> media=rowMeans(voti[1:3,2:4]) # CALCOLA LA MEDIA ARITMETICA PER RIGA
```

```
> media
```

```
[1] 24.00000 26.00000 25.33333
```

```
> voti=data.frame(studenti, prog, mat, stat, media)
```

```
> voti
```

```
  studenti prog mat stat  media
```

```
1  Mario  24 18 30 24.00000
```

```
2  Paolo  27 21 30 26.00000
```

```
3  Elena  21 25 30 25.33333
```

ARROTONDIAMO A DUE DECIMALI LA MEDIA TRAMITE LA FUNZIONE round

```
> round(media, 2) # round(DATI, N. CIFRE DECIMALI DA TENERE)
```

```
[1] 24.00 26.00 25.33
```

```
> media=round(media, 2)
```

```
> media
```

```
[1] 24.00 26.00 25.33
```

AGGIUNGIAMO LA COLONNA DELLA MEDIA ARROTONDATA ALLA NOSTRA TABELLA

```
> voti=data.frame(studenti, prog, mat, stat, media)
```

```
> voti
```

```
  studenti prog mat stat  media
```

```
1  Mario  24 18 30 24.00
```

```
2  Paolo  27 21 30 26.00
```

3 Elena 21 25 30 25.33

PER FARE IL GRAFICO DEI VOTI IN MATEMATICA DEI 3 STUDENTI, SI USA BARPLOT, INDICANDO COME PRIMO ARGOMENTO LA VARIABILE DI CUI SI DESIDERA FARE IL GRAFICO. LE ETICHETTE (I NOMI) SI INDICANO CON L'OPZIONE "NAMES.ARG"

```
> barplot(mat, names.arg=studenti)
```

OPPURE

```
> barplot(voti$mat, names.arg=voti$studenti)
```

OPPURE

```
> barplot(voti[,3], names.arg=studenti)
```

APPROFONDIMENTO EXTRA (NON VERRA' CHIESTO AGLI ESAMI!)

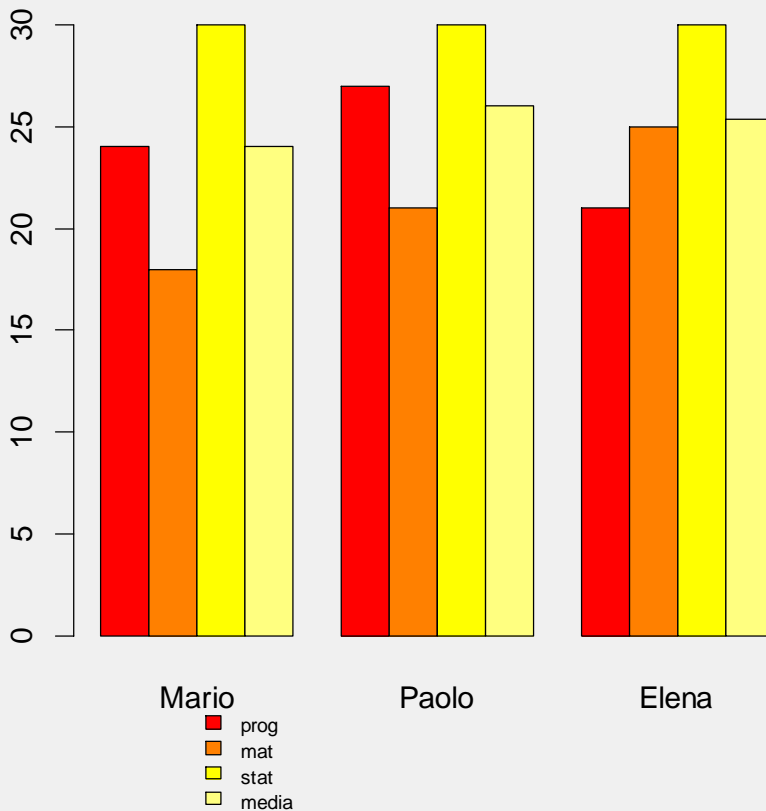
PER CREARE UN UNICO GRAFICO CON TUTTI I VOTI DEGLI STUDENTI

```
> barplot(t(as.matrix(voti[,2:5])), beside = TRUE, names.arg=studenti, col=heat.colors(4))
```

```
> par(xpd=TRUE) # SERVE PER FAR DISEGNARE LA LEGENDA FUORI DALL'AREA DEL GRAFICO
```

```
> legend(2.8, -2, c("prog", "mat", "stat", "media"), cex = 0.7, bty = "n", fill=heat.colors(4))
```

POSIZIONE ORIZZ, VERT, ETICHETTE, DIM FONT, PRESENZA BORDO, RIEMPIMENTO)



RICORDARSI ALLA FINE DI TOGLIERE L'OPZIONE XPD!

```
> par(xpd=FALSE)
```


ESERCIZIO SULLA DISTRIBUZIONE DEGLI SMARTPHONE

```
> OScell=c("Android", "Iphone", "Windows", "Altro")
```

```
# RICORDIAMO CHE R È CASE SENSITIVE
```

```
> oscell
```

```
Error: object 'oscell' not found
```

```
> OScell
```

```
[1] "Android" "Iphone" "Windows" "Altro"
```

```
# CREIAMO IL VETTORE DEL NUMERO DI CELLULARI SCEGLIENDO IL NOME "numcell"
```

```
> numcell=c(50, 40, 10, 2)
```

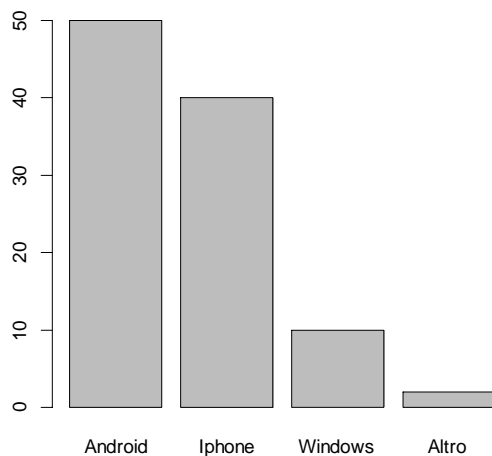
```
> cell=data.frame(OScell, numcell)
```

```
> cell
```

```
  OScell numcell
1 Android    50
2 Iphone   40
3 Windows   10
4 Altro     2
```

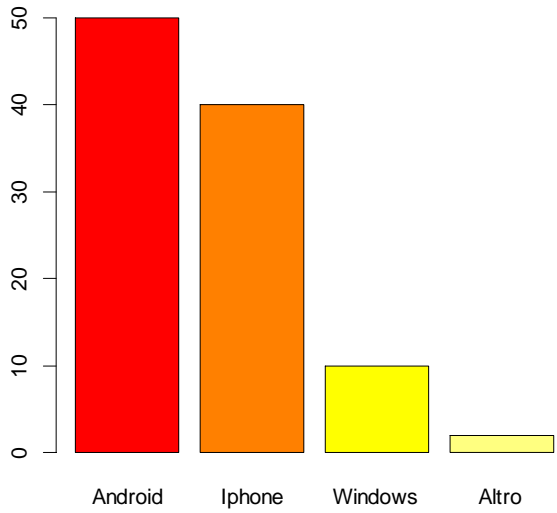
```
# CREIAMO L'ISTOGRAMMA DELLA DISTRIBUZIONE DEI CELLULARI
```

```
> barplot(cell$numcell, names.arg=OScell)
```



PER AGGIUNGERE UN PO' DI COLORI, USIAMO LA COMBINAZIONE PRESETTATA DEGLI heat.colors

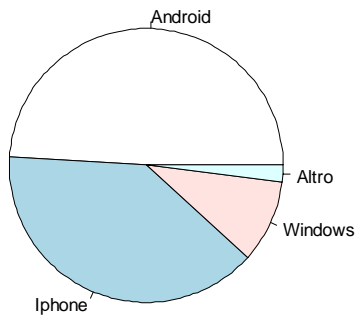
```
> barplot(cell$numcell, names.arg=OScell, col=heat.colors(4))
```



PER UN TIPO DI RAPPRESENTAZIONE COME QUESTA, È PIÙ COMODO USARE UN GRAFICO A TORTA

FARE GRAFICO A TORTA DEI CELLULARI

```
> pie(numcell, labels=OScell)
```



```
# APPROFONDIMENTO EXTRA (NON VERRA' CHIESTO AGLI ESAMI)
```

```
# GRAFICO A TORTA CON PERCENTUALI
```

```
lbls <- OScell
```

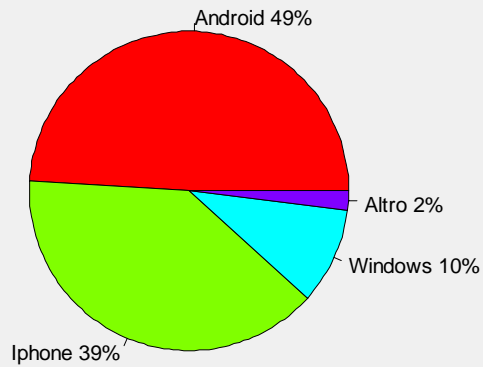
```
pct <- round(numcell /sum(numcell)*100)
```

```
lbls <- paste(lbls, pct) # AGGIUNGE LE PERCENTUALI ALLE LABEL
```

```
lbls <- paste(lbls,"%",sep="") # AGGIUNGE IL SIMBOLO DI "%" ALLE LABEL
```

```
pie(numcell,labels = lbls, col=rainbow(length(lbls)), main="Grafico a torta")
```

Grafico a torta



ESERCIZIO SULLA SERIE STORICA DEL “NILO”

PER AVERE UNA LISTA DI TUTTI I DATASET PRE CARICATI IN R

```
> data()
```

SCEGLIAMO IL DB DEL LIVELLO DEL FIUME NILO DAL 1871 AL 1970

```
> Nile
```

Time Series:

Start = 1871

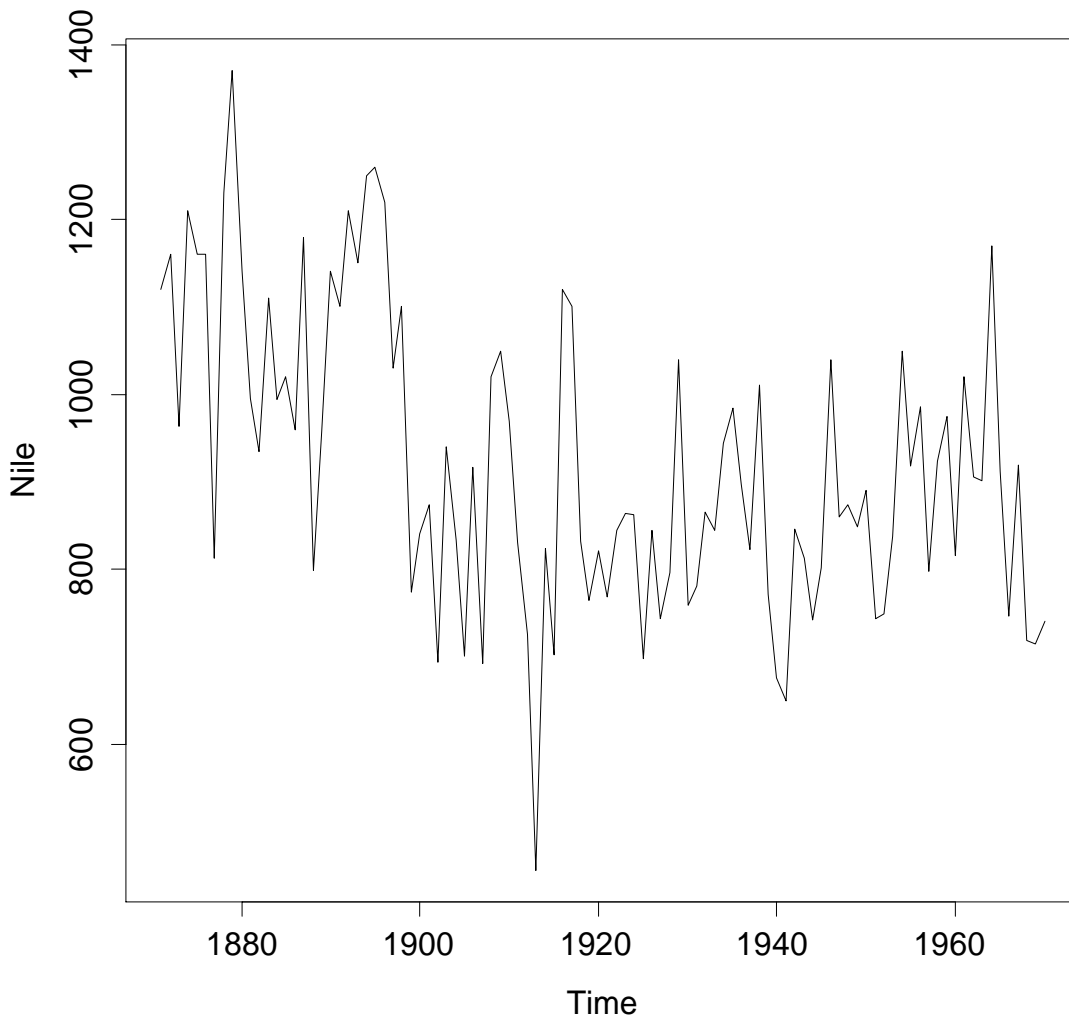
End = 1970

Frequency = 1

```
[1] 1120 1160 963 1210 1160 1160 813 1230 1370 1140 995 935 1110 994 1020  
[16] 960 1180 799 958 1140 1100 1210 1150 1250 1260 1220 1030 1100 774 840  
[31] 874 694 940 833 701 916 692 1020 1050 969 831 726 456 824 702  
[46] 1120 1100 832 764 821 768 845 864 862 698 845 744 796 1040 759  
[61] 781 865 845 944 984 897 822 1010 771 676 649 846 812 742 801  
[76] 1040 860 874 848 890 744 749 838 1050 918 986 797 923 975 815  
[91] 1020 906 901 1170 912 746 919 718 714 740
```

PER FARE IL GRAFICO DELLA SERIE STORICA NILO E' SUFFICIENTE USARE LA FUNZIONE plot

```
> plot(Nile)
```



PER CALCOLARE LA MEDIA

```
> mean(Nile)
```

```
[1] 919.35
```

PER CALCOLARE LA MEDIANA

```
> median(Nile)
```

```
[1] 893.5
```

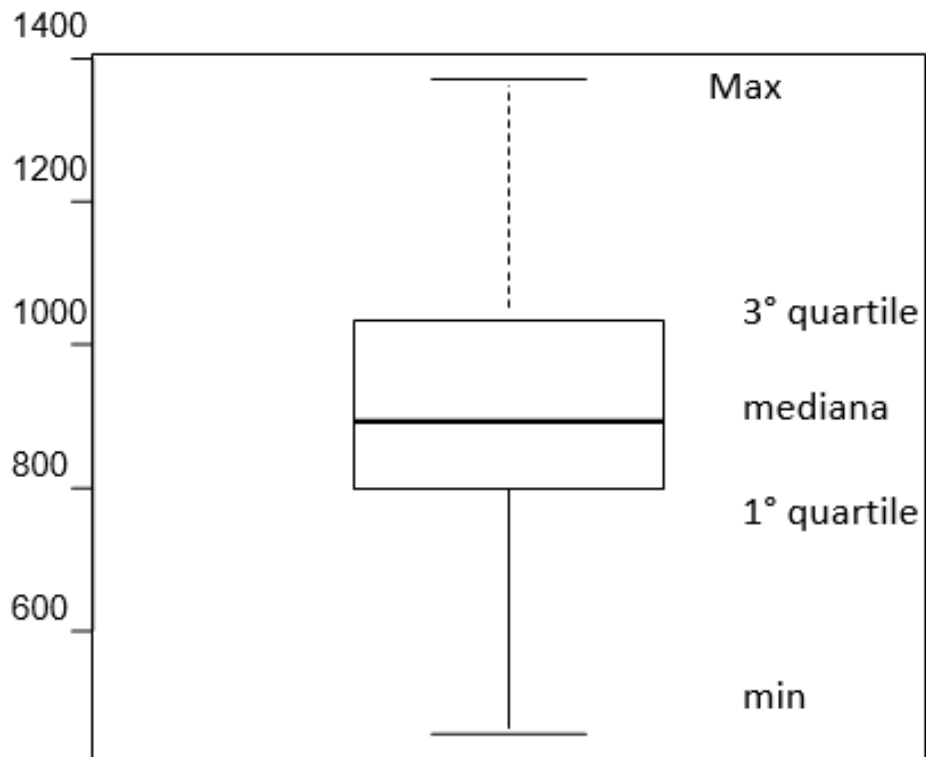
PER CALCOLARE I QUANTILI (IN QUESTO CASO I QUARTILI E IL MINIMO E IL MASSIMO)

```
> quantile(Nile, c(0, 0.25, 0.50, 0.75, 1))
```

```
 0%  25%  50%  75% 100%  
456.0 798.5 893.5 1032.5 1370.0
```

PER CREARE UN GRAFICO DI TIPO "BOXPLOT"

```
> boxplot(Nile)
```



INDICI DI VARIABILITA'

PER TROVARE LO SCARTO QUADRATICO MEDIO CAMPIONARIO

```
> sd(Nile)
```

```
[1] 169.2275
```

PER CALCOLARE LA VARIANZA CAMPIONARIA

```
> var(Nile)
```

```
[1] 28637.95
```

ALCUNE UTILI FUNZIONI

- ▶ **sum()** calcola la somma degli elementi di un vettore di dati;
- ▶ **length()** restituisce la numerosità di un vettore;
- ▶ **range()** per trovare il minimo e il massimo di un vettore;
- ▶ **mean()** calcola la media;
- ▶ **weighted.mean(x, pesi)** calcola la media ponderata;
- ▶ **median()** calcola la mediana;
- ▶ **sd()** calcola lo scarto quadratico medio campionario di un vettore di dati;
- ▶ **var()** calcola la varianza campionaria di un vettore di dati o la covarianza tra due vettori;
- ▶ **cor()** calcola la correlazione tra due vettori;
- ▶ **summary()** riporta le principali statistiche descrittive di un vettore o di una matrice di dati.

ALCUNI PACCHETTI UTILI

- ▶ **ggplot2**: pacchetto per migliorare e facilitare la creazione di grafici
- ▶ **plyr**: per analizzare i dati raggruppandoli in sotto insiemi o combinandoli fra di loro (analisi group-by)
- ▶ **rcpp**: per scrivere funzioni di R che richiamano codice C++
- ▶ **XML**: per creare documenti XML
- ▶ **zoo**: per l'analisi delle serie storiche
- ▶ **quantmod**: fornisce degli strumenti per il download di dati finanziari, grafici e la loro analisi
- ▶ **shiny**: per trasformare le analisi di R in applicazioni interattive per il web

<https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages>

<https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages>

LINK UTILI

- ▶ <http://www.statmethods.net/>
- ▶ <http://www.r-bloggers.com/>
- ▶ <http://www.rdocumentation.org/>
- ▶ <http://rseek.org/>
- ▶ <http://www.inside-r.org/>
- ▶ <http://www.ats.ucla.edu/stat/r/>