1° Parte R | Data Analytics

VEN

23-06-2022

Contents

Vittorio Amoruso

Nicola Zucchia

Erion Islamay

1. Languages Used

R, Rmd, HTML, CSS

Inclusione dei Packages

```
library(boot)
library(car)
library(ellipse)
library(ggplot2)
library(gridExtra)
library(corrplot)
library(RColorBrewer)
```

```
library(GGally)
library(cluster)
```

Load and Examine of DataSet

Text Edit

1) data presente all'interno del file in formato testuale
2) data conforme alla convenzione csv
3) Header

```
mydata<-read.csv("imports-85.data",header = FALSE)
```

Correttezza e Struttura

```
dim(mydata)
```

```
## [1] 205  26
```

Il Dataframe presenta 205 osservazioni in 26 variabili

Visualizzazione del Contenuto

```
# View(ds)

head(mydata, 5)
```

```
##   V1  V2          V3  V4  V5   V6          V7  V8    V9  V10   V11  V12  V13
## 1  3   ? alfa-romero gas std  two convertible rwd front 88.6 168.8 64.1 48.8
## 2  3   ? alfa-romero gas std  two convertible rwd front 88.6 168.8 64.1 48.8
## 3  1   ? alfa-romero gas std  two   hatchback rwd front 94.5 171.2 65.5 52.4
## 4  2 164        audi gas std four       sedan fwd front 99.8 176.6 66.2 54.3
## 5  2 164        audi gas std four       sedan 4wd front 99.4 176.6 66.4 54.3
##    V14  V15  V16 V17  V18  V19  V20 V21 V22  V23 V24 V25   V26
## 1 2548 dohc four 130 mpfi 3.47 2.68   9 111 5000  21  27 13495
## 2 2548 dohc four 130 mpfi 3.47 2.68   9 111 5000  21  27 16500
## 3 2823 ohcv  six 152 mpfi 2.68 3.47   9 154 5000  19  26 16500
## 4 2337  ohc four 109 mpfi 3.19 3.40  10 102 5500  24  30 13950
## 5 2824  ohc five 136 mpfi 3.19 3.40   8 115 5500  18  22 17450
```

Assegnazione dei nomi - CamelCase

```
names(mydata) <- c('Symboling', 'NormalizedLosses', 'Make', 'FuelType', 'Aspiration', 'NumOfDoors', 'Bo
```

Structure + Data Wrangling

Qualitative vs Quantitative

```
str(mydata)
```

```
## 'data.frame':    205 obs. of  26 variables:
##  $ Symboling       : int  3 3 1 2 2 2 1 1 1 0 ...
##  $ NormalizedLosses: chr  "?" "?" "?" "164" ...
##  $ Make            : chr  "alfa-romero" "alfa-romero" "alfa-romero" "audi" ...
##  $ FuelType        : chr  "gas" "gas" "gas" "gas" ...
##  $ Aspiration      : chr  "std" "std" "std" "std" ...
##  $ NumOfDoors      : chr  "two" "two" "two" "four" ...
##  $ BodyStyle       : chr  "convertible" "convertible" "hatchback" "sedan" ...
##  $ DriveWheels     : chr  "rwd" "rwd" "rwd" "fwd" ...
##  $ EngineLocation  : chr  "front" "front" "front" "front" ...
##  $ 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 ...
##  $ EngineType      : chr  "dohc" "dohc" "ohcv" "ohc" ...
##  $ NumOfCylinders  : chr  "four" "four" "six" "four" ...
##  $ EngineSize      : int  130 130 152 109 136 136 136 136 131 131 ...
##  $ FuelSystem      : chr  "mpfi" "mpfi" "mpfi" "mpfi" ...
##  $ Bore            : chr  "3.47" "3.47" "2.68" "3.19" ...
##  $ Stroke          : chr  "2.68" "2.68" "3.47" "3.40" ...
##  $ CompressionRatio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 7 ...
##  $ Horsepower      : chr  "111" "111" "154" "102" ...
##  $ PeakRpm         : chr  "5000" "5000" "5000" "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           : chr  "13495" "16500" "16500" "13950" ...
```

```
mydata$Symboling      <- factor(mydata$Symboling)
mydata$Make           <- factor(mydata$Make)
mydata$FuelType       <- factor(mydata$FuelType)
mydata$Aspiration     <- factor(mydata$Aspiration)
mydata$NumOfDoors     <- factor(mydata$NumOfDoors)
mydata$BodyStyle      <- factor(mydata$BodyStyle)
mydata$DriveWheels    <- factor(mydata$DriveWheels)
mydata$EngineLocation <- factor(mydata$EngineLocation)
mydata$EngineType     <- factor(mydata$EngineType)
mydata$NumOfCylinders <- factor(mydata$NumOfCylinders)
mydata$FuelSystem     <- factor(mydata$FuelSystem)

mydata$NormalizedLosses <- as.numeric(mydata$NormalizedLosses)
mydata$Bore             <- as.numeric(mydata$Bore)
mydata$Stroke           <- as.numeric(mydata$Stroke)
mydata$Horsepower       <- as.numeric(mydata$Horsepower)
mydata$PeakRpm          <- as.numeric(mydata$PeakRpm)
mydata$Price            <- as.numeric(mydata$Price)
```

```
str(mydata)
```

```
## 'data.frame':    205 obs. of  26 variables:
```

```
##  $ Symboling       : Factor w/ 6 levels "-2","-1","0",..: 6 6 4 5 5 5 4 4 4 3 ...
##  $ NormalizedLosses: num  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 ...
##  $ FuelType        : 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 ...
##  $ NumOfDoors      : Factor w/ 3 levels "?","four","two": 3 3 3 2 2 3 2 2 2 3 ...
##  $ BodyStyle       : 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 ...
##  $ EngineType      : Factor w/ 7 levels "dohc","dohcv",..: 1 1 6 4 4 4 4 4 4 4 ...
##  $ NumOfCylinders  : 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 ...
##  $ CompressionRatio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 7 ...
##  $ Horsepower      : num  111 111 154 102 115 110 110 110 140 160 ...
##  $ PeakRpm         : num  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           : num  13495 16500 16500 13950 17450 ...
```

Eliminazione V. non di Interesse

```
mydata <- mydata[ , - c(1, 2) ]
```

Gestione degli NA, '?'

```
mydata <- na.omit(mydata)
```

```
cleaner <- function(ds, sc, show = FALSE) {

# ds = DataSet , sc = SpecialCharacter

  unlist <- as.numeric( )

  for (i in 1:dim(ds)[1])

  {

    dr <- ds[ i , ]

    if ( any(dr == sc) ) unlist[length(unlist) + 1] <- i

  }

  if (show == TRUE)
```

```
  {

    print('Le unita eliminate sono.. ')

    print(unlist)

  }

  return(ds[ - unlist , ])

}
```

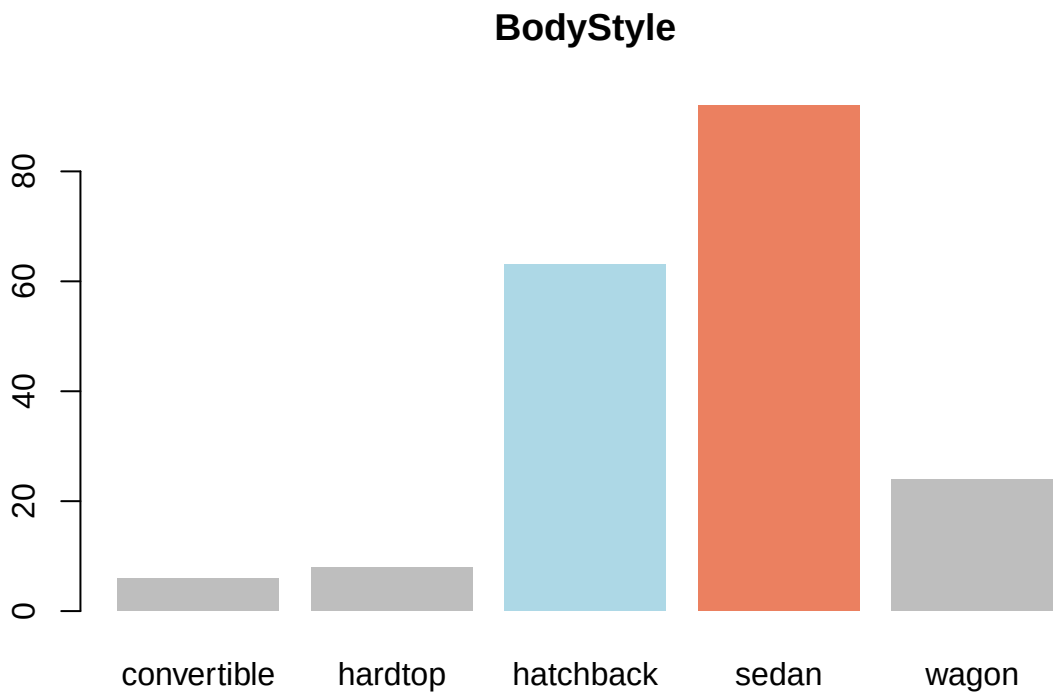
```
mydata <- cleaner(mydata, '?', show = TRUE)
```

```
## [1] "Le unita eliminate sono.. "
## [1] 27 57
```

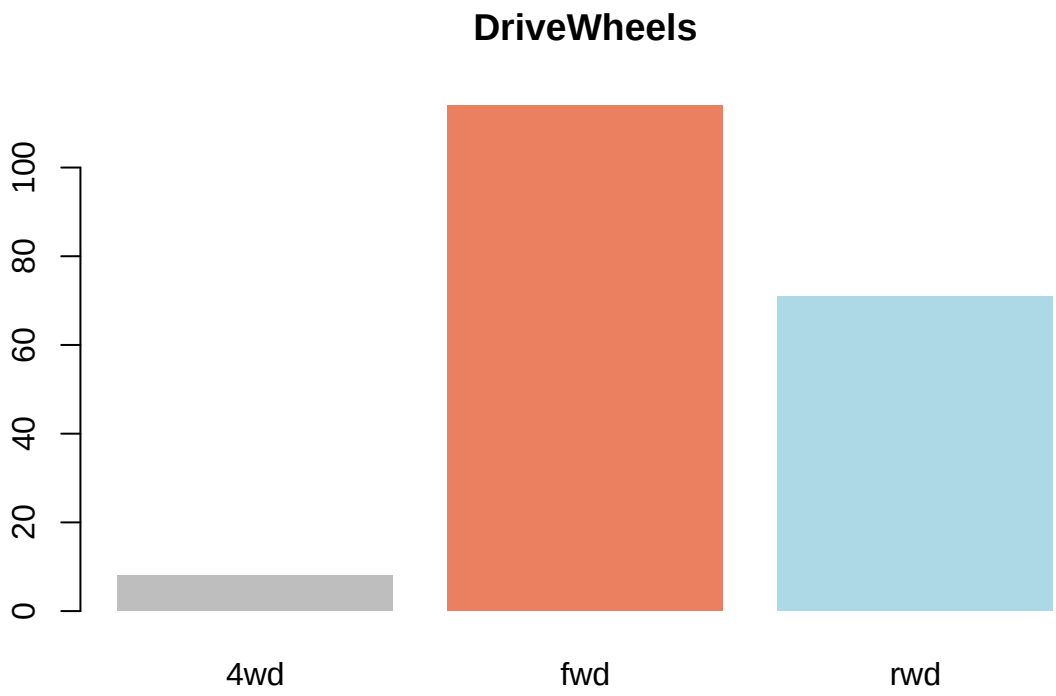
Categoriali Single

```
Make           <- table(mydata$Make)
FuelType       <- table(mydata$FuelType)
Aspiration     <- table(mydata$Aspiration)
NumOfDoors     <- table(mydata$NumOfDoors)
BodyStyle      <- table(mydata$BodyStyle)
DriveWheels    <- table(mydata$DriveWheels)
EngineLocation <- table(mydata$EngineLocation)
EngineType     <- table(mydata$EngineType)
NumOfCylinders <- table(mydata$NumOfCylinders)
FuelSystem     <- table(mydata$FuelSystem)
```

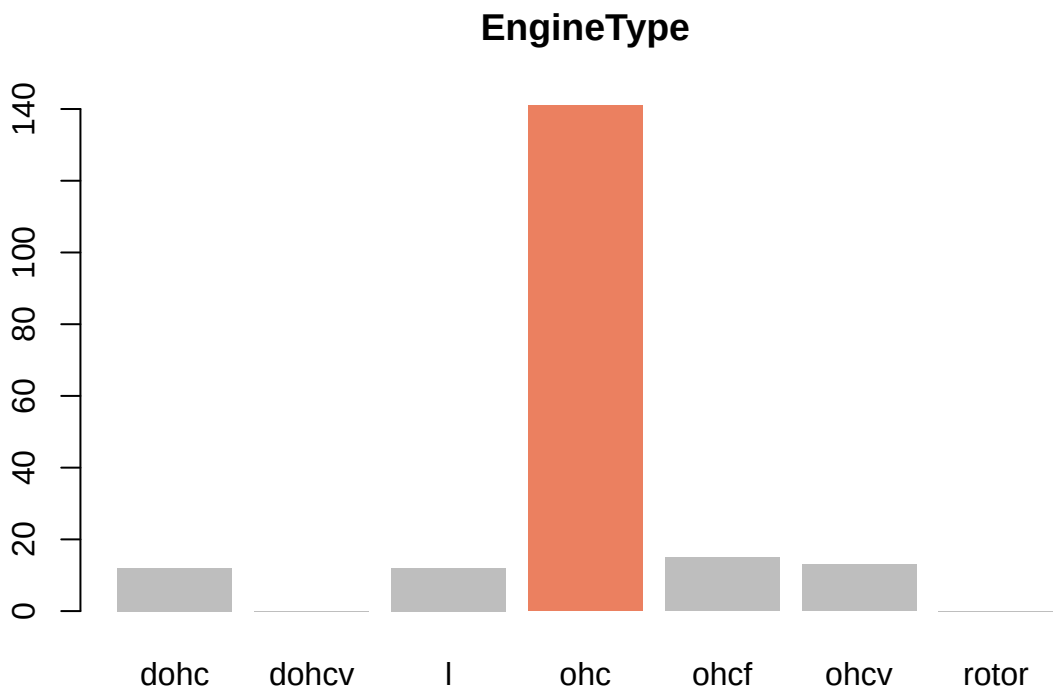
```
barplot(BodyStyle, col = c("grey", "grey", "lightblue", "#eb8060", "grey"), border=NA, main = "BodyStyl
```



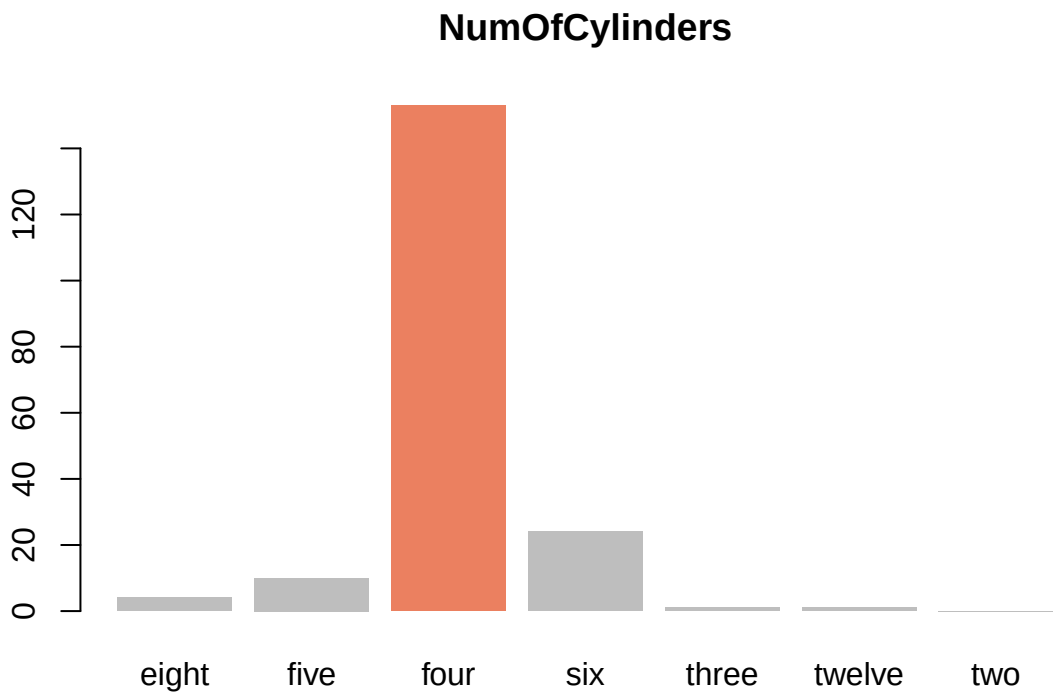

```
barplot(DriveWheels, col = c("grey", "#eb8060", "lightblue"), border=NA, main = "DriveWheels")
```



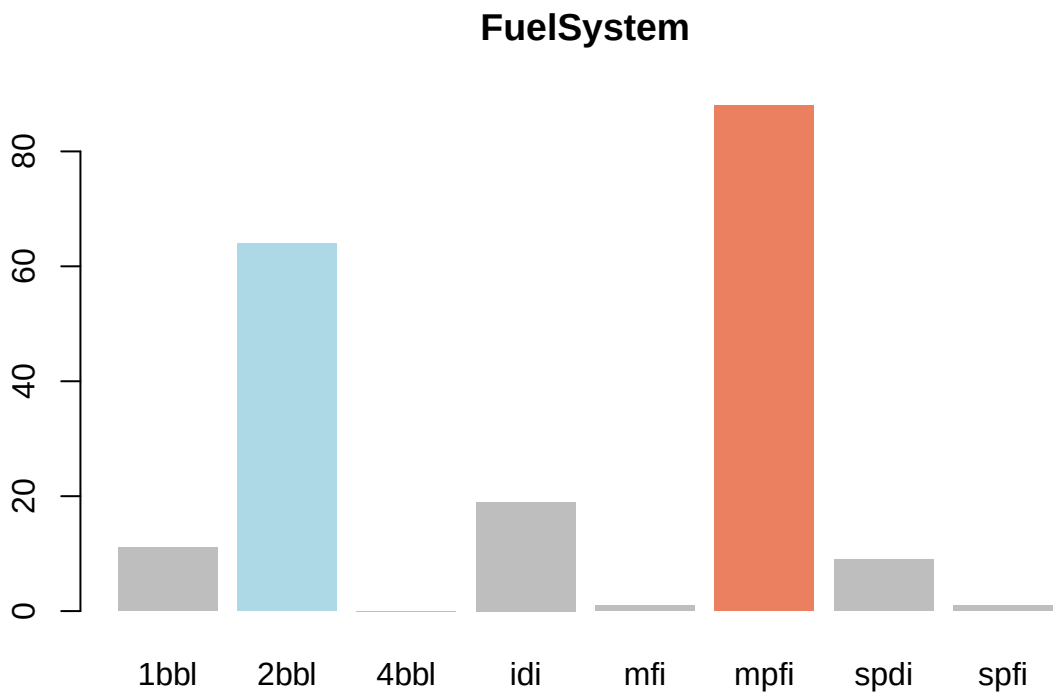

```
barplot(EngineType, col = c("grey", "grey", "grey", "#eb8060", "grey", "grey", "grey"), border=NA, main
```




```
barplot(NumOfCylinders, col = c("grey", "grey", "#eb8060", "grey", "grey", "grey", "grey"), border=NA, n
```




```
barplot(FuelSystem, col = c("grey", "lightblue", "grey", "grey", "grey", "#eb8060", "grey", "grey"), bo
```



Categoriali Bivariate

```
table(mydata$FuelType, mydata$Aspiration)
```

```
##
##          std turbo
##   diesel   6    13
##   gas    152    22
```

```
chisq.test(table(mydata$FuelType, mydata$Aspiration))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(mydata$FuelType, mydata$Aspiration)
## X-squared = 32.238, df = 1, p-value = 1.364e-08
```

```
table(mydata$DriveWheels, mydata$FuelType)
```

```
##
##       diesel gas
##   4wd      0   8
```

```
##   fwd      8 106
##   rwd     11  60
```

```
chisq.test(table(mydata$DriveWheels, mydata$FuelType))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(mydata$DriveWheels, mydata$FuelType)
## X-squared = 4.4523, df = 2, p-value = 0.1079
```

Considerazioni

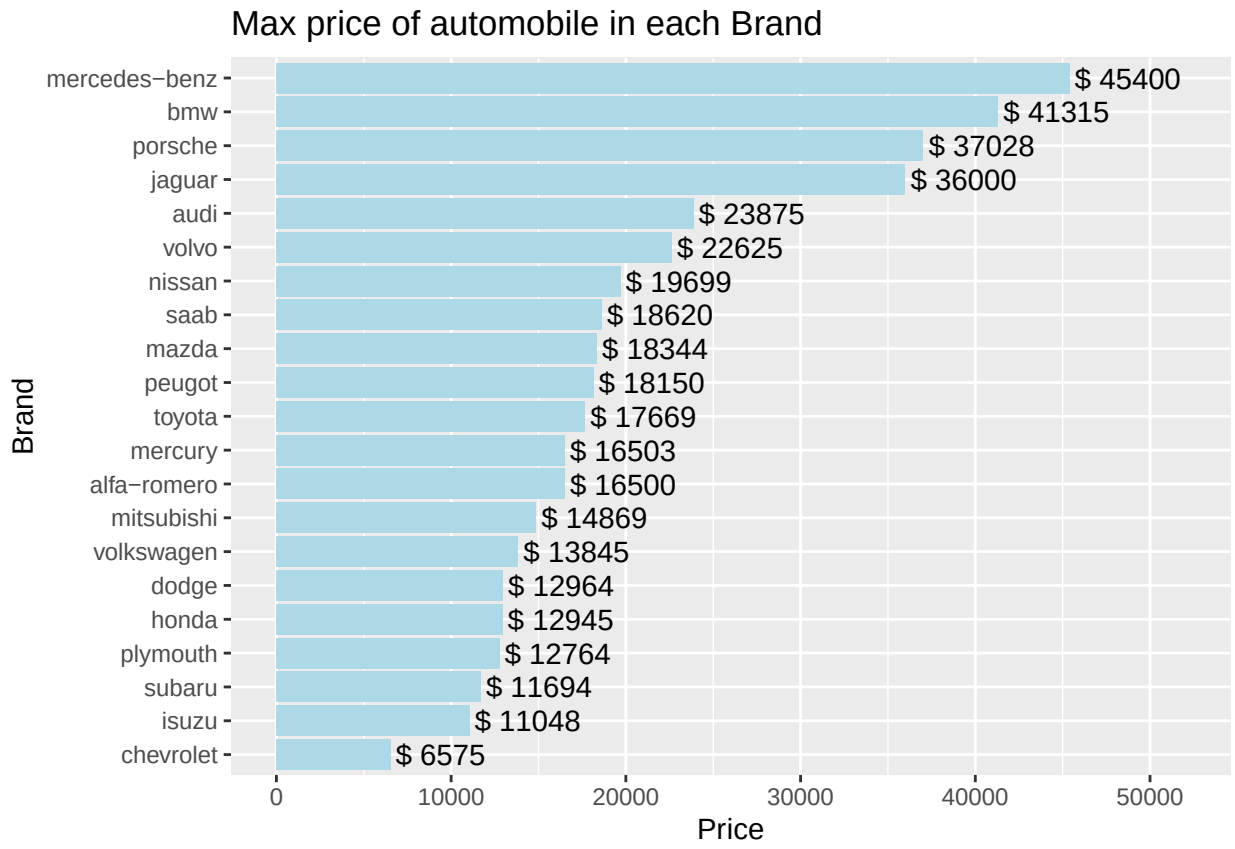
```
#plot maximum price according to the automobile maker i.e., brand
maxPrice <- aggregate(mydata$Price , by = list(mydata$Make), FUN = "max")

names(maxPrice) <- c("Brand", "Price")

maxPrice <- maxPrice[ order(maxPrice$Price) , ]

maxPrice$Brand <- factor(maxPrice$Brand, levels = maxPrice$Brand)

ggplot(data = maxPrice, aes(x = Price, y = Brand)) +
    geom_bar(stat = "identity", fill = "lightblue") +
    geom_text(aes(label = paste0("$ ", Price)), hjust = -0.05) +
    coord_cartesian(xlim = c(0, 52000)) +
    labs(title = "Max price of automobile in each Brand")
```



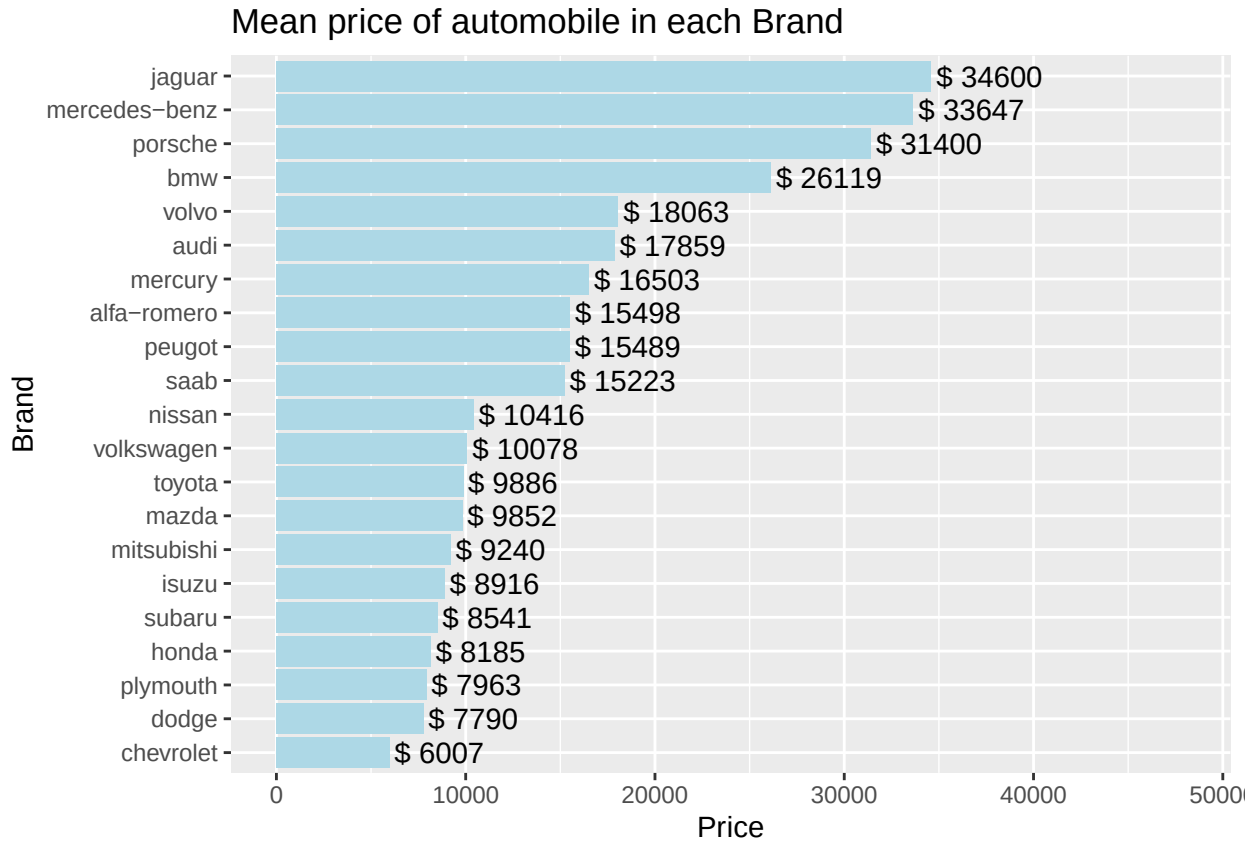

```
meanPrice <- aggregate(mydata$Price, by = list(mydata$Make), FUN = "mean")

names(meanPrice) <- c("Brand","Price")

meanPrice <- meanPrice[ order(meanPrice$Price) , ]

meanPrice$Brand <- factor(meanPrice$Brand, levels = meanPrice$Brand)

ggplot(data = meanPrice, aes(x = Price, y = Brand)) +
    geom_bar(stat = "identity", fill = "Lightblue") +
    geom_text(aes(label = paste0("$ ", round(Price))), hjust = -0.05) +
    coord_cartesian(xlim = c(0,48000)) +
    labs(title = "Mean price of automobile in each Brand")
```



2. Analisi Variabili QUANTITATIVE

```
mydata.num<- mydata[,c("WheelBase","Length","Width","Height","CurbWeight",
                       "EngineSize","Bore","Stroke","CompressionRatio","Horsepower",
                       "PeakRpm","CityMpg","HighwayMpg","Price")]
summary(mydata.num)
```

```
##    WheelBase          Length          Width           Height
##  Min.   : 86.60   Min.   :141.1   Min.   :60.30   Min.   :47.80
##  1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.10   1st Qu.:52.00
##  Median : 97.00   Median :173.2   Median :65.40   Median :54.10
##  Mean   : 98.92   Mean   :174.3   Mean   :65.89   Mean   :53.87
##  3rd Qu.:102.40   3rd Qu.:184.6   3rd Qu.:66.90   3rd Qu.:55.70
##  Max.   :120.90   Max.   :208.1   Max.   :72.00   Max.   :59.80
##    CurbWeight     EngineSize          Bore           Stroke
##  Min.   :1488   Min.   : 61.0   Min.   :2.540   Min.   :2.070
##  1st Qu.:2145   1st Qu.: 98.0   1st Qu.:3.150   1st Qu.:3.110
##  Median :2414   Median :120.0   Median :3.310   Median :3.290
##  Mean   :2562   Mean   :128.1   Mean   :3.331   Mean   :3.249
##  3rd Qu.:2952   3rd Qu.:146.0   3rd Qu.:3.590   3rd Qu.:3.410
##  Max.   :4066   Max.   :326.0   Max.   :3.940   Max.   :4.170
##  CompressionRatio   Horsepower       PeakRpm        CityMpg
##  Min.   : 7.00    Min.   : 48.0   Min.   :4150   Min.   :13.00
```

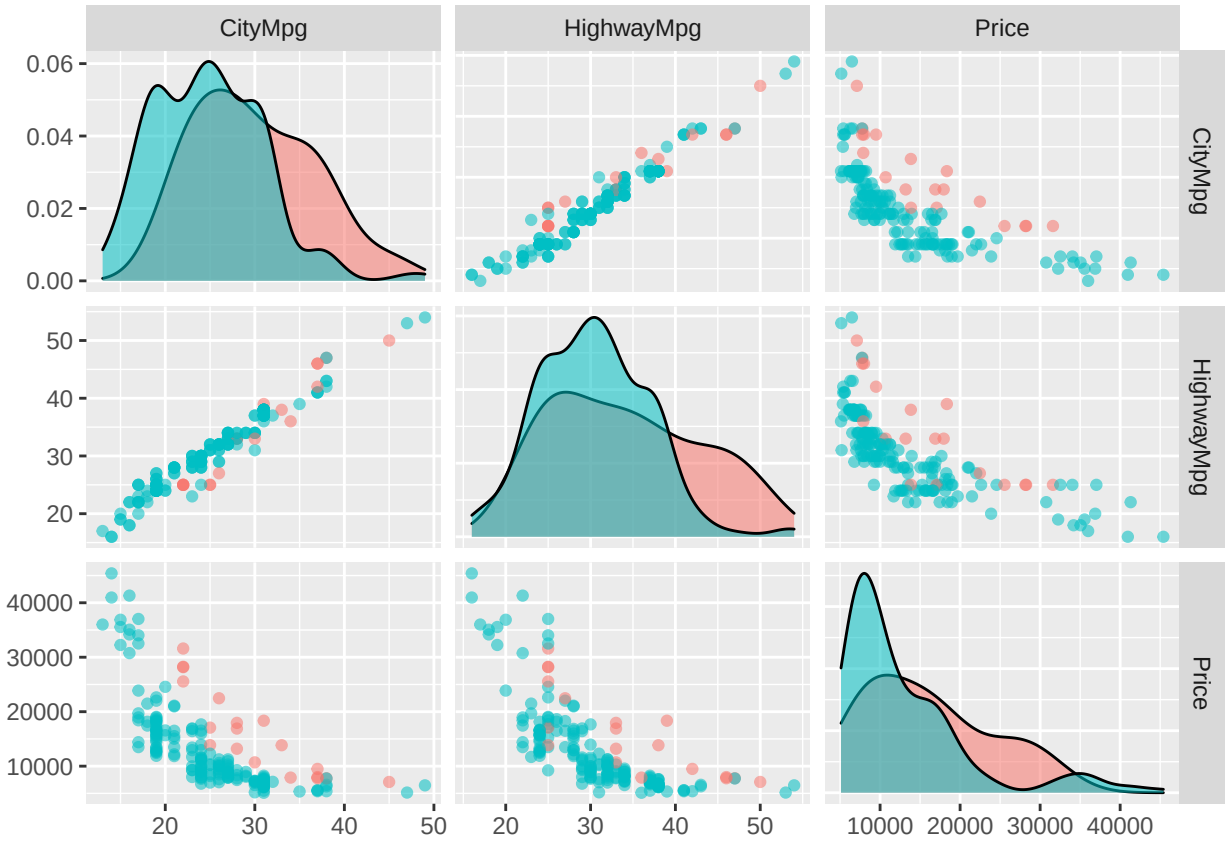
```
##  1st Qu.: 8.50    1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00
##  Median : 9.00    Median : 95.0   Median :5100   Median :25.00
##  Mean   :10.14    Mean   :103.5   Mean   :5100   Mean   :25.33
##  3rd Qu.: 9.40    3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00
##  Max.   :23.00    Max.   :262.0   Max.   :6600   Max.   :49.00
##    HighwayMpg        Price
##  Min.   :16.00   Min.   : 5118
##  1st Qu.:25.00   1st Qu.: 7738
##  Median :30.00   Median :10245
##  Mean   :30.79   Mean   :13285
##  3rd Qu.:34.00   3rd Qu.:16515
##  Max.   :54.00   Max.   :45400
```

- Diamo uno sguardo ad alcune variabili quantitative e alle loro distribuzioni

```
ggpairs(mydata.num,
        columns = c("Length","Width","CurbWeight", "Price"),
        aes(color = mydata$FuelType,  # Color by group (cat. variable)
            alpha = 0.5), upper = list(continuous = "points"))
```


```
ggpairs(mydata.num,
        columns = c("EngineSize","Horsepower", "Price"),
        aes(color = mydata$FuelType,  # Color by group (cat. variable)
            alpha = 0.5), upper = list(continuous = "points"))
```

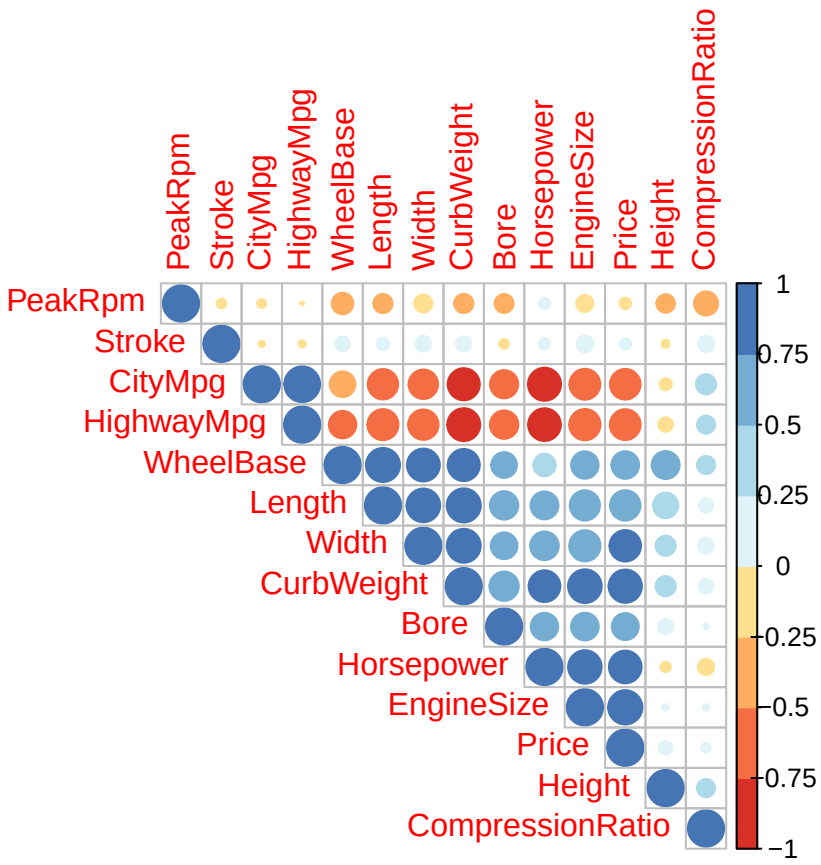

```
ggpairs(mydata.num,
        columns = c("CityMpg","HighwayMpg","Price"),
        aes(color = mydata$FuelType,  # Color by group (cat. variable)
            alpha = 0.5), upper = list(continuous = "points"))
```



ANALISI DELLE CORRELAZIONI

- Guardiamo a un quadro generale delle varie correlazioni tra le variabili quantitative, andando a studiare nel dettaglio le più significative.

```
corr<- round(cor(mydata.num),2)
corrplot(corr, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```



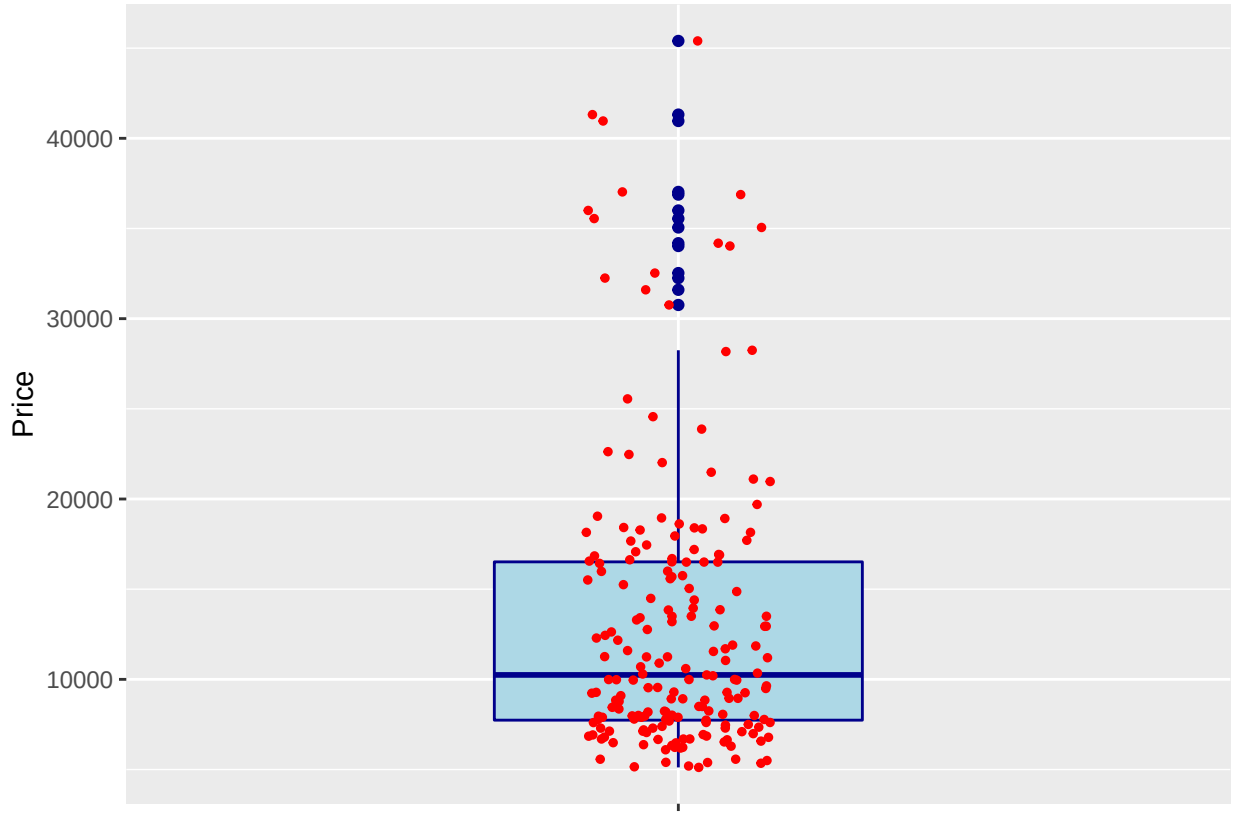
ANALISI UNIVARIATA

Studiamo le distribuzioni delle principali variabili quantitative interessanti mostrando qualche istogramma e curve di densità.

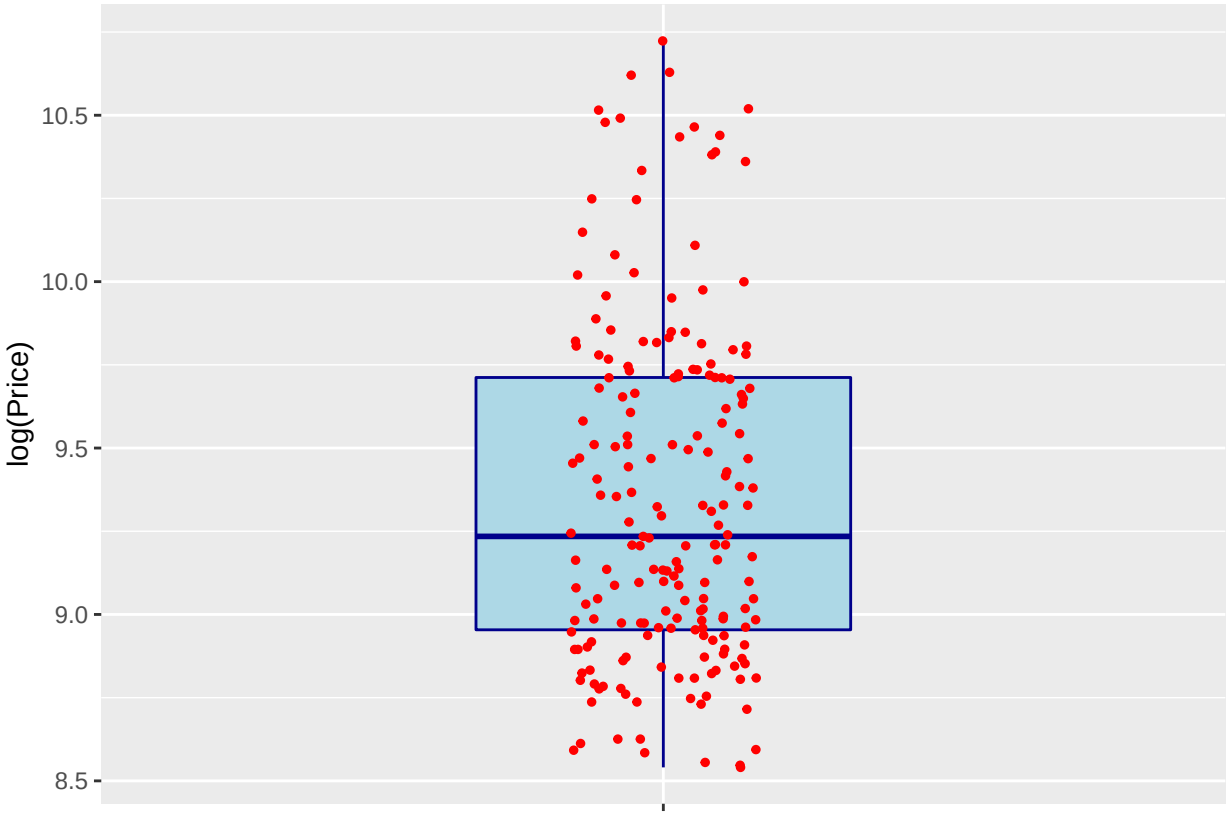
Si osserva che, per alcune variabili, la scarsa ampiezza campionaria non rende efficace la visualizzazione dei dati tramite istogrammi.

PRICE

```
ggplot(mydata, aes(x = "", y = Price)) +
  geom_boxplot(width = 0.4, colour="darkblue", fill="lightblue") +
  geom_jitter( colour="red",
               width = 0.1, size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(x = NULL)
```

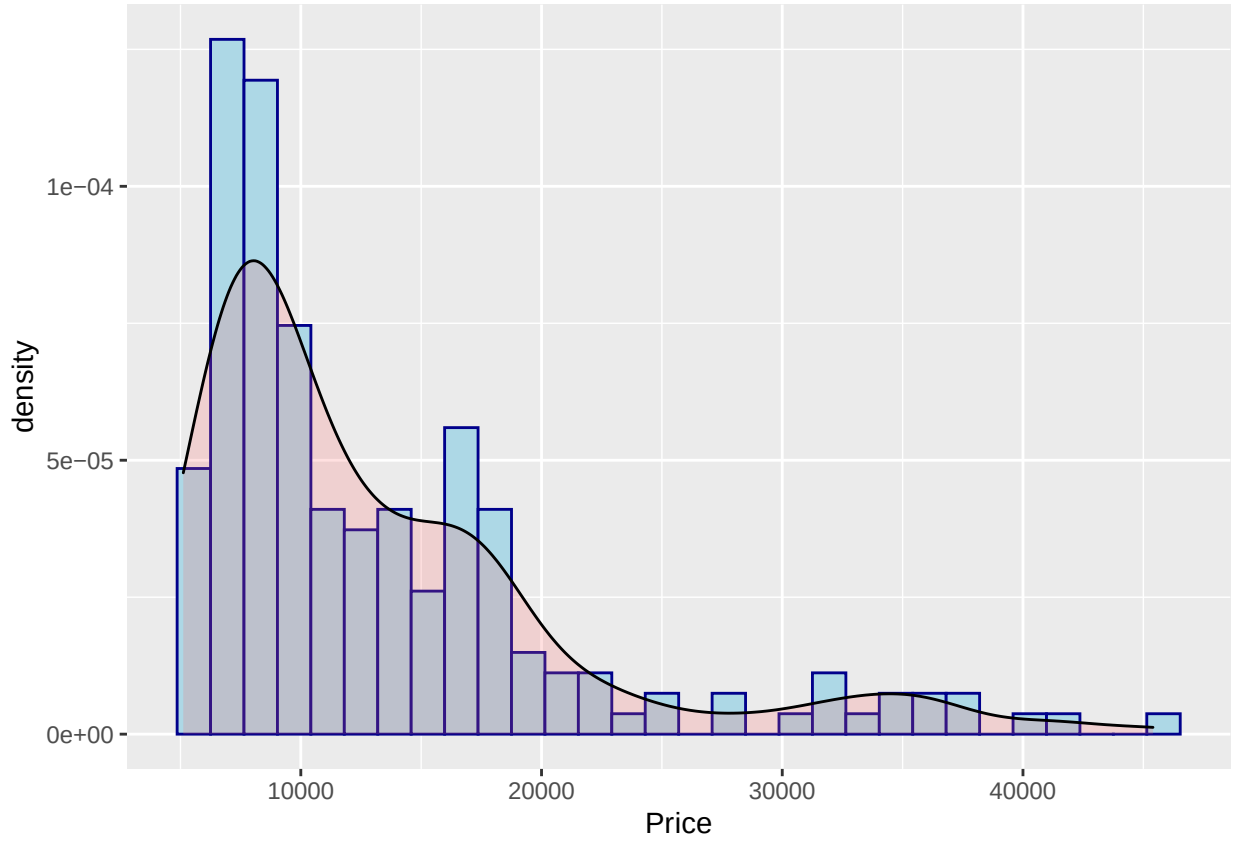



```
ggplot(mydata, aes(x = "", y = log(Price))) +
  geom_boxplot(width = 0.4, colour="darkblue", fill="lightblue") +
  geom_jitter( colour="red",
               width = 0.1, size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  labs(x = NULL)
```

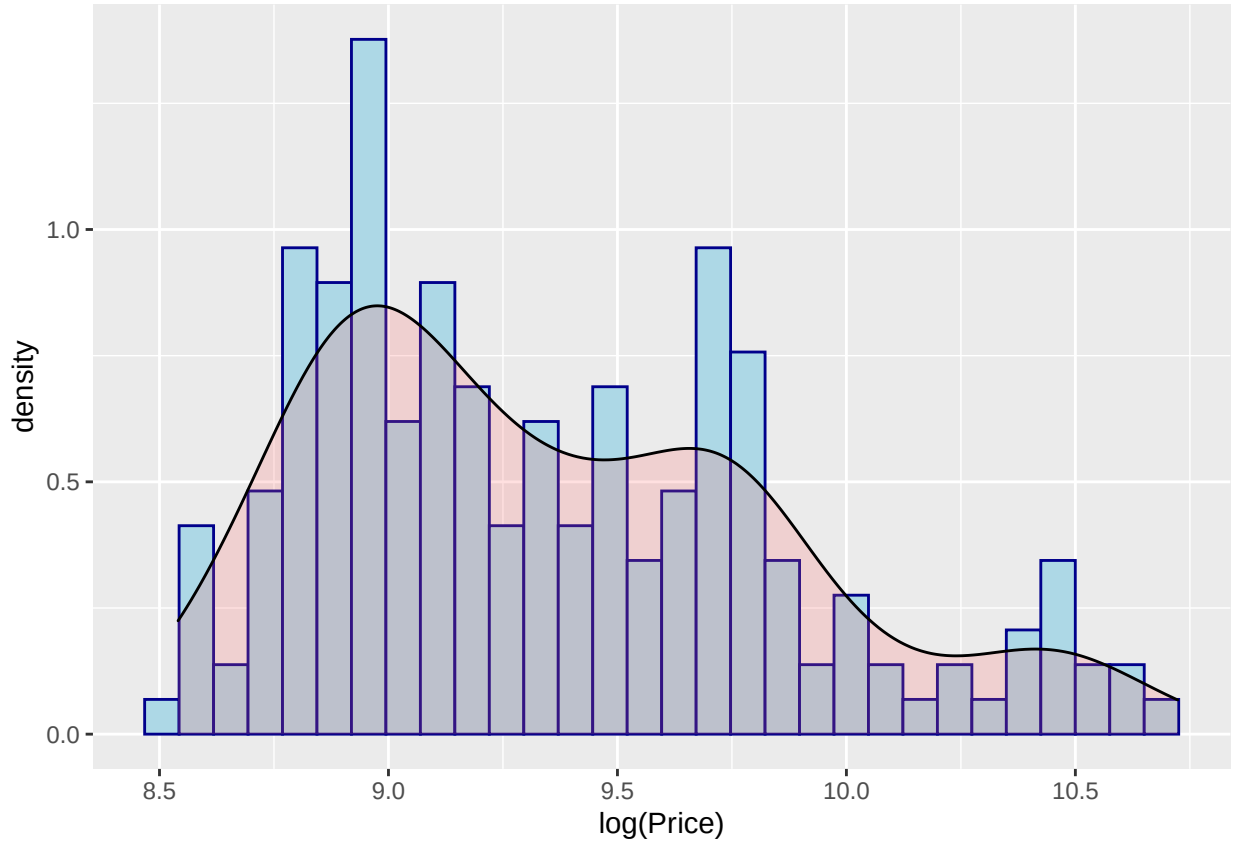


Come si nota dal boxplot, la variabile Price presenta una asimmetria marcata e molti outliers nella coda destra della distribuzione. Si osserva come fare una trasformazione logaritmica renda la distribuzione più regolare. Per esserne sicuri facciamo un confronto con la Gaussiana.

```
#si vede che il prezzo è fortemente assimetrico quindi forse da trasformare in log
#plot(density(mydata$price))
ggplot(mydata, aes(x=Price)) +
 geom_histogram(aes(y=..density..), colour="darkblue", fill="lightblue")+
 geom_density(alpha=.2, fill="#FF6666")
```




```
ggplot(mydata, aes(x=log(Price))) +
 geom_histogram(aes(y=..density..), colour="darkblue", fill="lightblue")+
 geom_density(alpha=.2, fill="#FF6666")
```



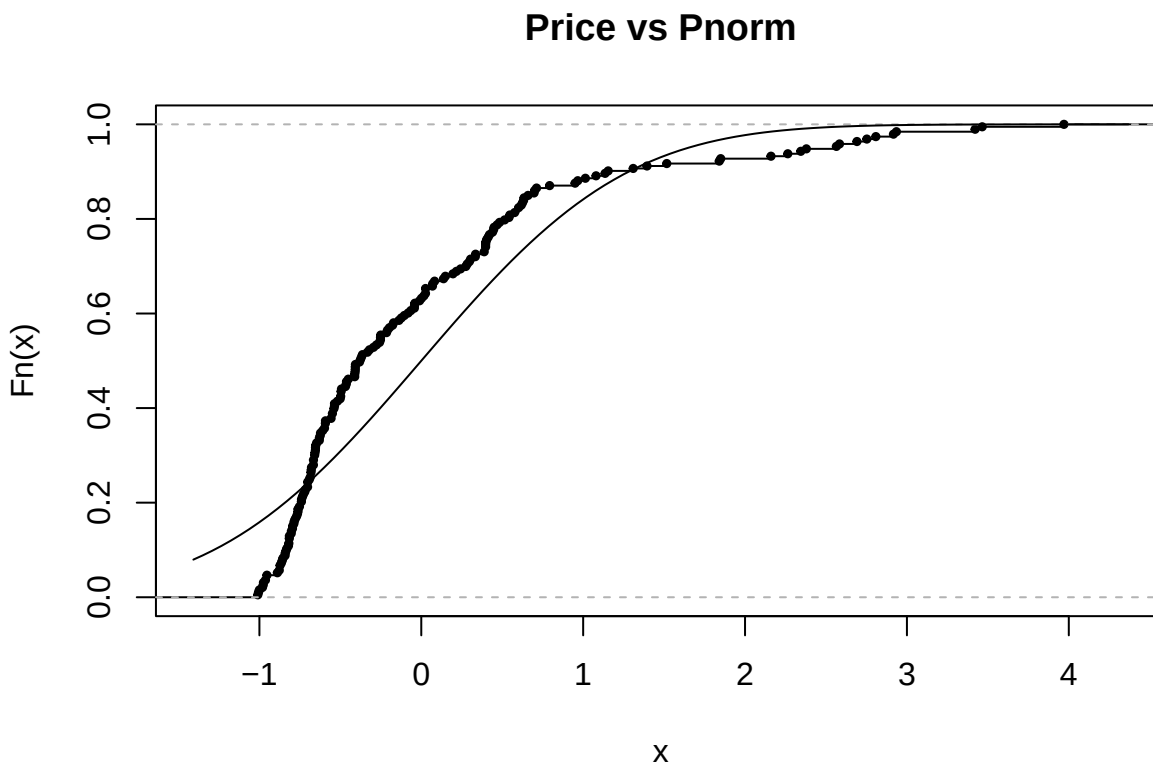
La variabile logPrice sembra avere una tendenza trimodale a partire dal grafico della sua densità.

Si osserva come fare una trasformazione logaritmica renda la distribuzione più regolare. Per esserne sicuri facciamo un confronto con la Gaussiana.

CONFRONTI CON LA GAUSSIANA

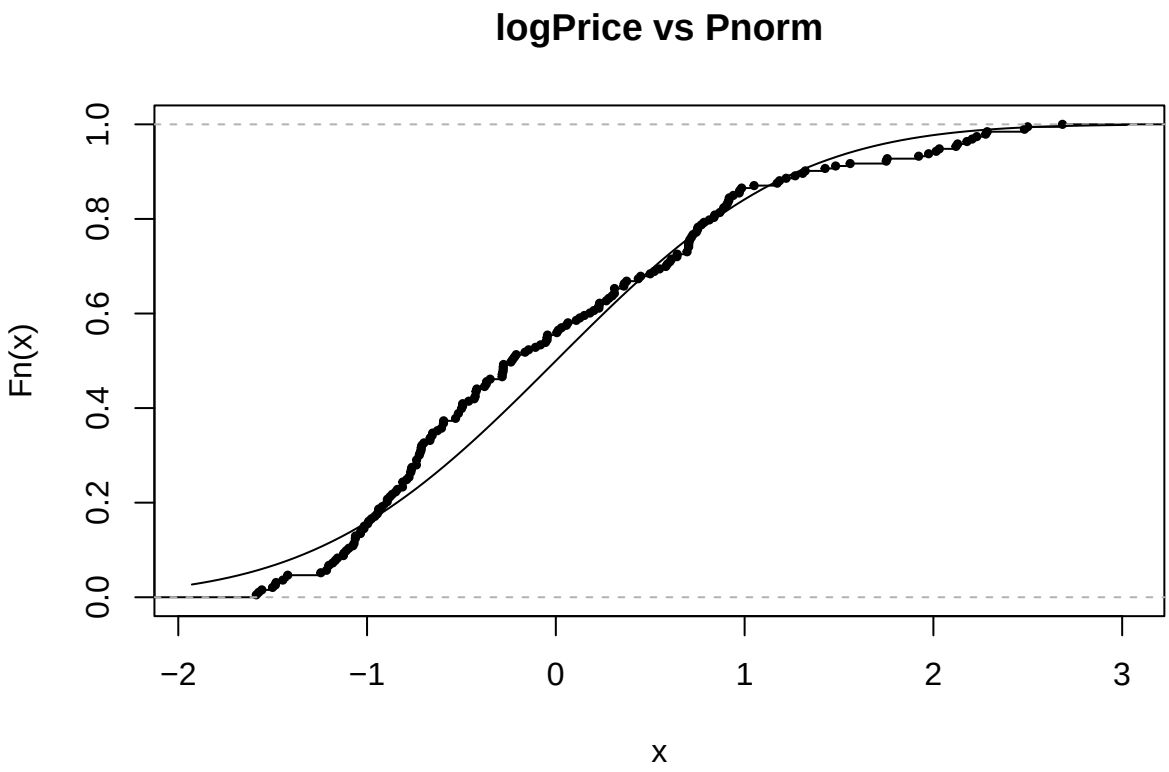
Testiamo la normalità della variabile price e della sua trasformata logPrice.

```
plot(ecdf(scale(mydata$Price)), cex=0.5, main="Price vs Pnorm")
curve(pnorm(x), add=TRUE)
```



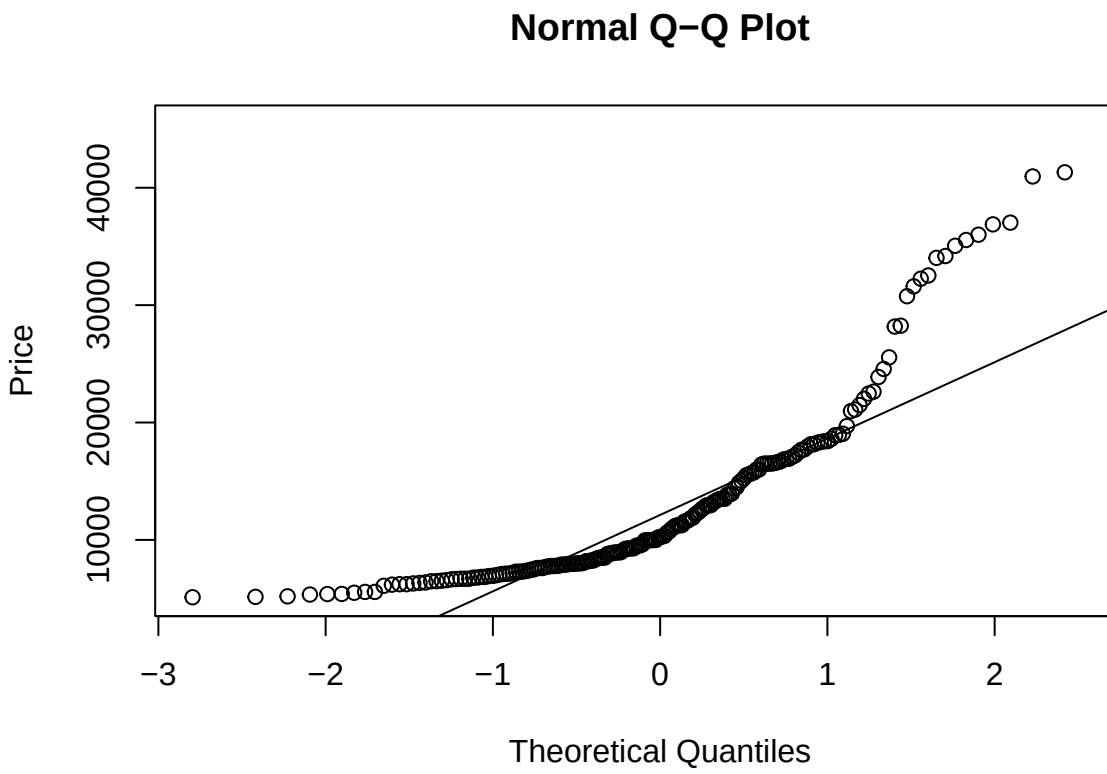
PRICE vs PNORM

```
plot(ecdf(scale(log(mydata$Price))), cex=0.5, main="logPrice vs Pnorm")
curve(pnorm(x), add=TRUE)
```



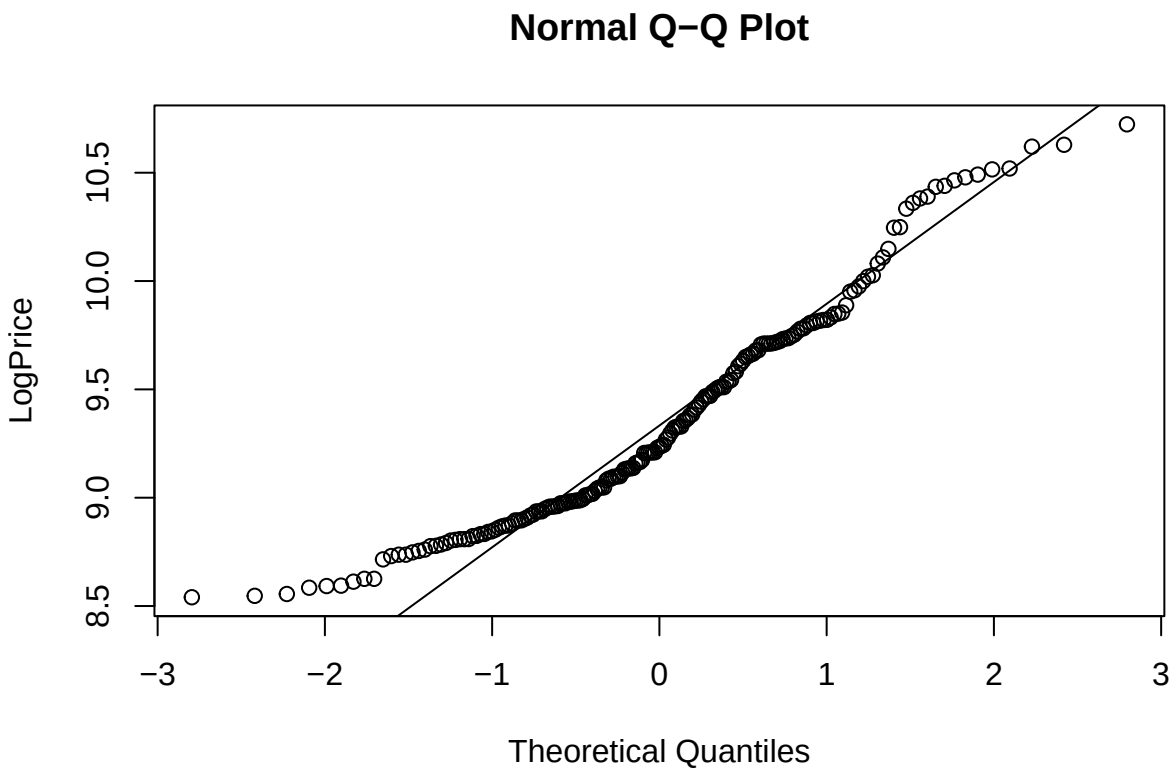
Il confronto con la funzione di ripartizione empirica suggerisce che una la variabile trasformata tenda a distribuirsi più normalmente della variabile di partenza.

```
qqnorm(mydata$Price, ylab = "Price")
qqline(mydata$Price)
```



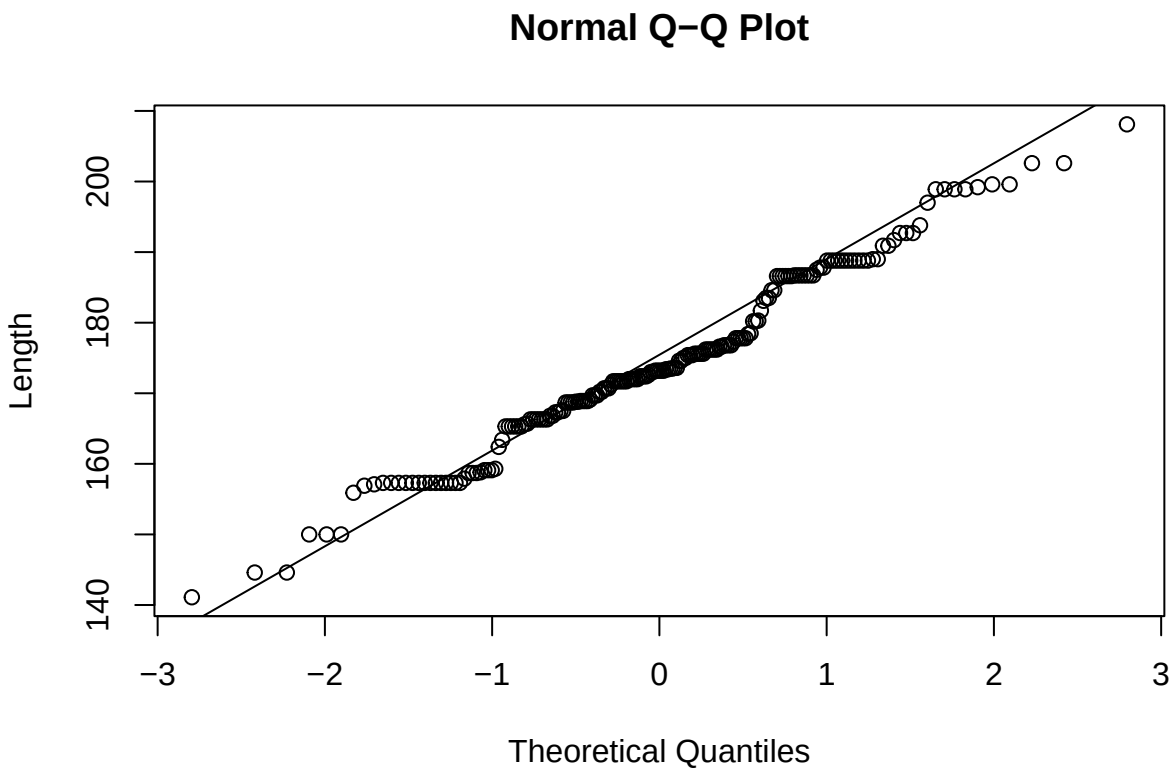
Confrontiamo i QQplot

```
qqnorm(log(mydata$Price), ylab = "LogPrice")
qqline(log(mydata$Price))
```

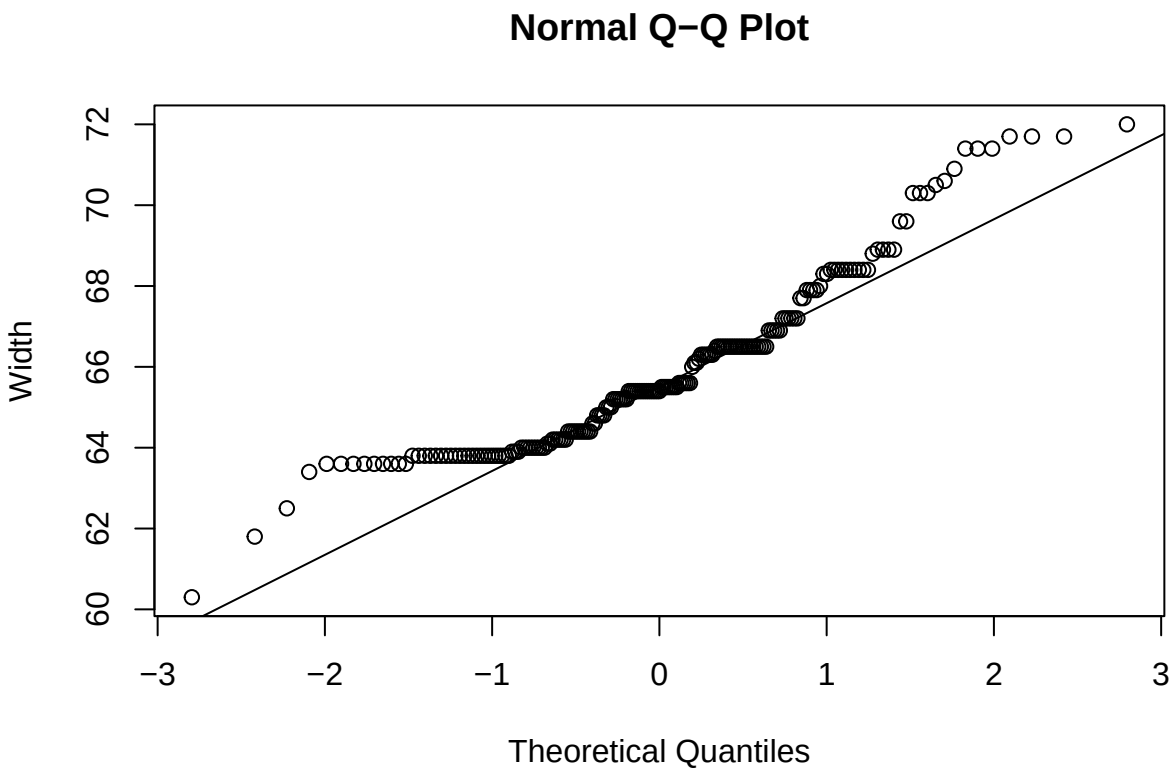


Facciamo un confronto anche di altre variabili quantitative che potrebbero invece avere una distribuzione normale

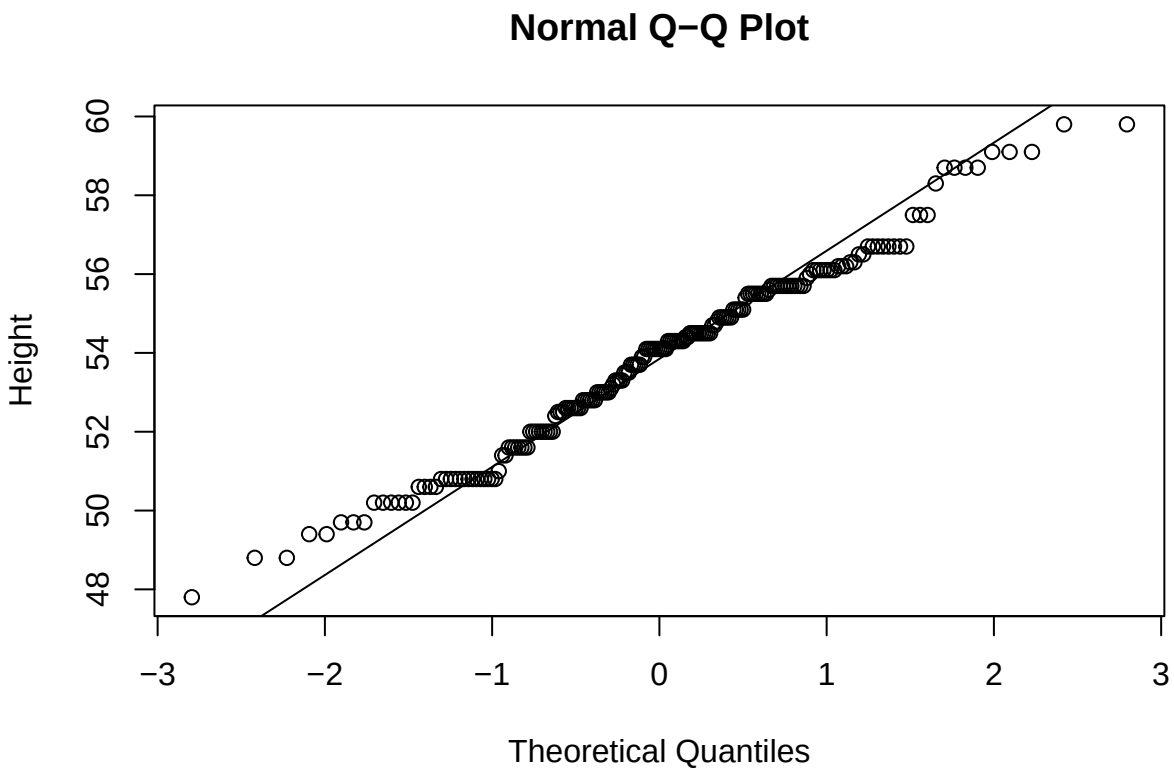
```
#lunghezza
qqnorm(mydata$Length, ylab = "Length")
qqline(mydata$Length)
```




```r
#largezza
qqnorm(mydata$Width, ylab = "Width")
qqline(mydata$Width)
```




```
#altezza
qqnorm(mydata$Height, ylab = "Height")
qqline(mydata$Height)
```



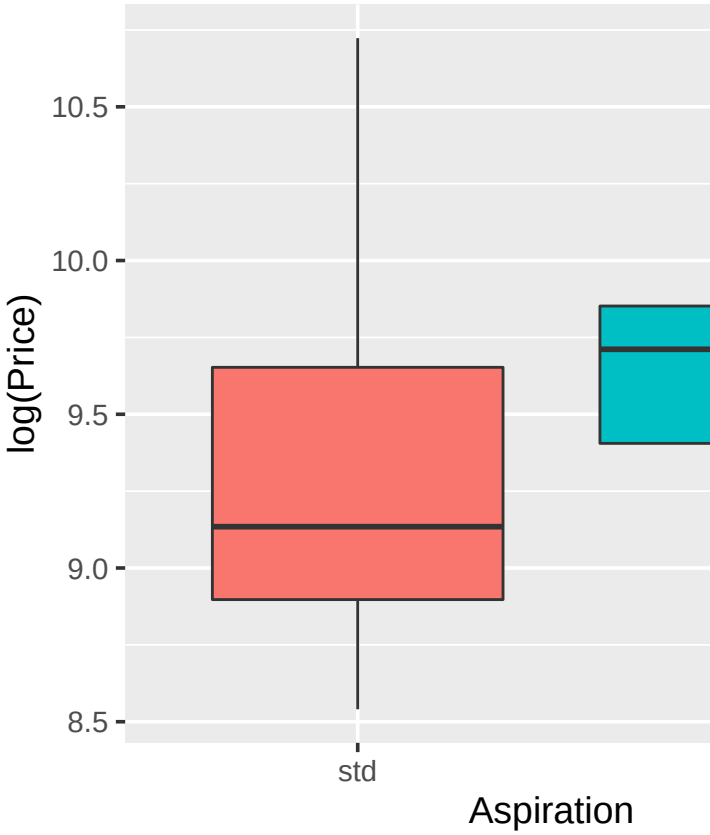
A partire dai qqplot si deduce che le variabili Length e Height hanno una tendenza normale, mentre la variabile Width no.

ANALISI BIVARIATA

Cominciamo a cercare un po di relazioni fra le variabili. Consideriamo il prezzo come VARIABILE RISPOSTA e cambiamo la VARIABILE COVARIATA di volta in volta.

confronto tra due insiemi di dati osservati:

```
ggplot(mydata, aes(x=Aspiration, y=log(Price), fill=Aspiration))+
  geom_boxplot() + theme_gray(base_size = 14)
```



PRICE vs ASPIRATION (qualitativa vs categoriale)

```
mean(log(mydata$Price[mydata$Aspiration=="std"]))
```

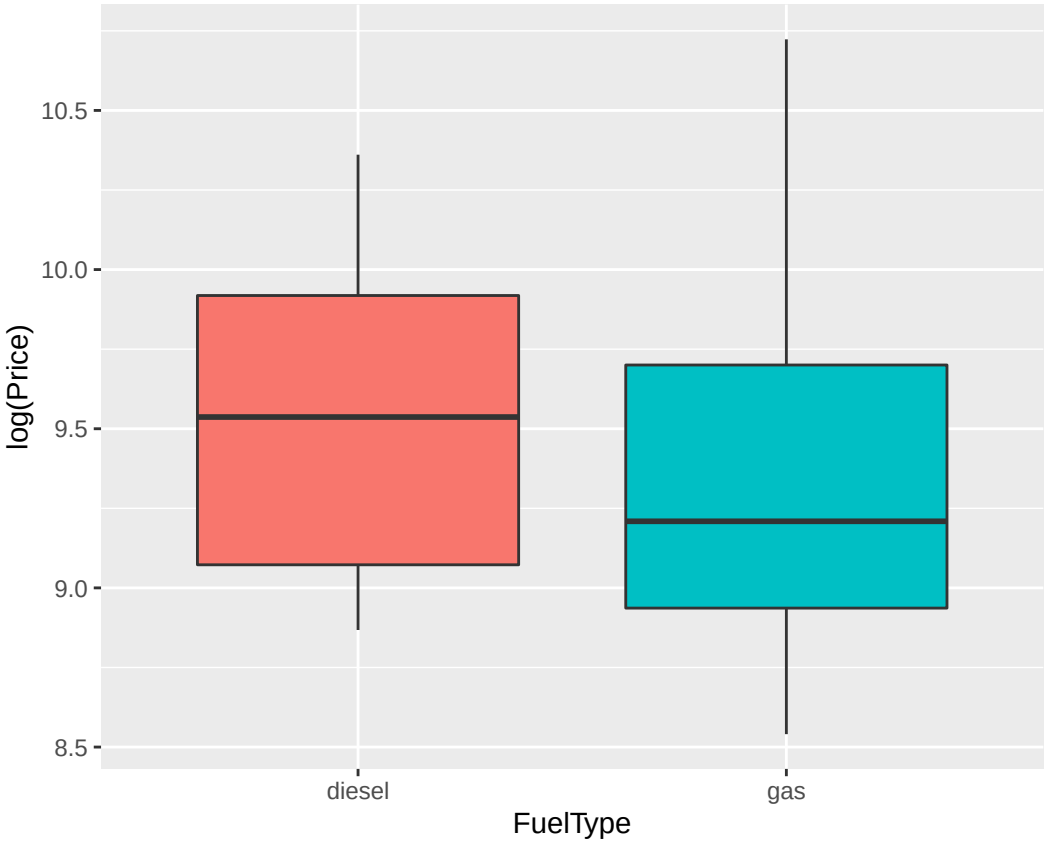
```
## [1] 9.287065
```

```
mean(log(mydata$Price[mydata$Aspiration=="turbo"]))
```

```
## [1] 9.642255
```

I boxplot mostrano che può esserci una correlazione significativa tra il tipo di aspirazione e il prezzo.

```
ggplot(mydata, aes(x=FuelType, y=log(Price), fill=FuelType))+
  geom_boxplot()
```



PRICE vs FUELTYPE

```
#i diesel costao in media un po di piu
mean(log(mydata$Price[mydata$FuelType=="diesel"]))
```

```
## [1] 9.571661
```

```
mean(log(mydata$Price[mydata$FuelType=="gas"]))
```

```
## [1] 9.327435
```

Notiamo che ancora una volta c'è una differenza di medie tra le auto a benzina e quelle a diesel. Indaghiamo sulla significatività di tale differenza.

TEST SULLE DIFFERENZE TROVATE

Facciamo un test per vedere se questa differenza è significativa 1) PRICE vs FUELTYPE

```
t.test(log(mydata$Price[mydata$FuelType=="diesel"]),
           log(mydata$Price[mydata$FuelType=="gas"]), alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  log(mydata$Price[mydata$FuelType == "diesel"]) and log(mydata$Price[mydata$FuelType == "gas"]
```

```
## t = 2.0289, df = 22.314, p-value = 0.05457
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.005213846  0.493666882
## sample estimates:
## mean of x mean of y
##  9.571661  9.327435
```

Siccome il p-value è attorno a 0.05, siamo all'interno dell'intervallo di confidenza al 95%, e quindi non abbiamo prove sufficienti per rifiutare l'ipotesi nulla che il prezzo delle auto diesel non sia significativamente diverso da quello delle macchine a benzina.

Passiamo a vedere se la differenza per tipo di aspirazione è significativa

2) PRICE vs ASPIRATION

```
t.test(log(mydata$Price[mydata$Aspiration=="std"]),
           log(mydata$Price[mydata$Aspiration=="turbo"]), alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  log(mydata$Price[mydata$Aspiration == "std"]) and log(mydata$Price[mydata$Aspiration == "turbo
## t = -4.6963, df = 65.777, p-value = 1.393e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5062036 -0.2041761
## sample estimates:
## mean of x mean of y
##  9.287065  9.642255
```

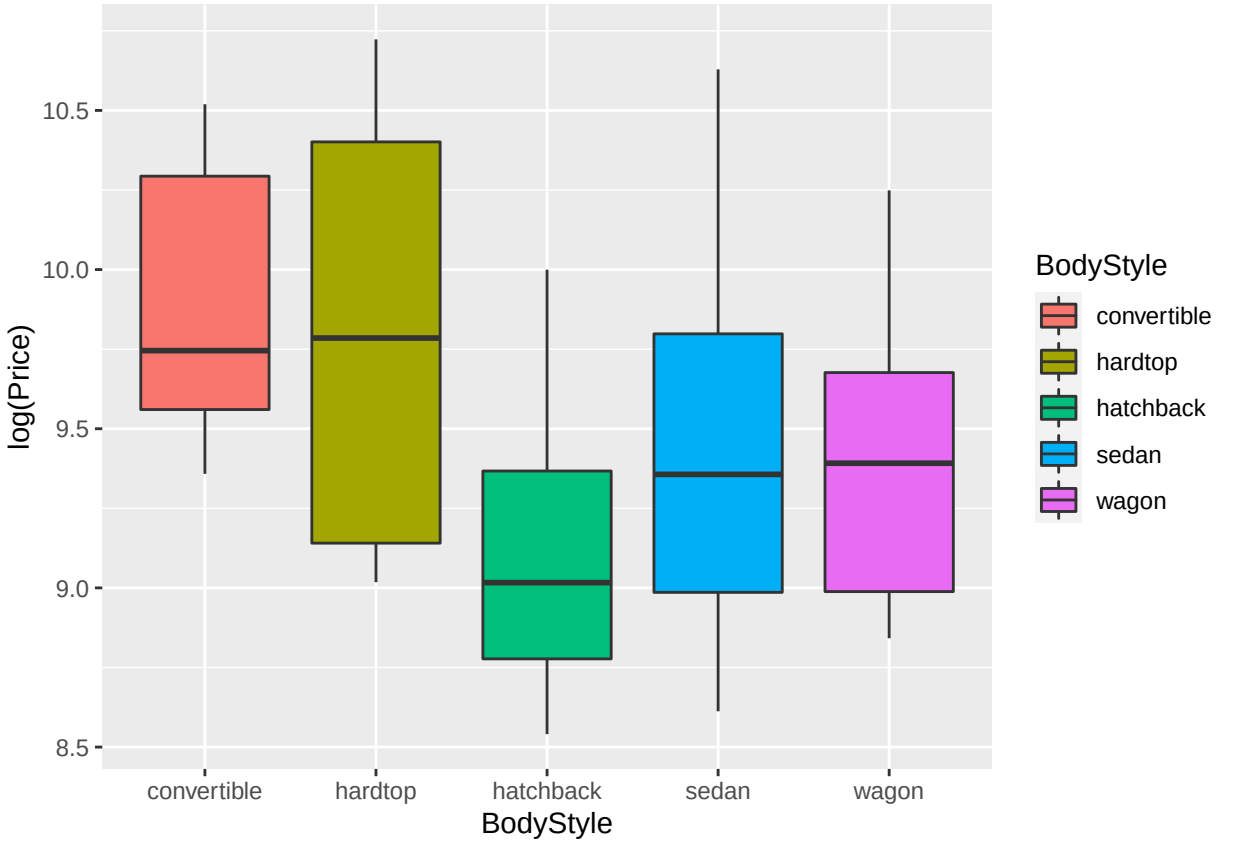
Siccome il p-value è molto minore di 0.05, abbiamo prove sufficienti per rifiutare l'ipotesi nulla che il prezzo delle auto ad aspirazione standard non è significativamente diverso da quello delle macchine ad aspirazione turbo. In altre parole, il test ci dice che le macchina Turbo sono significativamente più costose delle macchine Standard.

3) PRICE vs BODYSTYLE

```
BodyStyle
```

```
##
## convertible     hardtop   hatchback       sedan       wagon
##           6           8          63          92          24
```

```
ggplot(mydata, aes(x=BodyStyle, y=log(Price), fill=BodyStyle))+
  geom_boxplot()
```




```
#si vede che le macchine con hatcback (bagagliaio alto) sono piu economiche delle altre
```

Notando che le medie variano tra i vari stili di vetture, indaghiamo con il test ANOVA la significatività di tali differenze.

```
#ANOVA
df_aov <- aov(log(Price) ~ BodyStyle, data = mydata) #Fischer's classic ANOVA function
summary(df_aov)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## BodyStyle     4   7.93  1.9820   8.822 1.51e-06 ***
## Residuals   188  42.24  0.2247
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(df_aov)
```

```
## Call:
##    aov(formula = log(Price) ~ BodyStyle, data = mydata)
##
## Terms:
##                 BodyStyle Residuals
## Sum of Squares    7.92787  42.23654
## Deg. of Freedom         4       188
```

```
##
## Residual standard error: 0.4739857
## Estimated effects may be unbalanced
```

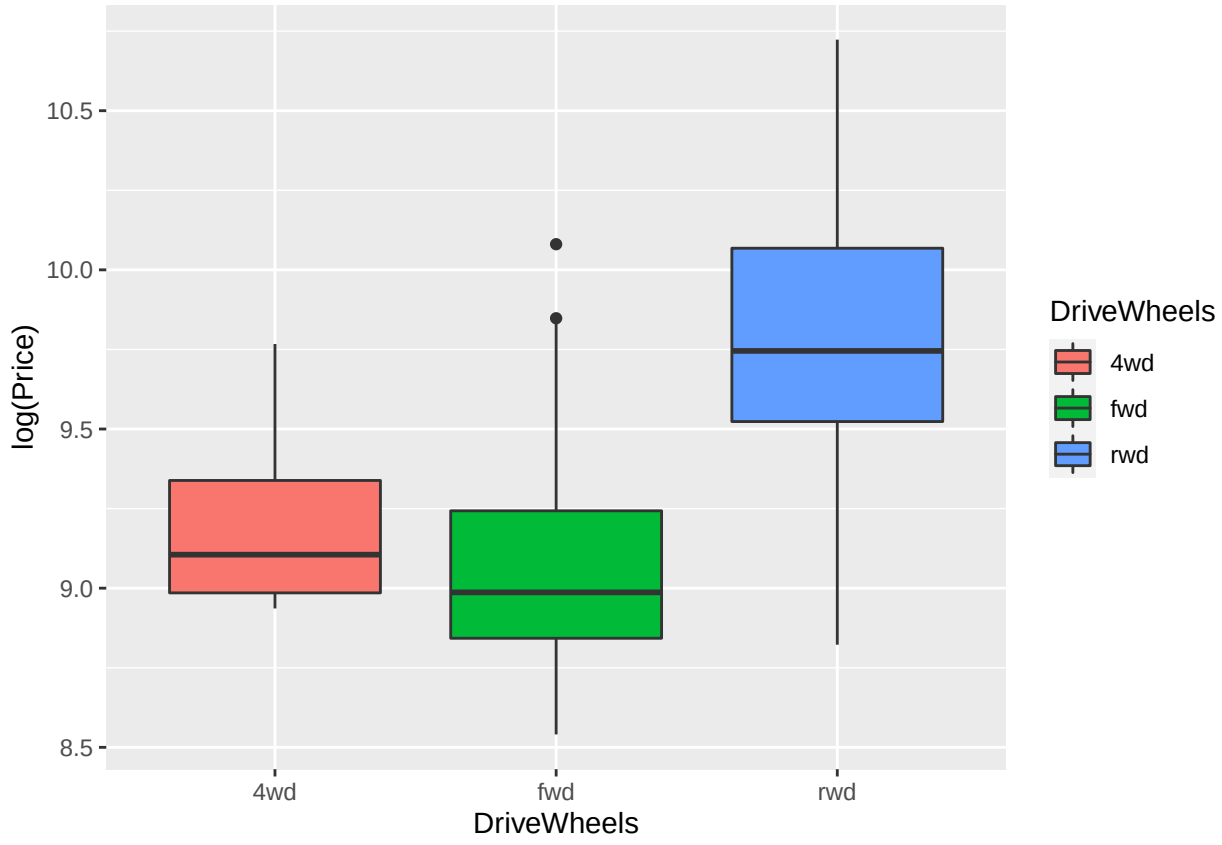
Dal test possiamo notare che c'è una differenza significativa, fra le medie dei gruppi. Dal boxplot possiamo assumere che almeno il prezzo delle auto hatcback sia sensibilmente differente dal prezzo delle altre auto.

4) PRICE vs DRIVE.WHEELS

```
DriveWheels
```

```
##
## 4wd fwd rwd
##   8 114  71
```

```
ggplot(mydata, aes(x=DriveWheels, y=log(Price), fill=DriveWheels))+
  geom_boxplot()
```




```
#ANOVA
wheel_price_aov <- aov(log(Price) ~ DriveWheels, data = mydata)
summary(wheel_price_aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## DriveWheels   2  24.12  12.060   87.98 <2e-16 ***
## Residuals   190  26.05   0.137
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(wheel_price_aov)
```

```
## Call:
##    aov(formula = log(Price) ~ DriveWheels, data = mydata)
##
## Terms:
##                 DriveWheels Residuals
## Sum of Squares     24.11978  26.04462
## Deg. of Freedom           2       190
##
## Residual standard error: 0.3702391
## Estimated effects may be unbalanced
```

Il F-value è molto piccolo e questo porta a rifiutare l'ipotesi nulla che non vi sia sostanziale variabilità nel prezzo a seconda del tipo di trazione.

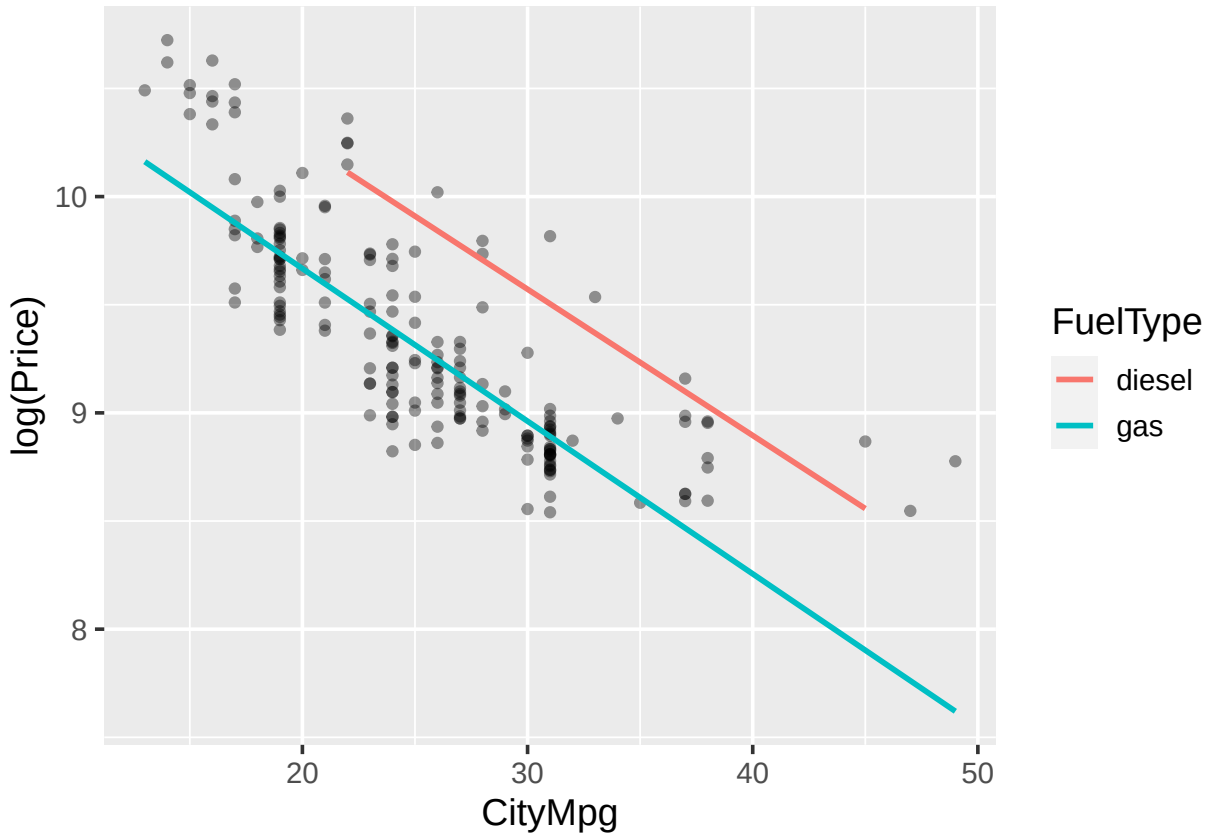
3. ANALISI MULTIVARIATA

Vogliamo visualizzare alcune delle relazioni che intercorrono fra più di due variabili del dataset.

PRICE vs CITYMPG vs FUELTYPE Vogliamo vedere la relazione che intercorre tra il **prezzo** delle auto e il loro **consumo di carburante** in città insieme alla tipologia di **carburante**.

Possiamo ipotizzare che le auto costose facciano meno chilometri con un litro, e che le machine diesel costino in media di più.

```
ggplot(mapping = aes(x=CityMpg, y=log(Price)), data=mydata) +
  geom_point( alpha=0.4) + geom_smooth(aes(colour=FuelType),method = "lm", se=F) +
  theme_gray(base_size = 14)
```

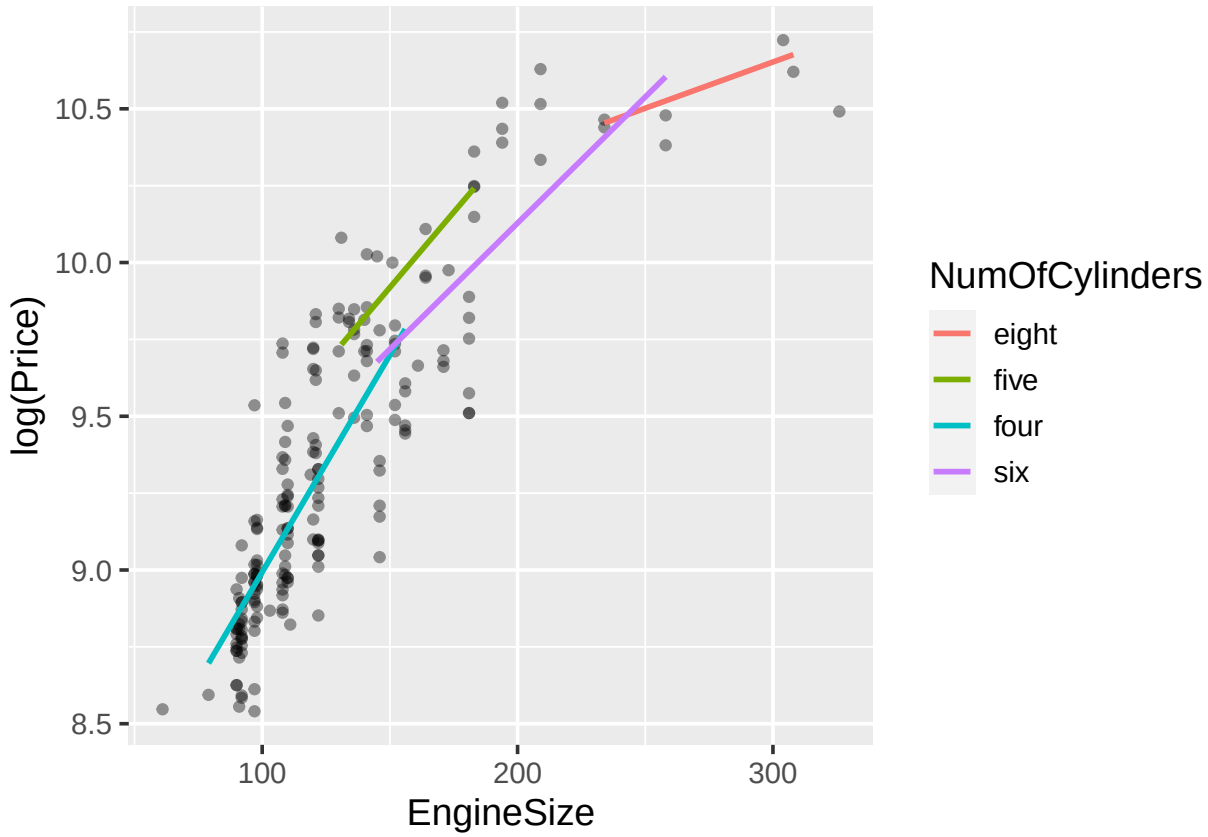


Come ipotizzavamo, esiste una correlazione negativa tra prezzo e consumo in città.

PRICE vs ENGINE.SIZE vs NUM.CILINDERS Adesso siamo interessati a vedere il collegamento tra prezzo, grandezza del motore e numero di cilindri.

Anche qui ci aspettiamo che ci sia una correlazione positiva: i motori grossi costano di più e hanno più cilindri.

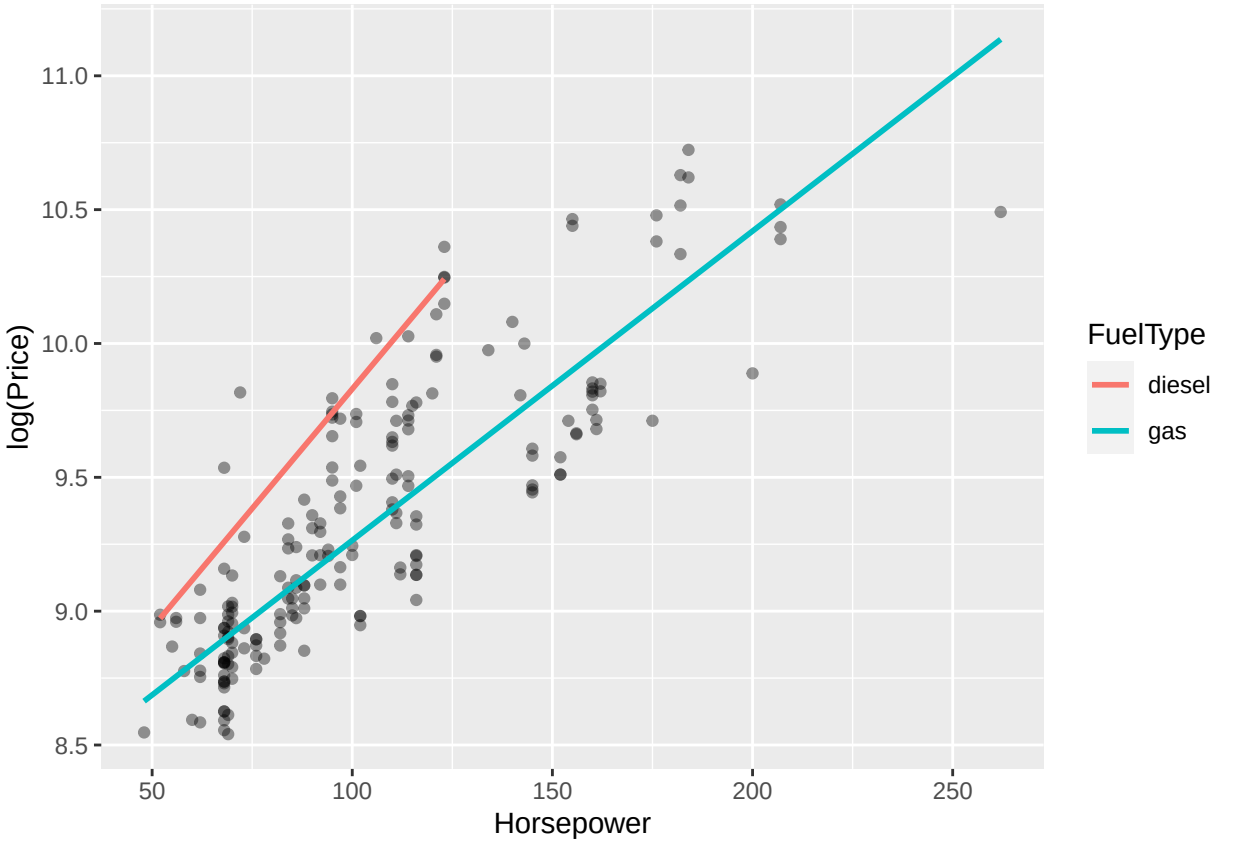
```
ggplot(mapping = aes(x=EngineSize, y=log(Price)), data=mydata) +
  geom_point( alpha=0.4) + geom_smooth(aes(colour=NumOfCylinders),method = "lm", se=F) +
   theme_gray(base_size = 14)
```



Anche qui l'ipotesi sembra essere corretta.

PRICE vs HORSEPOWER vs FUEL.TYPE Un'altra analisi interessante è vedere che relazione c'è tra **prezzo** e numero di **cavalli**. E anche qui cerchiamo di scoprire se cambia qualcosa a seconda del **carburante**.

```
# c'è una relazione interessante tra horsepower e prezzo come ci potevamo aspettare
ggplot(mapping = aes(x=Horsepower, y=log(Price)), data=mydata) +
  geom_point(alpha=0.4) + geom_smooth(aes(colour=FuelType), method = "lm", se=F)
```



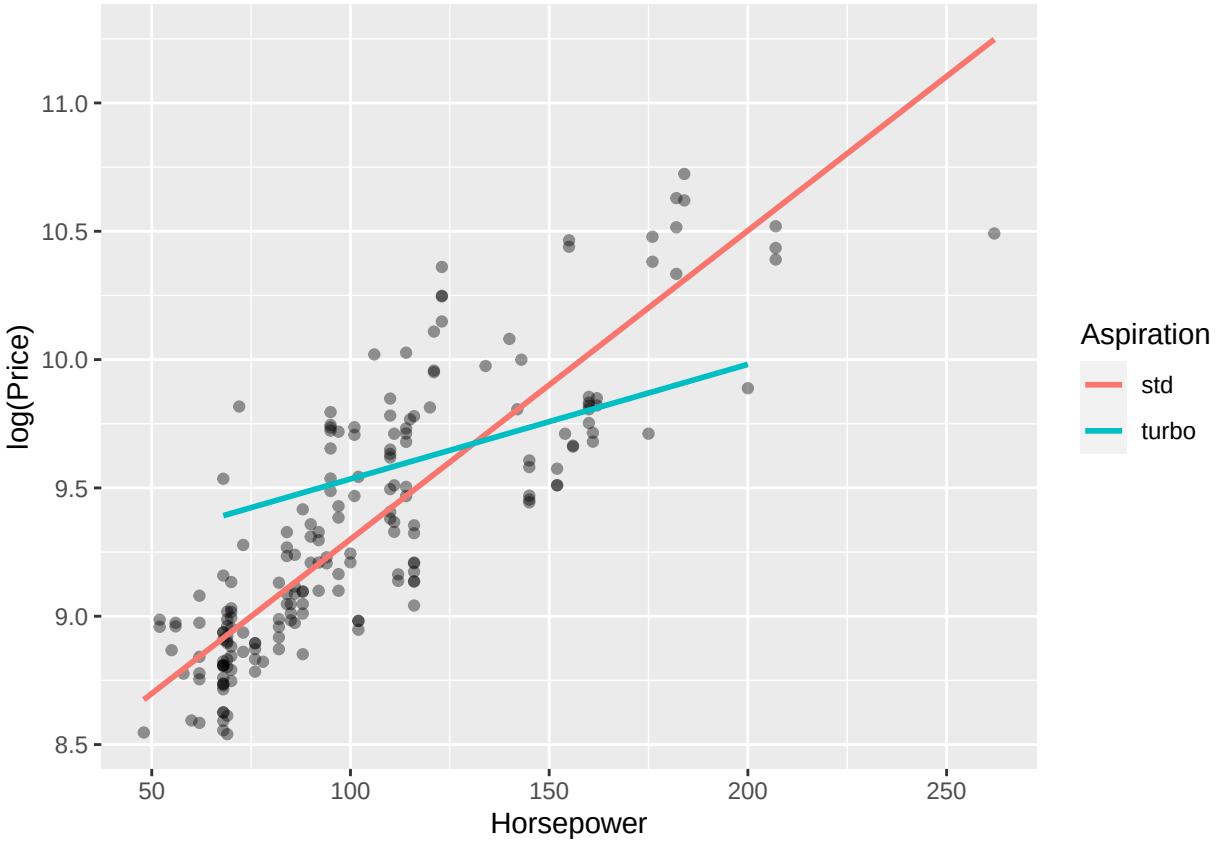
In sintesi notiamo che più cavalli abbiamo , più costa l'auto, ma quelle **diesel** all'aumentare dei cavalli sembrano essere più costose.

Insomma se voglio una macchina con tanti cavalli meglio prenderla a benzina.

PRICE vs HORSEPOWER vs ASPIRATION Infine analizziamo come varia il **prezzo** a seconda del numero di **cavalli** e il tipo di **aspirazione**.

Anche qui ci aspettiamo che il prezzo sia più alto al crescere del numero dei cavalli e che l'aspirazione turbo abbia una tendenza ad essere più costosa.

```
ggplot(mapping = aes(x=Horsepower, y=log(Price)), data=mydata) +
  geom_point(alpha=0.4) + geom_smooth(aes(colour=Aspiration), method = "lm", se=F)
```



E invece notiamo una cosa interessante: il prezzo dell'aspirazione turbo è più alto fino a circa 130 cavalli, ma poi conviene prendere una macchina turbo anzichè standard! Questo probabilmente perché con l'aspirazione turbo si risparmia qualcosa in termini di costo dell' infrastruttura del motore e prestazioni.

REGRESSIONE LINEARE

Una volta fatta un'analisi esplorativa dei dati, decidiamo di provare a creare un **modello di regressione** lineare che provi a descrivere in modo significativo i dati che abbiamo a disposizione.

Regressione lineare semplice Iniziamo con una regressione lineare semplice, prendendo come variabile covariata **Horsepower**, che sappiamo avere una correlazione alta col prezzo.

```
reg<-lm(log(Price)~Horsepower, data=mydata)
summary(reg)
```

```
##
## Call:
## lm(formula = log(Price) ~ Horsepower, data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64356 -0.18862 -0.06348  0.19537  0.81976
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.1872907  0.0589965  138.78   <2e-16 ***
## Horsepower  0.0112502  0.0005354   21.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2816 on 191 degrees of freedom
## Multiple R-squared:  0.698,  Adjusted R-squared:  0.6965
## F-statistic: 441.5 on 1 and 191 DF,  p-value: < 2.2e-16
```

Notiamo che le stime dell'intercetta e dello slope della retta di regressione sono significative dal punto statistico, e che il modello è accettabile avendo un R2 elevato.

Dopo una serie di test abbiamo deciso di migliorare il modello aggiungendo altre variabili correlate al prezzo.

```
linear <- lm(log(Price) ~ Make + Aspiration + BodyStyle + WheelBase + Length + Width +
               Height + CurbWeight + NumOfCylinders + EngineSize + PeakRpm +
               FuelType + EngineLocation, data = mydata)
summary(linear)
```

Regressione lineare

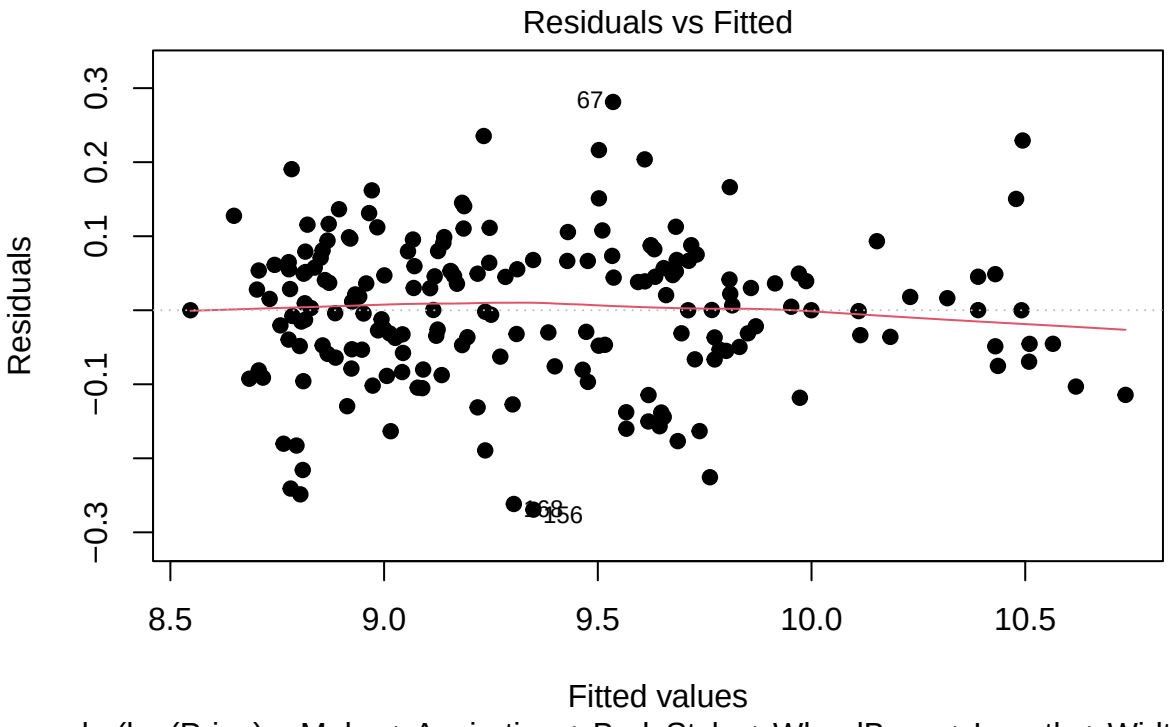
```
##
## Call:
## lm(formula = log(Price) ~ Make + Aspiration + BodyStyle + WheelBase +
##     Length + Width + Height + CurbWeight + NumOfCylinders + EngineSize +
##     PeakRpm + FuelType + EngineLocation, data = mydata)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.269327 -0.054891  0.002972  0.061271  0.281341
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           7.696e+00  8.493e-01   9.061 5.74e-16 ***
## Makeaudi              6.723e-02  1.191e-01   0.565 0.573214
## Makebmw               2.983e-01  8.938e-02   3.338 0.001061 **
## Makechevrolet        -3.259e-01  1.141e-01  -2.857 0.004877 **
## Makedodge            -3.790e-01  9.153e-02  -4.141 5.70e-05 ***
## Makehonda            -2.156e-01  9.003e-02  -2.395 0.017832 *
## Makeisuzu            -3.933e-01  1.111e-01  -3.540 0.000530 ***
## Makejaguar           -4.413e-01  1.303e-01  -3.386 0.000902 ***
## Makemazda            -1.350e-01  8.658e-02  -1.559 0.121028
## Makemercedes-benz    -1.511e-01  1.425e-01  -1.060 0.290705
## Makemercury          -2.059e-01  1.445e-01  -1.425 0.156332
## Makemitsubishi       -4.379e-01  8.630e-02  -5.074 1.11e-06 ***
## Makenissan           -2.003e-01  8.112e-02  -2.469 0.014660 *
## Makepeugot           -4.509e-01  1.059e-01  -4.257 3.61e-05 ***
## Makeplymouth         -3.982e-01  9.064e-02  -4.393 2.08e-05 ***
## Makeporsche           2.196e-01  1.468e-01   1.496 0.136638
## Makesaab              3.769e-02  9.866e-02   0.382 0.702940
```

```
## Makesubaru           -2.967e-01  8.582e-02  -3.457 0.000706 ***
## Maketoyota           -2.651e-01  7.940e-02  -3.338 0.001059 **
## Makevolkswagen       -1.375e-01  8.710e-02  -1.579 0.116374
## Makevolvo            -1.794e-01  9.622e-02  -1.865 0.064161 .
## Aspirationturbo       1.267e-01  3.053e-02   4.149 5.53e-05 ***
## BodyStylehardtop     -1.447e-01  6.943e-02  -2.085 0.038748 *
## BodyStylehatchback   -1.839e-01  6.337e-02  -2.903 0.004248 **
## BodyStylesedan       -1.195e-01  6.653e-02  -1.797 0.074341 .
## BodyStylewagon       -1.572e-01  7.470e-02  -2.105 0.036945 *
## WheelBase             2.007e-02  5.229e-03   3.838 0.000181 ***
## Length               -4.986e-03  2.955e-03  -1.687 0.093587 .
## Width                 1.049e-02  1.329e-02   0.789 0.431355
## Height               -3.548e-02  7.607e-03  -4.663 6.73e-06 ***
## CurbWeight            6.897e-04  8.051e-05   8.566 1.08e-14 ***
## NumOfCylindersfive   -1.176e-01  1.080e-01  -1.088 0.278100
## NumOfCylindersfour   -5.768e-02  1.429e-01  -0.404 0.686998
## NumOfCylinderssix    -1.035e-01  1.370e-01  -0.756 0.450855
## NumOfCylindersthree   1.845e-01  2.044e-01   0.903 0.368004
## NumOfCylinderstwelve -3.097e-02  1.930e-01  -0.160 0.872712
## EngineSize            7.446e-04  9.128e-04   0.816 0.415959
## PeakRpm               4.632e-05  2.846e-05   1.627 0.105751
## FuelTypegas           5.321e-02  4.092e-02   1.300 0.195438
## EngineLocationrear    5.460e-01  1.566e-01   3.486 0.000639 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1095 on 153 degrees of freedom
## Multiple R-squared:  0.9634, Adjusted R-squared:  0.9541
## F-statistic: 103.3 on 39 and 153 DF,  p-value: < 2.2e-16
```

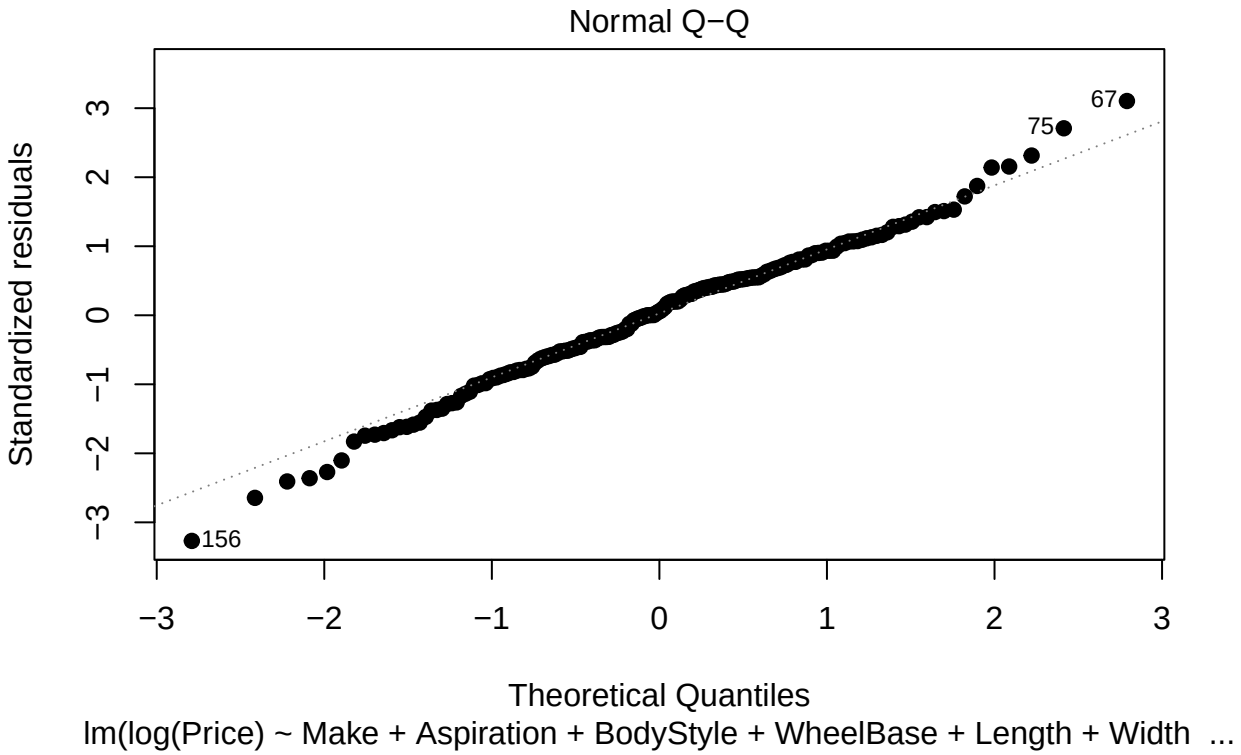
Il modello ottenuto è molto soddisfacente è possiamo ritenere che sia significativamente predittivo sulla variabile risposta price.

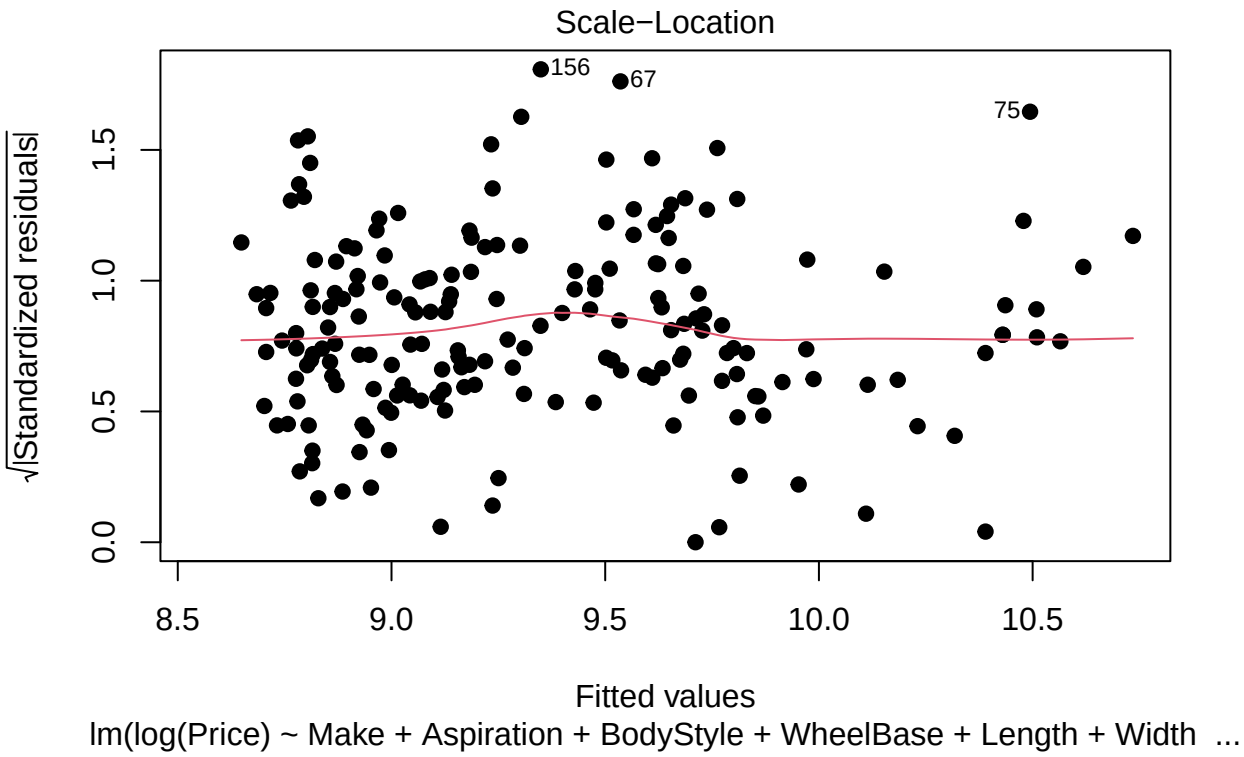
Per esserne sicuri comunque andiamo a vedere l'analisi dei residui.

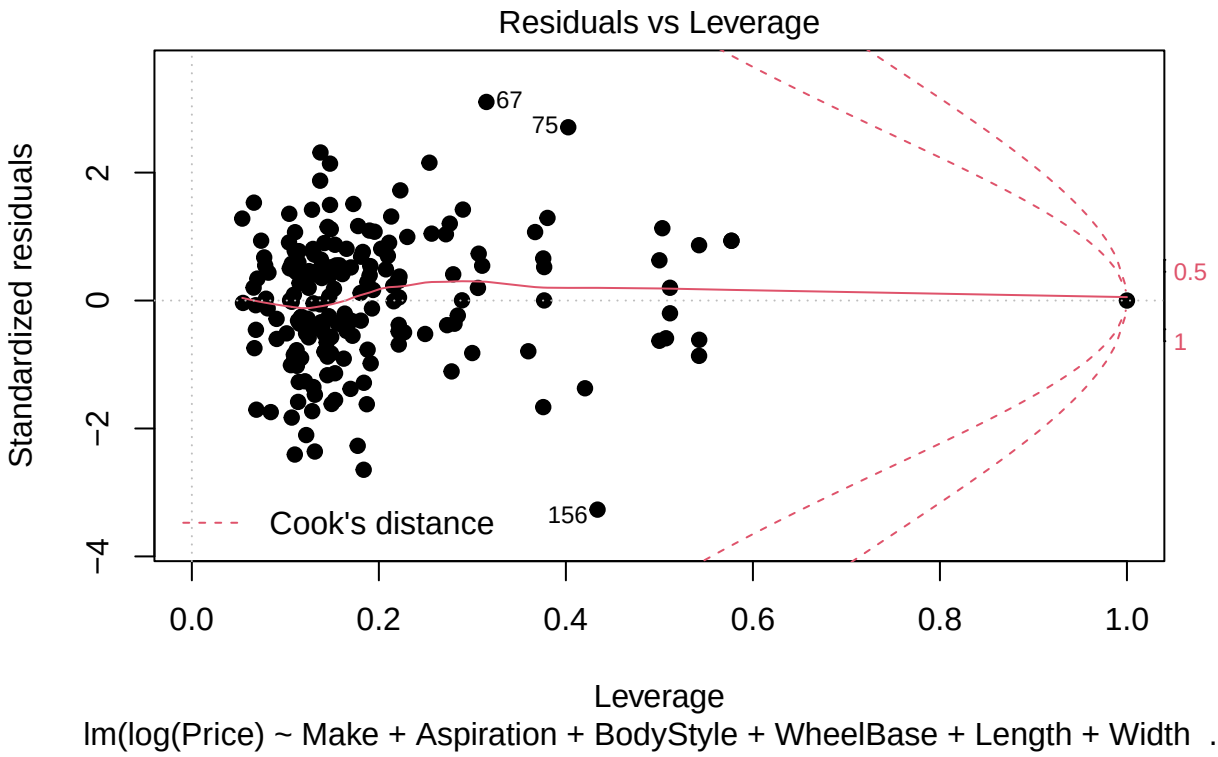
```
plot(linear,pch = 19)
```



Analisi dei residui




Scale−Location
156
67
75
√|Standardized residuals|
0.0
0.5
1.0
1.5
8.5
9.0
9.5
10.0
10.5
Fitted values
lm(log(Price) ~ Make + Aspiration + BodyStyle + WheelBase + Length + Width ...



Cerchiamo di capire se i residui soddisfano le proprietà di omschedasticità:

1) Il grafico tra i **fitted values** (il prezzo) e i **residui** ha una retta di regressione che è quasi **orizzontale**. Idealmente vogliamo che la distribuzione dei residui non cambi rispetto ai fitted value (B0=0)

2) Sembra inoltre che i residui si distribuiscano come una **normale** giudicando il grafico quantile quantile, cioè stanno piu o meno sulla bisettrice.

3) grafico **fitted values** e **residui standardizzati**, anche qui la retta di regressione è orizzontale e vediamo che praticamente tutti i dati dati rientrano nell'intervallo

$$-1.5, +1, 5$$

della gaussiana a parte qualche piccolo outlier.

4) Nell'ultimo grafico vediamo che gli **outliers** stanno dentro alla distanza di Cook. Quindi la loro leva non crea grossi problemi al modello, e possiamo ritenerci soddisfatti.

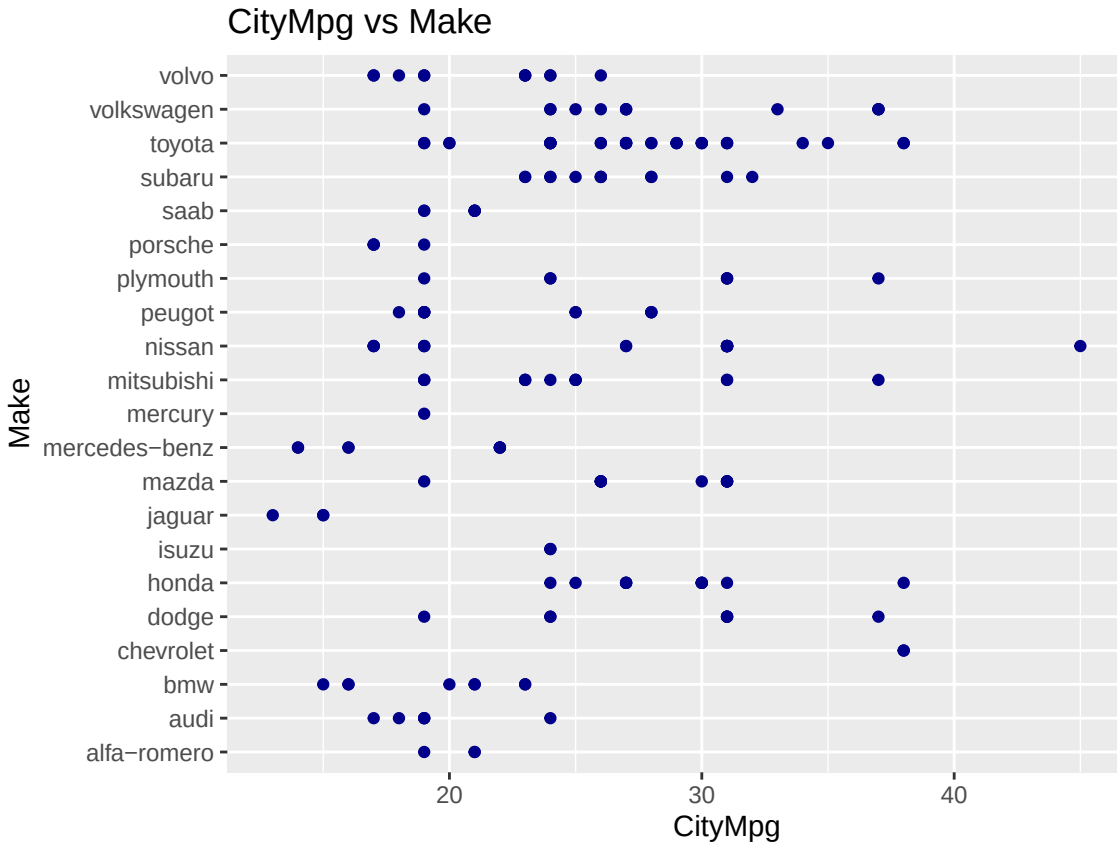
CLUSTERING

Vogliamo individuare dei sottogruppi di osservazioni che siano omogenee secondo un determinato criterio.

In particolare vogliamo vedere come possiamo suddividere il dataset in base al **prezzo** dell'auto e **consumo in città**, che potrebbe essere una domanda interessante che si potrebbe porre un compratore.

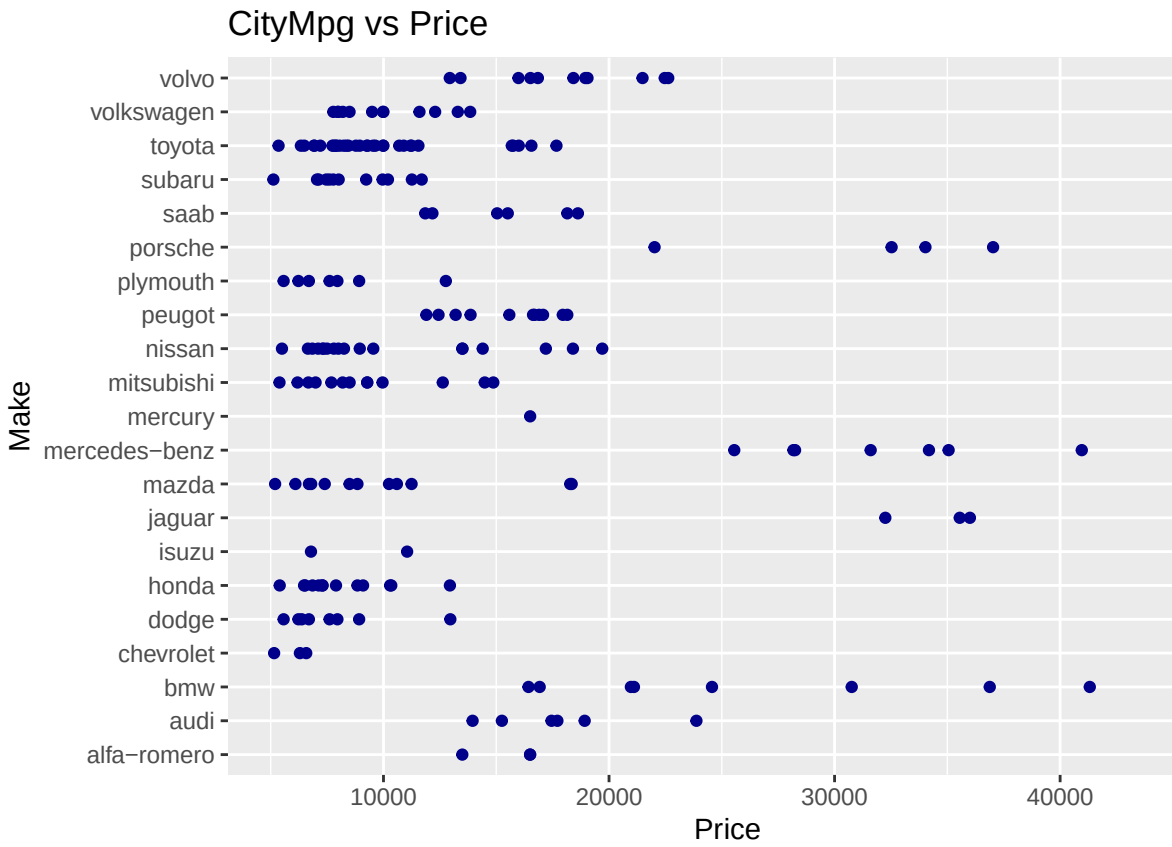
Visualizziamo innanzitutto la distribuzione dei dati e poi cerchiamo una serie di gruppi simili tra loro.

```
ggplot(data = mydata, aes(x=CityMpg,y=Make))+ geom_point(colour="darkblue", fill="lightblue") + ggtitle
```



CITYMPG vs MAKE

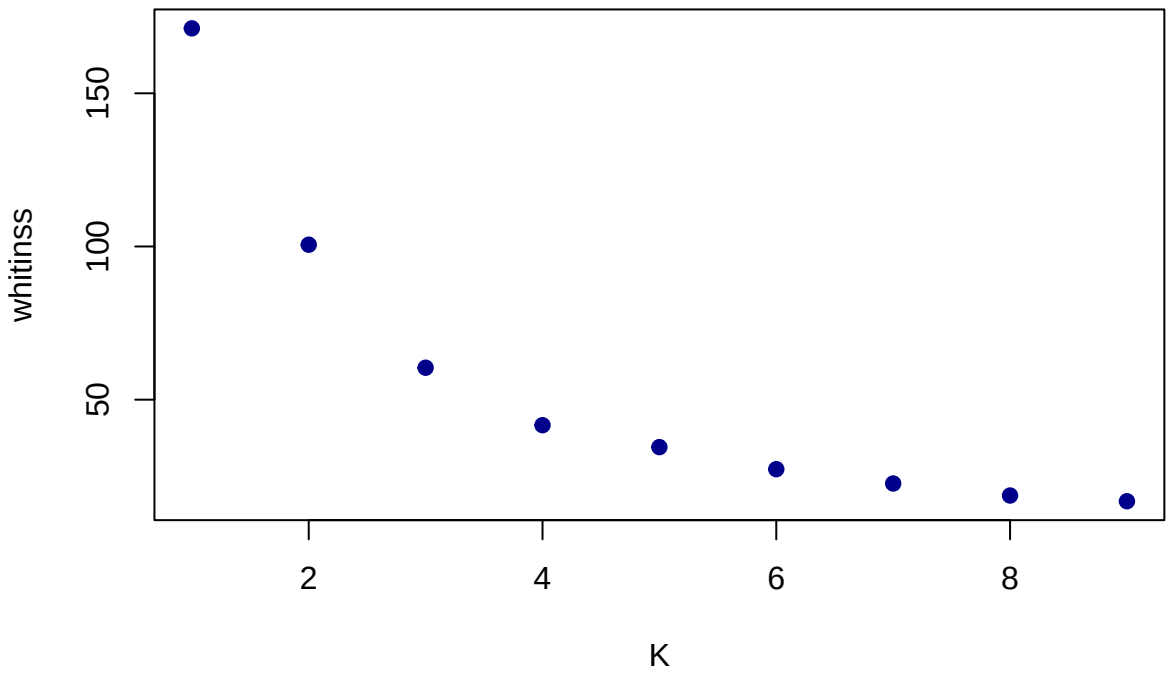
```
ggplot(data = mydata, aes(x=Price,y=Make))+ geom_point(colour="darkblue", fill="lightblue") + ggtitle("
```



PRICE vs MAKE

Decidiamo di utilizzare il metodo delle **K-medie**, cercando di trovare un'indicazione del K migliore da utilizzare come riferimento per il metodo dei **medoidi**.

```
df0<- mydata[, c("Make","CityMpg","Price")]
df1<- mydata[, c("CityMpg","Price")]
crit<-0
for (i in 2:10 ) {
  set.seed(7)
  mydatagroup<- kmeans(scale(df1),i, nstart = 10)
  crit[i-1]<-mydatagroup$tot.withinss
}
plot(1:9,crit, pch=19, xlab = "K", ylab = "whitinss", col="darkblue")
```

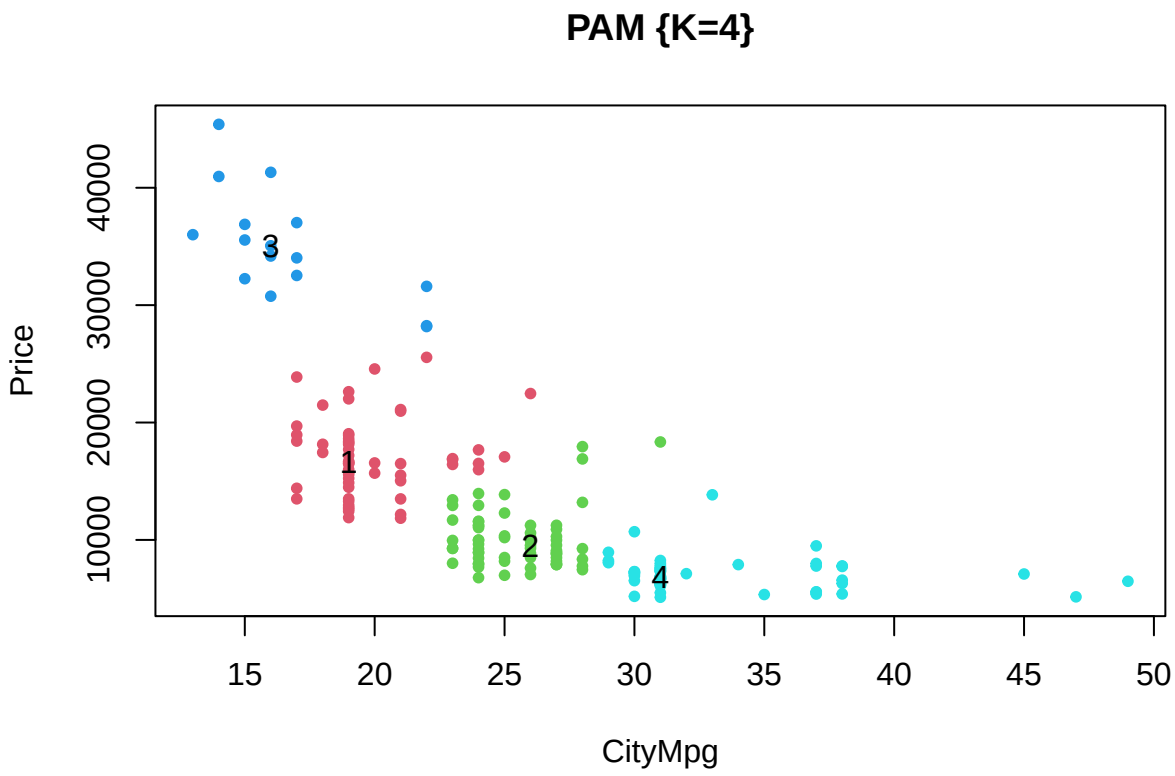


Abbiamo scelto di utilizzare K=4 per il metodo dei medoidi.

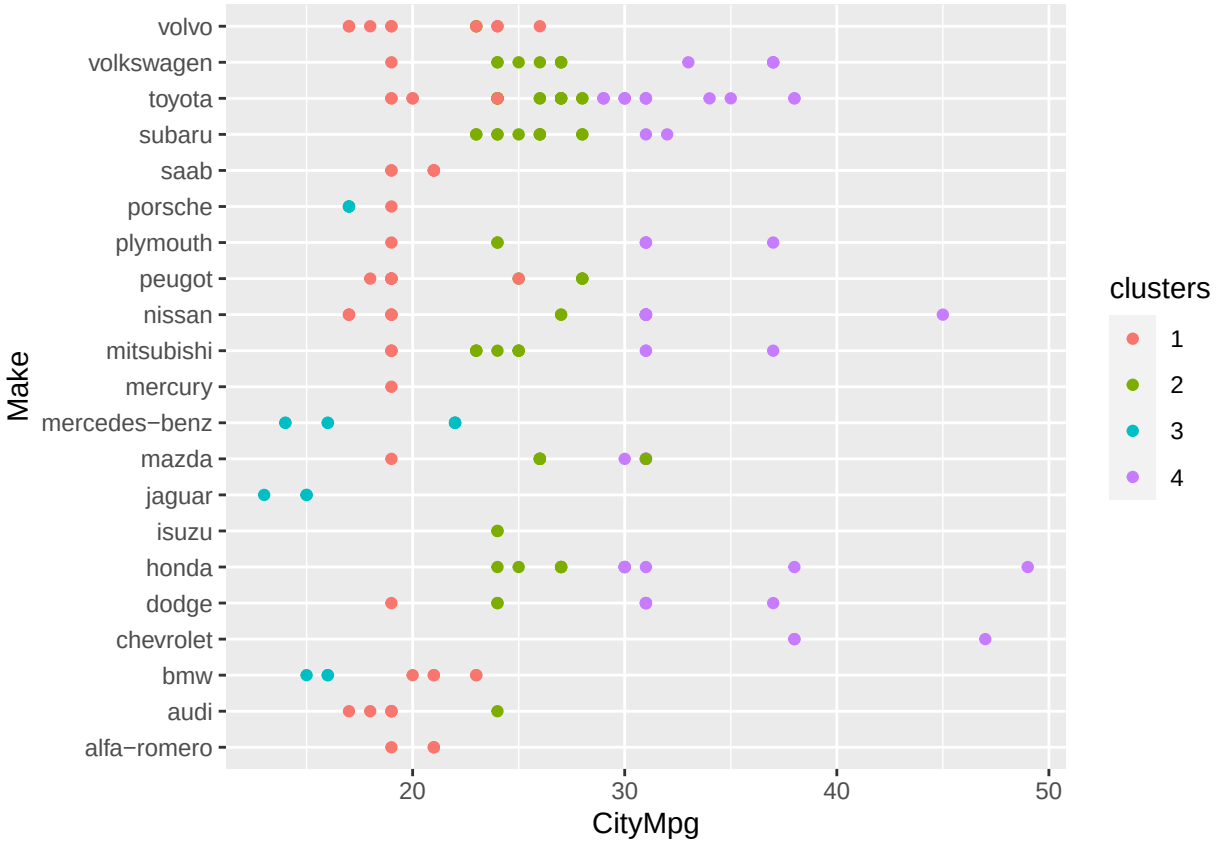
```
pam.out<- pam(df1,4, metric = "euclidean", stand = T)
pam.out$medoids
```

```
##     CityMpg Price
## 116      19 16630
## 167      26  9538
## 73       16 35056
## 93       31  6849
```

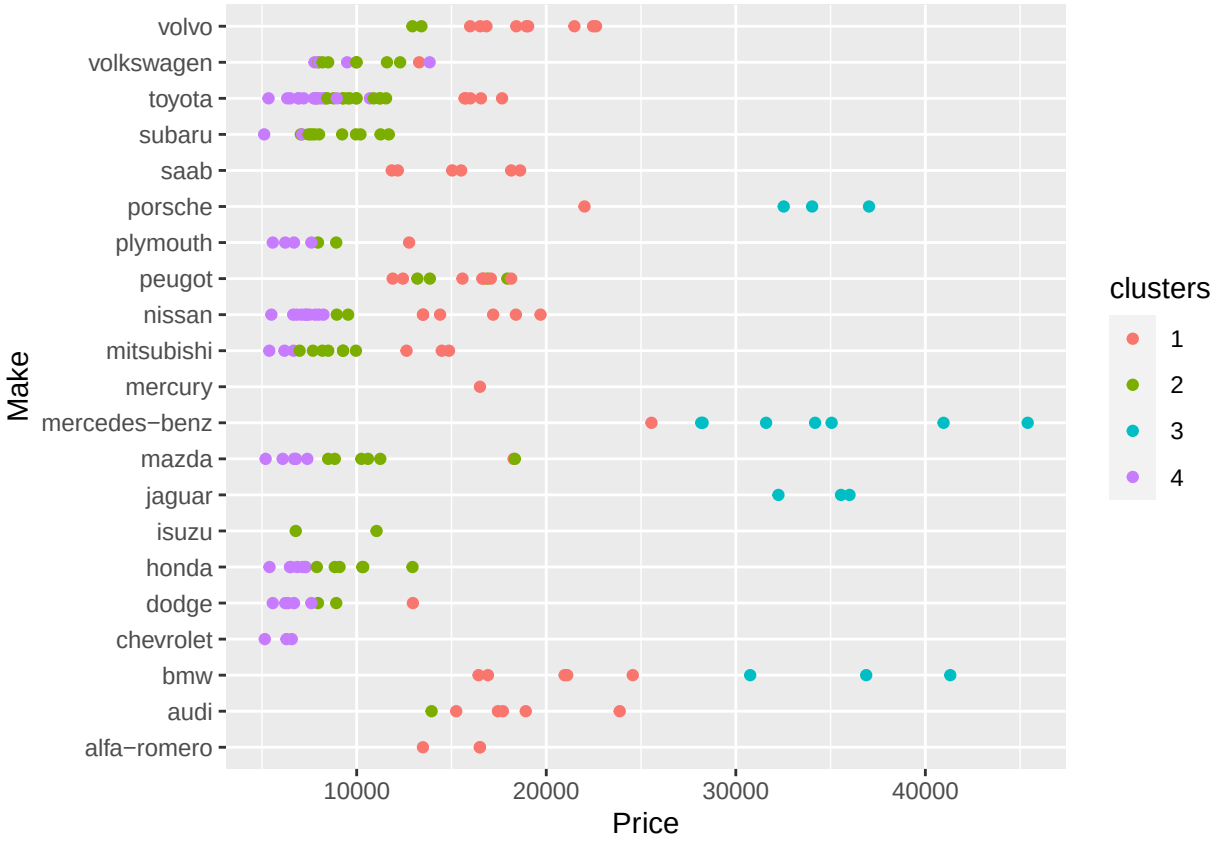
```
plot(df1, col=(pam.out$cluster+1), main="PAM {K=4}",pch=20)
points(pam.out$medoids, pch=as.character(pam.out$cluster[pam.out$id.med]))
```




```
clusters<-as.factor(pam.out$cluster)
ggplot(data = df0, aes(x=CityMpg,y=Make))+ geom_point(aes(colour=clusters))
```




```
ggplot(data = df0, aes(x=Price,y=Make))+ geom_point(aes(colour=clusters))
```




```
plot(pam.out, which=2, main="", col = "green")
```

L'avarage silhouette è soddisfacente per aver scelto K=4.

CONCLUSIONI

- Abbiamo trovato relazioni interessanti tra le variabili, e abbiamo testato la significatività di queste relazioni.
- Abbiamo notato che LogPrice può essere interpretato come una normale.
- Abbiamo scovato informazioni interessanti tra le variabili correlate al prezzo.
- Abbiamo creato un modello di regressione lineare che possa predire, in base ai dati, l'andamento del prezzo.
- Abbiamo creato un suddivisione del dataset in base al consumo di carburante in città, aiutando un possibile compratore nella scelta dell'acquisto dell'auto.