Statistica Descrittiva Univariata

Massimo Aria

March 20, 2018

DISTRIBUZIONI DI FREQUENZA

```
# importare il dataframe automobile (fonte UCI Machine Learning repository)

df=read.table("automobile.csv",header=TRUE,sep=";",dec=",")
str(df)
```

```
## 'data.frame':    205 obs. of  26 variables:
##  $ symbolig      : int  3 3 1 2 2 2 1 1 1 0 ...
##  $ normlosses    : int  NA NA NA 164 164 NA 158 NA 158 NA ...
##  $ make          : Factor w/ 22 levels "alfa-romero",..: 1 1 1 2 2 2 2 2 2 2 ...
##  $ fuel          : Factor w/ 2 levels "diesel","gas": 2 2 2 2 2 2 2 2 2 2 ...
##  $ aspiration    : Factor w/ 2 levels "std","turbo": 1 1 1 1 1 1 1 1 2 2 ...
##  $ doors         : Factor w/ 3 levels "","four","two": 3 3 3 2 2 3 2 2 2 3 ...
##  $ body          : Factor w/ 5 levels "convertible",..: 1 1 3 4 4 4 4 5 4 3 ...
##  $ drivewheels   : Factor w/ 3 levels "4wd","fwd","rwd": 3 3 3 2 1 2 2 2 2 1 ...
##  $ enginelocation: Factor w/ 2 levels "front","rear": 1 1 1 1 1 1 1 1 1 1 ...
##  $ wheelbase     : num  88.6 88.6 94.5 99.8 99.4 ...
##  $ length        : num  169 169 171 177 177 ...
##  $ width         : num  64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 67.9 ...
##  $ height        : num  48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 ...
##  $ curbweight    : int  2548 2548 2823 2337 2824 2507 2844 2954 3086 3053 ...
##  $ engine        : Factor w/ 7 levels "dohc","dohcv",..: 1 1 6 4 4 4 4 4 4 4 ...
##  $ cylinders     : Factor w/ 7 levels "eight","five",..: 3 3 4 3 2 2 2 2 2 2 ...
##  $ enginesize    : int  130 130 152 109 136 136 136 136 131 131 ...
##  $ fuelsystem    : Factor w/ 8 levels "1bbl","2bbl",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ bore          : num  3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.13 ...
##  $ stroke        : num  2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 3.4 ...
##  $ compression   : num  9 9 9 10 8 8.5 8.5 8.5 8.3 7 ...
##  $ hp            : int  111 111 154 102 115 110 110 110 140 160 ...
##  $ rpm           : int  5000 5000 5000 5500 5500 5500 5500 5500 5500 5500 ...
##  $ cityMPG       : int  21 21 19 24 18 19 19 19 17 16 ...
##  $ highwayMPG    : int  27 27 26 30 22 25 25 25 20 22 ...
##  $ price         : int  13495 16500 16500 13950 17450 15250 17710 18920 23875 NA ...
```

```
# distribuzione di frequenza
table(df$make)                  # frequenze assolute
```

```
##
##   alfa-romero          audi           bmw     chevrolet         dodge
##             3             7             8             3             9
##         honda         isuzu        jaguar         mazda mercedes-benz
##            13             4             3            17             8
##       mercury    mitsubishi        nissan        peugot      plymouth
##             1            13            18            11             7
##       porsche       renault          saab        subaru        toyota
##             5             2             6            12            32
##    volkswagen         volvo
##            12            11
```

```r
table(df$make)/dim(df)[1]       # frequenze relative
```

```
##
##   alfa-romero          audi           bmw     chevrolet         dodge
##   0.014634146   0.034146341   0.039024390   0.014634146   0.043902439
##         honda         isuzu        jaguar         mazda mercedes-benz
##   0.063414634   0.019512195   0.014634146   0.082926829   0.039024390
##       mercury    mitsubishi        nissan        peugot      plymouth
##   0.004878049   0.063414634   0.087804878   0.053658537   0.034146341
##       porsche       renault          saab        subaru        toyota
##   0.024390244   0.009756098   0.029268293   0.058536585   0.156097561
##    volkswagen         volvo
##   0.058536585   0.053658537
```

```r
table(df$make)/dim(df)[1]*100   # frequenze percentuali
```

```
##
##   alfa-romero          audi           bmw     chevrolet         dodge
##     1.4634146     3.4146341     3.9024390     1.4634146     4.3902439
##         honda         isuzu        jaguar         mazda mercedes-benz
##     6.3414634     1.9512195     1.4634146     8.2926829     3.9024390
##       mercury    mitsubishi        nissan        peugot      plymouth
##     0.4878049     6.3414634     8.7804878     5.3658537     3.4146341
##       porsche       renault          saab        subaru        toyota
##     2.4390244     0.9756098     2.9268293     5.8536585    15.6097561
##    volkswagen         volvo
##     5.8536585     5.3658537
```

```r
# distribuzioni di frequenza cumulate
df$doors=ordered(df$doors,levels=c("two","four")) # si ordinano i livelli
table(df$doors)
```

```
##
##  two four
##   89  114
```

```r
cumsum(table(df$doors))
```

```
##  two four
##   89  203
```

```r
cumsum(table(df$doors))/dim(df)[1]
```

```
##       two      four
## 0.4341463 0.9902439
```

```r
# dividere in classi una variabile numerica
min(df$price,na.rm=TRUE)
```

```
## [1] 5118
```

```r
max(df$price,na.rm=TRUE)
```

```
## [1] 45400
```

```r
cut(df$price/1000,c(5,10,20,30,40,50))
```

```
##   [1] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20]
##   [9] (20,30] <NA>    (10,20] (10,20] (20,30] (20,30] (20,30] (30,40]
```

```
##  [17] (40,50] (30,40] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]
##  [25] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (10,20] (5,10]  (5,10]
##  [33] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]
##  [41] (10,20] (10,20] (10,20] (5,10]  <NA>    <NA>    (10,20] (30,40]
##  [49] (30,40] (30,40] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (10,20]
##  [57] (10,20] (10,20] (10,20] (5,10]  (5,10]  (10,20] (10,20] (10,20]
##  [65] (10,20] (10,20] (10,20] (20,30] (20,30] (20,30] (30,40] (30,40]
##  [73] (30,40] (40,50] (40,50] (10,20] (5,10]  (5,10]  (5,10]  (5,10]
##  [81] (5,10]  (5,10]  (10,20] (10,20] (10,20] (5,10]  (5,10]  (5,10]
##  [89] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]
##  [97] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (10,20] (10,20] (10,20]
## [105] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20]
## [113] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (5,10]  (5,10]
## [121] (5,10]  (5,10]  (5,10]  (5,10]  (10,20] (20,30] (30,40] (30,40]
## [129] (30,40] <NA>    (5,10]  (5,10]  (10,20] (10,20] (10,20] (10,20]
## [137] (10,20] (10,20] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]
## [145] (5,10]  (10,20] (5,10]  (10,20] (5,10]  (10,20] (5,10]  (5,10]
## [153] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]
## [161] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (5,10]
## [169] (5,10]  (5,10]  (10,20] (10,20] (10,20] (5,10]  (10,20] (5,10]
## [177] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (5,10]  (5,10]
## [185] (5,10]  (5,10]  (5,10]  (5,10]  (5,10]  (10,20] (5,10]  (10,20]
## [193] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20] (10,20]
## [201] (10,20] (10,20] (20,30] (20,30] (20,30]
## Levels: (5,10] (10,20] (20,30] (30,40] (40,50]
```

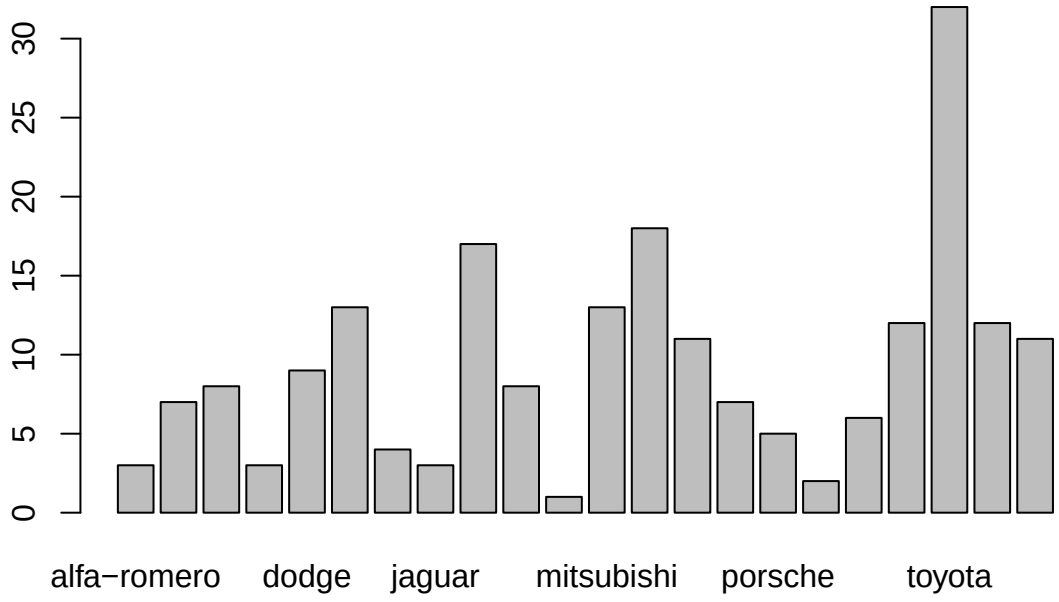
```
table(cut(df$price/1000,c(5,10,20,30,40,50)))
```

```
##
##  (5,10] (10,20] (20,30] (30,40] (40,50]
##      98      78      11      11       3
```

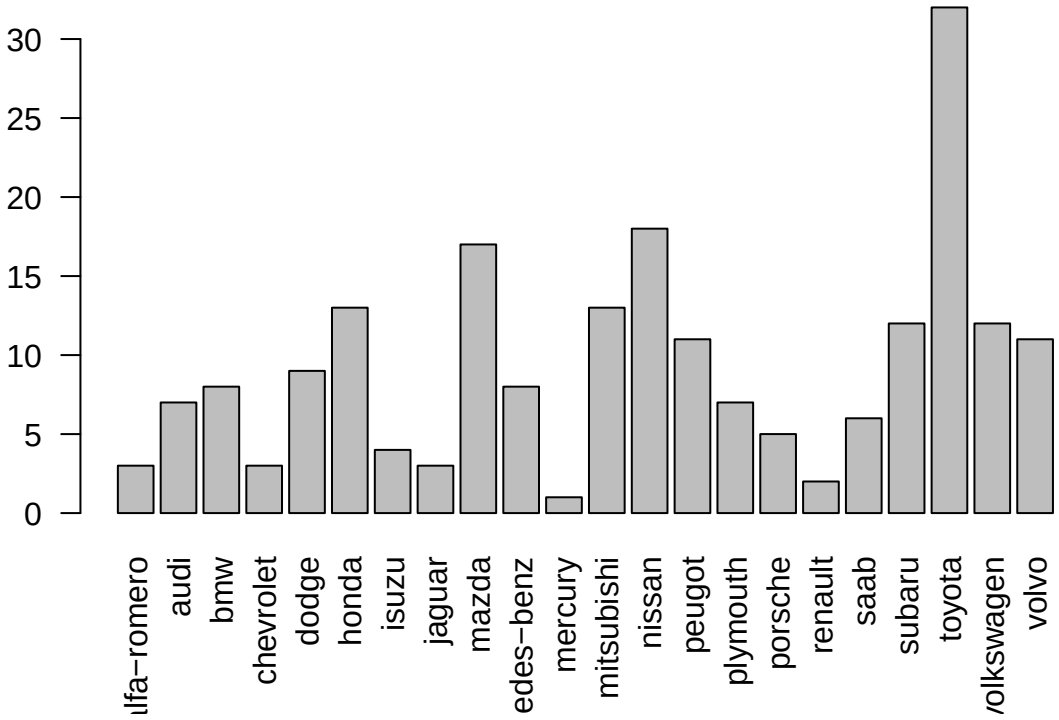
```
# analizziamo i risultati tabulari della funzione hist
h=hist(df$price/1000,c(5,10,20,30,40,50),plot=FALSE)
h
```

```
## $breaks
## [1]  5 10 20 30 40 50
##
## $counts
## [1] 98 78 11 11  3
##
## $density
## [1] 0.097512438 0.038805970 0.005472637 0.005472637 0.001492537
##
## $mids
## [1]  7.5 15.0 25.0 35.0 45.0
##
## $xname
## [1] "df$price/1000"
##
## $equidist
## [1] FALSE
##
## attr(,"class")
## [1] "histogram"
```

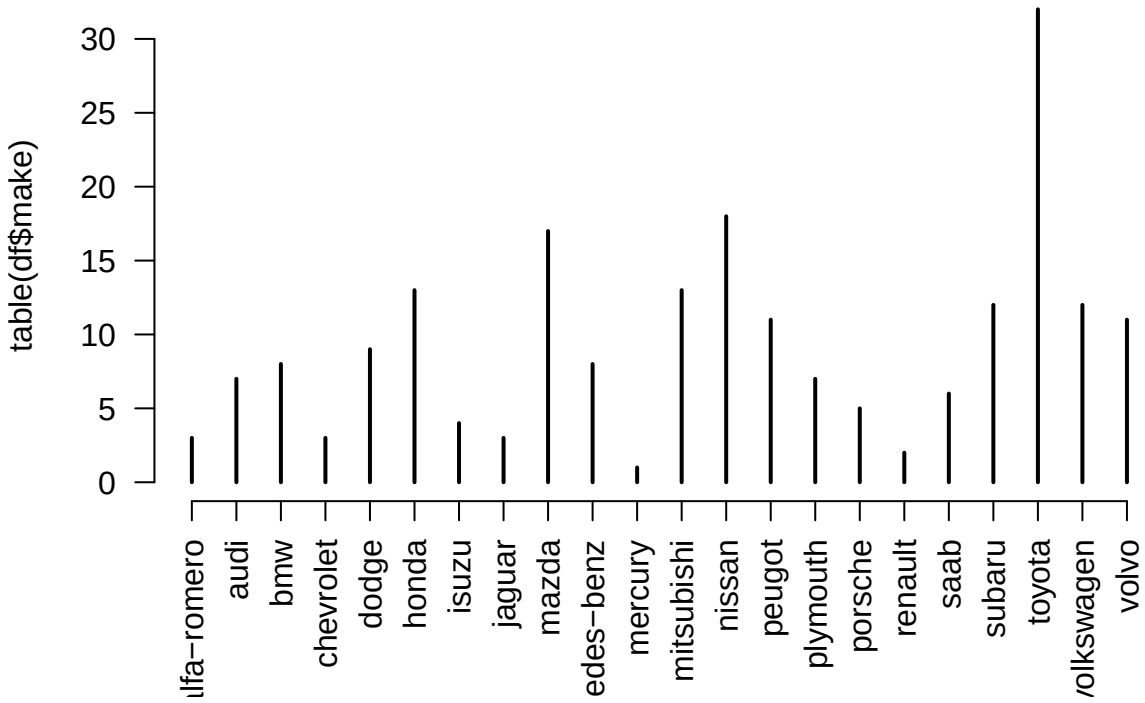
```
# diagramma a barre
plot(df$make)
```



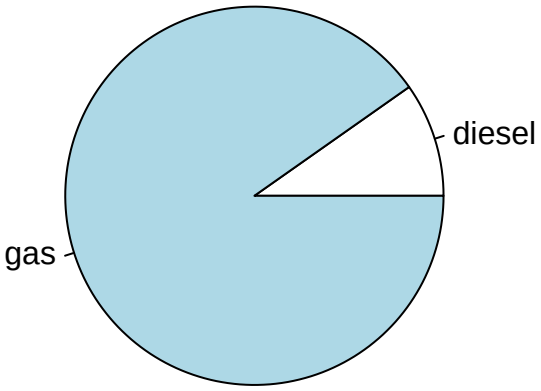

```
plot(df$make,las=2)
```



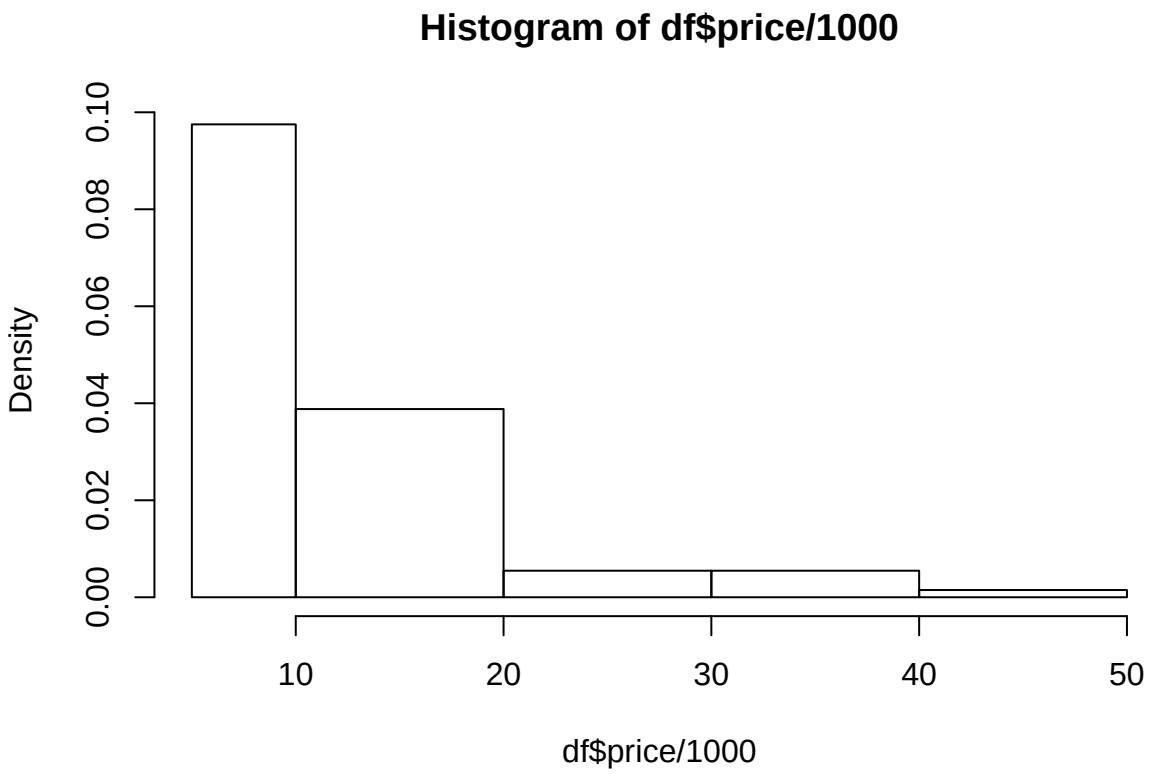

```
# diagramma a bastoncini
plot(table(df$make),las=2)
```



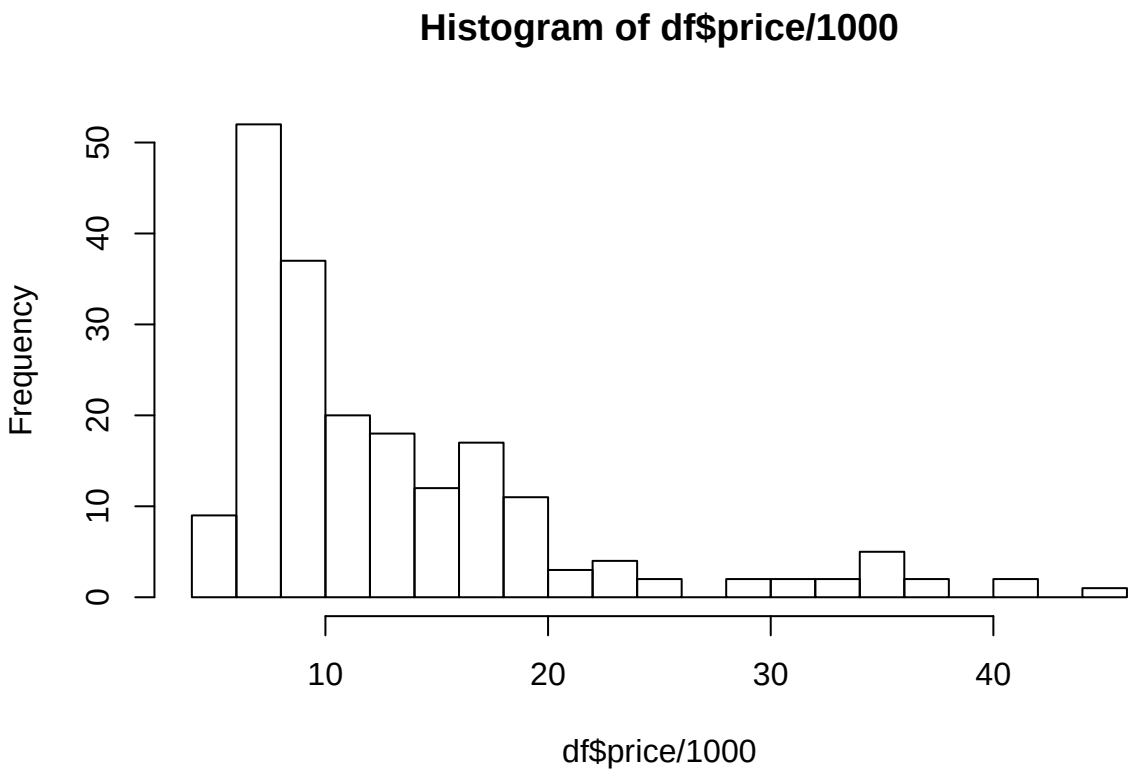

```
#diagramma a torta
pie(table(df$fuel))
```



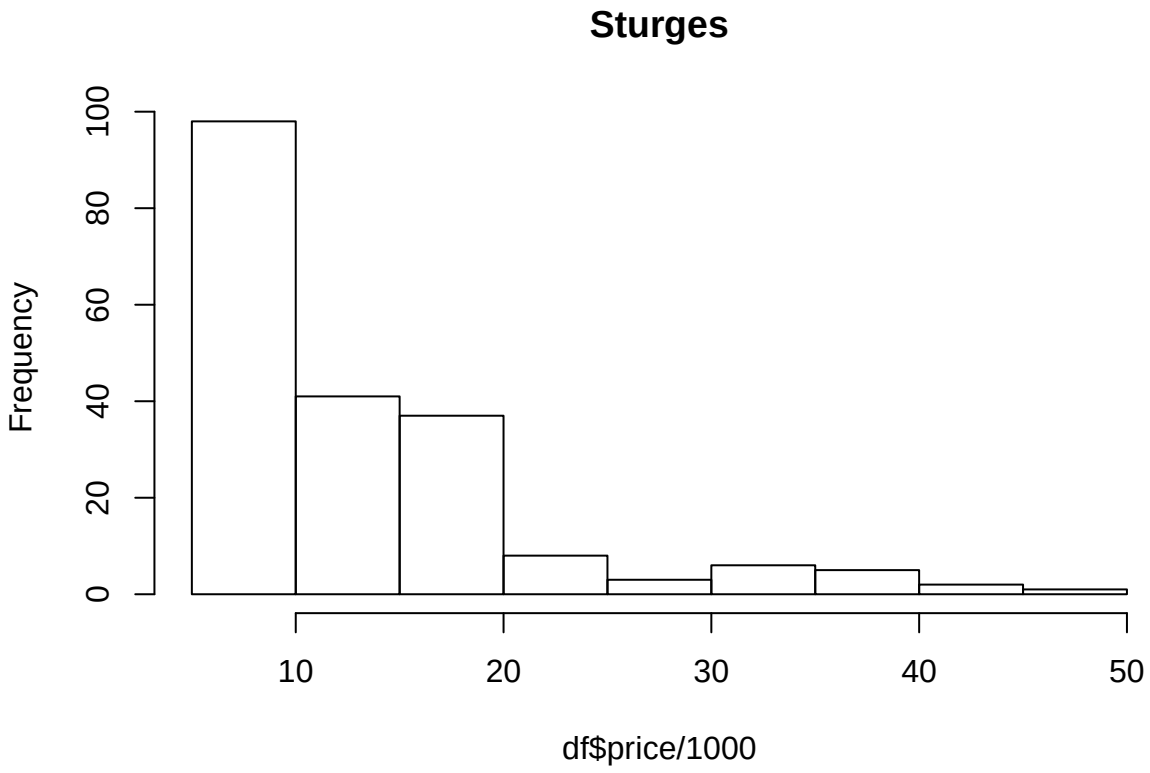

```
# Istogramma

hist(df$price/1000,breaks=c(5,10,20,30,40,50)) # con una divisione in classi predefinita
```




```r
hist(df$price/1000,breaks=20)  # con un numero di classi equiampie predefinita
```




```
hist(df$price/1000,breaks='Sturges',main="Sturges")  # con la formula di Sturges
```



INDICI DI POSIZIONE

```
# MEDIANA
median(df$price)
```

```
## [1] NA
```

```
median(df$price,na.rm=TRUE)
```

```
## [1] 10295
```

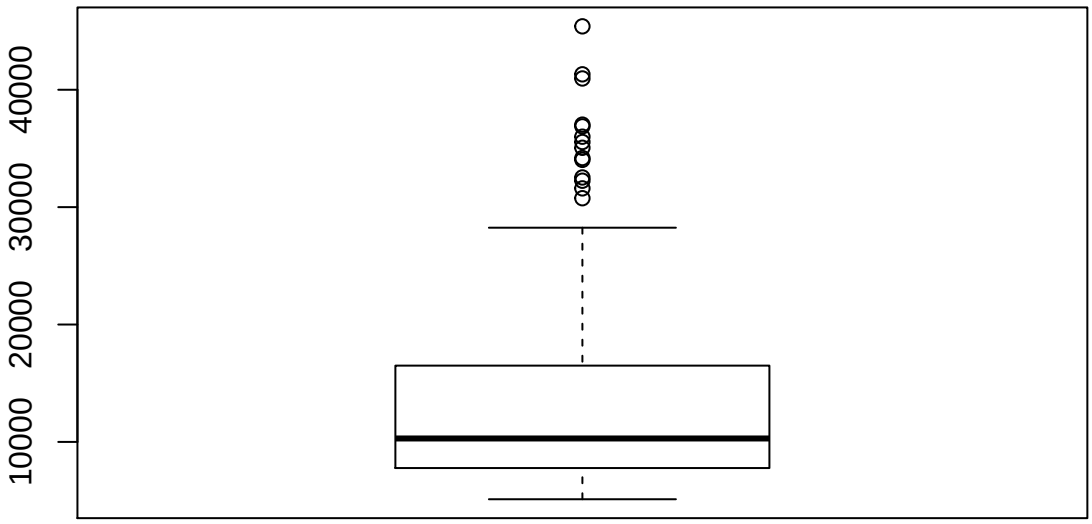
```
me=median(unclass(df$doors),na.rm=TRUE)
me
```

```
## [1] 2
```

```
levels(df$doors)[me]
```

```
## [1] "four"
```

```
# BOXPLOT

b=boxplot(df$price)
```




```
b
```

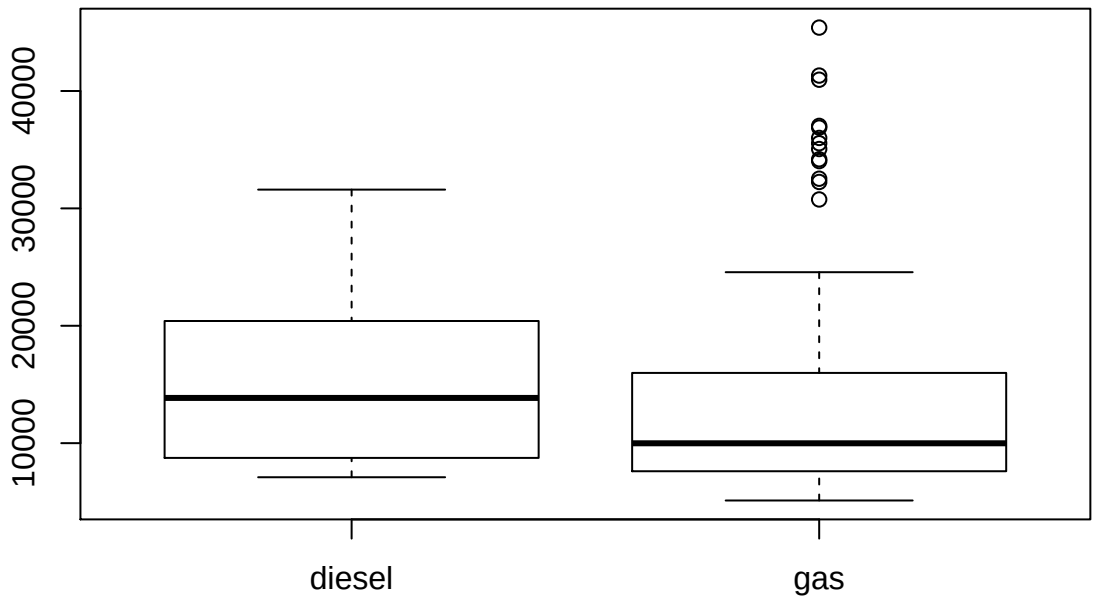
```
## $stats
##       [,1]
## [1,]  5118
## [2,]  7775
## [3,] 10295
## [4,] 16500
## [5,] 28248
## attr(,"class")
##         1
## "integer"
##
## $n
## [1] 201
##
## $conf
##           [,1]
## [1,]  9322.646
## [2,] 11267.354
##
## $out
##  [1] 30760 41315 36880 32250 35550 36000 31600 34184 35056 40960 45400
## [12] 32528 34028 37028
##
## $group
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
##
## $names
## [1] "1"
```

```
b$stats
```

```
##       [,1]
## [1,]  5118
## [2,]  7775
## [3,] 10295
## [4,] 16500
## [5,] 28248
## attr(,"class")
##         1
## "integer"
```

```
b=boxplot(df$price~df$fuel,names=levels(df$fuel))
```




```
# MEDIA ARITMETICA
mean(df$price,na.rm=TRUE)
```

```
## [1] 13207.13
```

```
mean(df$price,trim=0.1,na.rm=TRUE)
```

```
## [1] 11676.94
```

```
summary(df$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
```

```
##    5118    7775   10295   13207   16500   45400       4
```

INDICI DI DISPERSIONE

```
# VARIABILITA' RISPETTO AD UN CENTRO

# Varianza
var(df$price,na.rm=TRUE)
```

```
## [1] 63155863
```

```
# Scarto quadratico medio
sd(df$price,na.rm=TRUE)
```

```
## [1] 7947.066
```

```
sqrt(var(df$price,na.rm=TRUE))
```

```
## [1] 7947.066
```

```
var(df$price,na.rm=TRUE)^(1/2)
```

```
## [1] 7947.066
```

```
# Deviazione assoluta dalla mediana
mad(df$price,na.rm=TRUE)
```

```
## [1] 4901.476
```

```
?mad
```

```
## starting httpd help server ... done
```

```
# DISPERSIONE TRA LE OSSERVAZIONI

# Range
range(df$price,na.rm=TRUE)
```

```
## [1]  5118 45400
```

```
# Differenza interquartile
IQR(df$price,na.rm=TRUE)
```

```
## [1] 8725
```

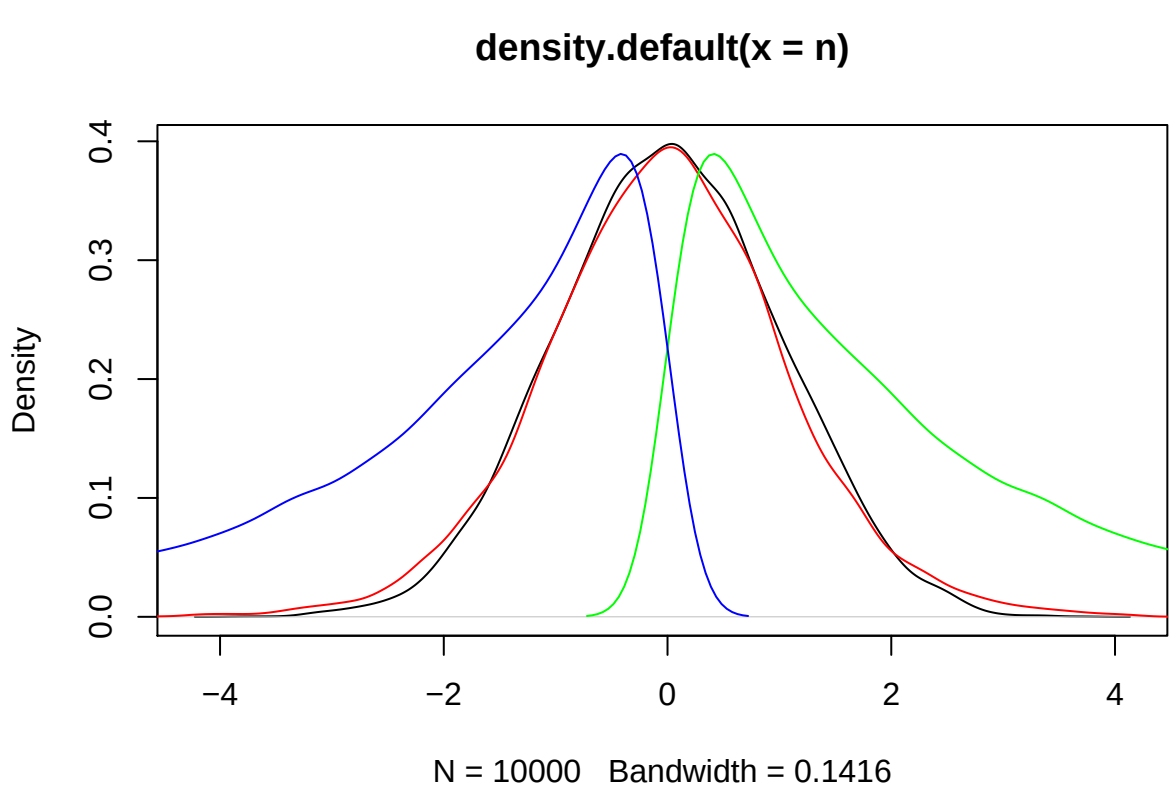
```
quantile(df$price,probs=0.75,na.rm=TRUE)-quantile(df$price,probs=0.25,na.rm=TRUE)
```

```
##  75%
## 8725
```

FORMA DI UNA DISTRIBUZIONE

```
# funzione per la misurazione della curtosi
kurt = function(x){
  x<- x[!is.na(x)]
  n <- length(x)
  s4 <- sqrt(var(x)*(n-1)/n)^4
  mx <- mean(x)
  sk <- sum((x-mx)^4)/s4
  sk/n
}

# funzione per la misurazione della asimmetria
```

```r
skew = function(x){
  x=x[!is.na(x)]
  n <- length(x)
  s3 <- sqrt(var(x)*(n-1)/n)^3
  mx <- mean(x)
  sk <- sum((x-mx)^3)/s3
  sk/n
}

n=rnorm(10000)    # genera 10000 osservazioni da una normale
t=rt(10000,10)    # genera 10000 osservazioni da una t di student con 10 gl
chi=rchisq(10000,2)  # genera 10000 osservazioni da una chi quadrato con 2 gl
plot(density(n))
lines(density(t),col="red")
lines(density(chi),col="green")
lines(density(chi*-1),col="blue")
```




```r
curtosi=c(n=kurt(n),t=kurt(t),chi=kurt(chi), chi2=kurt(chi*-1))
curtosi
```

```
##         n         t       chi      chi2
##  2.898729  3.937897 10.357552 10.357552
```

```r
asimmetria=c(n=skew(n),t=skew(t),chi=skew(chi), chi2=skew(chi*-1))
asimmetria
```

```
##            n            t          chi         chi2
```

```
## -0.003589236  0.021693670  2.105887504 -2.105887504
```