Deliberable 1

Lab 1 - Data Preparation

Othman Benmoussa & Eloi Cruz

March 27, 2022

Contents

1 Data Description: 100,000 UK Used Car Data set

This data dictionary describes data (https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes) - A sample of 5000 used sold cars has been randomly selected from Mercedes, BMW, Volkwagen and Audi manufacturers. So, firstly you have to combine used car from the 4 manufacturers into 1 dataframe.

The cars with engine size 0 are in fact electric cars, nevertheless Mercedes C class, and other given cars are not electric cars,so data imputation is required.

1.1 Variables description

- manufacturer: represents the company that manufactures the car (Factor: Audi, BMW, Mercedes or Volkswagen)
- model: the exact model of the car represented Car
- year: year of registration
- price: price in £
- transmission: type of gearbox
- mileage: distance already used by the car
- fuelType: fuel consumed by the car engine
- tax: road tax
- mpg: Consumption in miles per gallon

- engineSize: size in liters

2 Environment preparation

2.1 Load Required Packages: to be increased over the course

```
# Load Required Packages: to be increased over the course
options(contrasts=c("contr.treatment","contr.treatment"))

requiredPackages <- c("effects","FactoMineR","car", "factoextra","RColorBrewer","ggplot2","dplyr","ggmap

#use this function to check if each package is on the local machine
#if a package is installed, it will be loaded
#if any are not, the missing package(s) will be installed and loaded
package.check <- lapply(requiredPackages, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})
```

2.2 Cretae dataset

A random sample of 5000 cars is obtained from the original datasets audi, bmw,mercedes and VW. This will be the start point of the project and the data that we will be analized.

```
# Clear plots
if(!is.null(dev.list())) dev.off()

# Clean workspace
rm(list=ls())

# setwd("/Users/othmanbenmoussa/Desktop/FIB/ADEI/LAB0")

setwd("C:/Users/Eloi/Documents/ADEI/ADEI/Lab0") #Set working directory



# Lecture of DataFrames:
df1 <- read.table("audi.csv",header=T, sep=",")
df1$manufacturer <- "Audi"
df2 <- read.table("bmw.csv",header=T, sep=",")
df2$manufacturer <- "BMW"
df3 <- read.table("merc.csv",header=T, sep=",")
df3$manufacturer <- "Mercedes"
df4 <- read.table("vw.csv",header=T, sep=",")
df4$manufacturer <- "VW"
```

```r
# Union by row:
df <- rbind(df1,df2,df3,df4)

### Use birthday of 1 member of the group as random seed:
set.seed(11041998)
# Random selection of x registers:
sam<-as.vector(sort(sample(1:nrow(df),5000)))
df<-df[sam,]  # Subset of rows _ It will be my sample
rownames(df) <- 1:nrow(df)

#Remove original datasets
rm(df1)
rm(df2)
rm(df3)
rm(df4)

#Keep information in an .Rdata file:
save(list=c("df"),file="MostraCotxesLab1.RData")
```

2.3 Definition of useful functions

```r
# Mout <- which((df$tax < var_out$mouti)|(df$tax > var_out$mouts))

# Some useful functions
calcQ <- function(x) {
  s.x <- summary(x)
  iqr<-s.x[5]-s.x[2]
  list(souti=s.x[2]-3*iqr, mouti=s.x[2]-1.5*iqr, min=s.x[1], q1=s.x[2], q2=s.x[3],
       q3=s.x[5], max=s.x[6], mouts=s.x[5]+1.5*iqr, souts=s.x[5]+3*iqr ) }

countNA <- function(x) {
  mis_x <- NULL
  for (j in 1:ncol(x)) {mis_x[j] <- sum(is.na(x[,j])) }
  mis_x <- as.data.frame(mis_x)
  rownames(mis_x) <- names(x)
  mis_i <- rep(0,nrow(x))
  for (j in 1:ncol(x)) {mis_i <- mis_i + as.numeric(is.na(x[,j])) }
  list(mis_col=mis_x,mis_ind=mis_i) }

countX <- function(x,X) {
  n_x <- NULL
  for (j in 1:ncol(x)) {n_x[j] <- sum(x[,j]==X) }
  n_x <- as.data.frame(n_x)
  rownames(n_x) <- names(x)
  nx_i <- rep(0,nrow(x))
  for (j in 1:ncol(x)) {nx_i <- nx_i + as.numeric(x[,j]==X) }
  list(nx_col=n_x,nx_ind=nx_i) }

# CalcQ function application over price variable
list_price <- calcQ(df$price)
```

3 Univariate Descriptive Analysis, Factor, level coding

First of all we will start with the univariate descriptive analysis. This means that we will analyse all the variables one by one to understand the dataset in the most accurate way. In the next figures we can see the original data. We will analyse and describe it in more detail in the next sections. Then we will codify properly factors and remove non-informative variables

Data created summary:

```r
summary(df)
```

```
##     model                year          price        transmission      
##  Length:5000        Min.   :1999   Min.   :   899   Length:5000       
##  Class :character   1st Qu.:2016   1st Qu.: 13991   Class :character  
##  Mode  :character   Median :2017   Median : 19498   Mode  :character  
##                     Mean   :2017   Mean   : 21459                     
##                     3rd Qu.:2019   3rd Qu.: 26299                     
##                     Max.   :2020   Max.   :135124                     
##     mileage         fuelType              tax             mpg        
##  Min.   :     1   Length:5000        Min.   :  0.0   Min.   :  1.10  
##  1st Qu.:  5758   Class :character   1st Qu.:125.0   1st Qu.: 45.60  
##  Median : 16144   Mode  :character   Median :145.0   Median : 53.30  
##  Mean   : 22775                      Mean   :122.9   Mean   : 54.62  
##  3rd Qu.: 33187                      3rd Qu.:145.0   3rd Qu.: 61.40  
##  Max.   :214000                      Max.   :580.0   Max.   :470.80  
##    engineSize    manufacturer      
##  Min.   :0.000   Length:5000       
##  1st Qu.:1.500   Class :character  
##  Median :2.000   Mode  :character  
##  Mean   :1.895                     
##  3rd Qu.:2.000                     
##  Max.   :6.600                     
```

3.1 Description of the non numerical variables

There are 4 non numerical variables that we will convert into factors: model, transmission, fueltype and manufacturer.

3.1.1 Model

```r
df$model<-factor(paste0(df$manufacturer,"-",df$model))
```

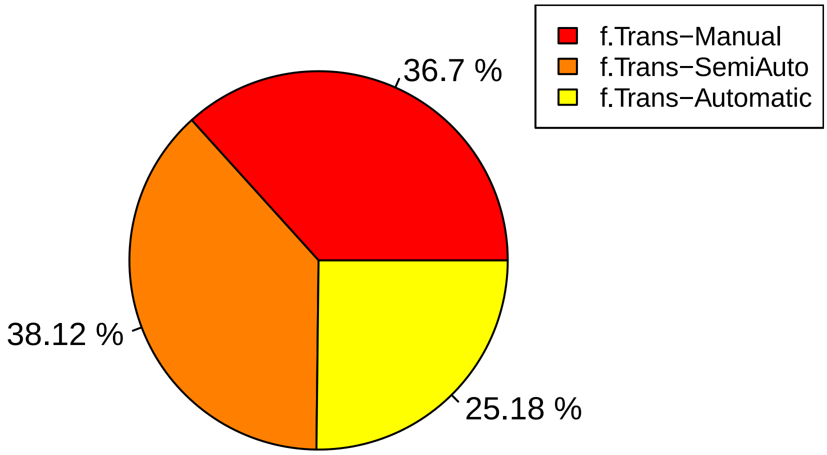
We can see that the dataset contains cars of 89 different models from the 4 different manufacturers.

3.1.2 Transmission

```r
df$transmission <- factor( df$transmission, levels = c("Manual","Semi-Auto","Automatic"),labels = paste(
# Pie
piepercent<-round(100*(table(df$transmission)/nrow(df)),dig=2); piepercent
```

```
## 
##    f.Trans-Manual  f.Trans-SemiAuto f.Trans-Automatic 
##             36.70             38.12             25.18 
```

```r
pie(table(df$transmission),col=heat.colors(3),labels=paste(piepercent,"%"))
legend("topright", levels(df$transmission), cex = 0.8, fill = heat.colors(3))
```




```
#table
table(df$transmission)
```

```
##
##    f.Trans-Manual  f.Trans-SemiAuto f.Trans-Automatic
##              1835              1906              1259
```

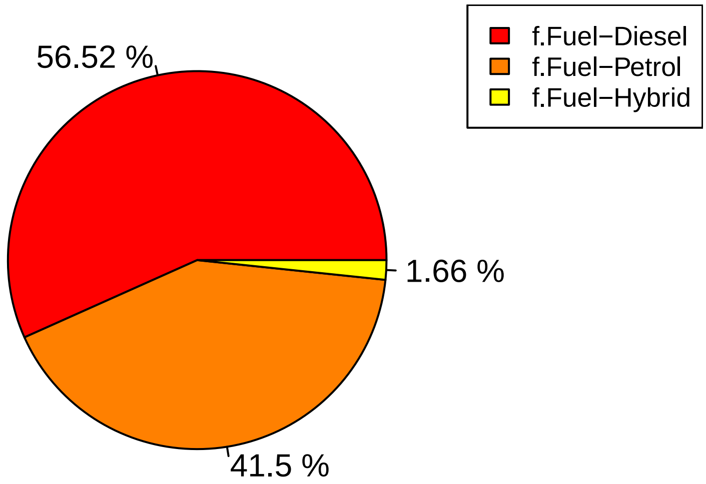
We can see that the sample contains more or less the same number of Manual and semi-auto individuals. Otherwise the number of automatic cars is a little lower.

3.1.3 Fuel type

```
df$fuelType <- factor(df$fuelType)
df$fuelType <- factor( df$fuelType, levels = c("Diesel","Petrol","Hybrid"), labels = paste0("f.Fuel-",c(
# Pie
piepercent<-round(100*(table(df$fuelType)/nrow(df)),dig=2); piepercent
```

```
##
## f.Fuel-Diesel f.Fuel-Petrol f.Fuel-Hybrid
##         56.52         41.50          1.66
```

```
pie(table(df$fuelType),col=heat.colors(3),labels=paste(piepercent,"%"))
legend("topright", levels(df$fuelType), cex = 0.8, fill = heat.colors(3))
```




```
#Table
table(df$fuelType)
```

```
##
## f.Fuel-Diesel f.Fuel-Petrol f.Fuel-Hybrid
##          2826          2075            83
```

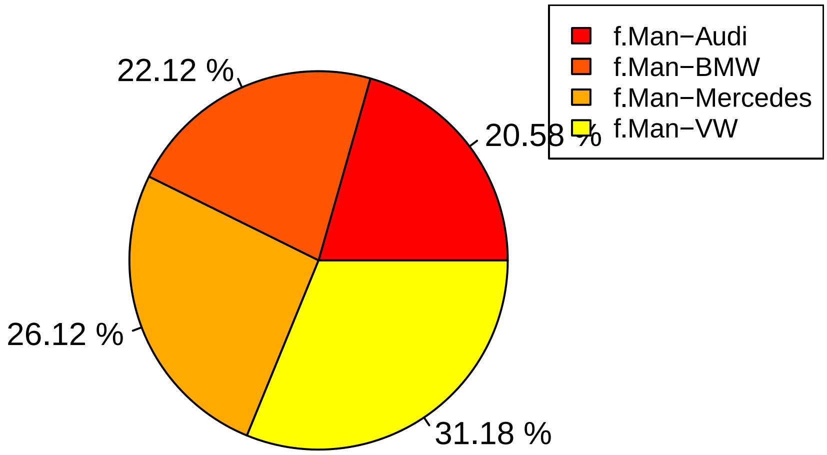
In that case we can see that most common fuel type for the cars of the dataset is Diesel (57%). The number of cars with a Petrol engine is representative too (42%). Otherwise the number of cars with a Hybrid engine is very little (2%).

3.1.4 Manufacturer

```
df$manufacturer <- factor(paste0("f.Man-",df$manufacturer))
# Pie
piepercent<-round(100*(table(df$manufacturer)/nrow(df)),dig=2); piepercent
```

```
##
##     f.Man-Audi      f.Man-BMW f.Man-Mercedes       f.Man-VW
##          20.58          22.12          26.12          31.18
```

```
pie(table(df$manufacturer),col=heat.colors(5),labels=paste(piepercent,"%"))
legend("topright", levels(df$manufacturer), cex = 0.8, fill = heat.colors(5))
```




```
#Table
table(df$fuelType)
```

```
##
## f.Fuel-Diesel f.Fuel-Petrol f.Fuel-Hybrid
##          2826          2075            83
```

As we choose the cars randomly the repartition between manufacturers is very equal. In one hand, The manufacturer that has less rows is audi with a 20% of the samples. In the other hand, the manufacturer that contains most rows is VW with a 30% of the samples.

3.1.5 Binary factor is Audi: Yes, No

We now create the binary target for the cars that are of the audi manufacturer for the further analysis.

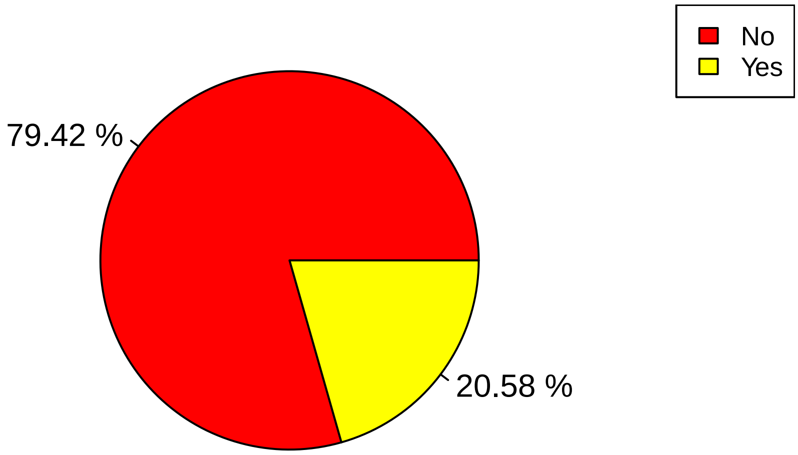
```
df$Audi<-ifelse(df$manufacturer == "f.Man-Audi",1,0)
df$Audi<-factor(df$Audi,labels=c("No","Yes"))
summary(df$Audi)
```

```
##   No  Yes
## 3971 1029
```

```
# Pie
piepercent<-round(100*(table(df$Audi)/nrow(df)),dig=2); piepercent
```

```
##
##    No   Yes
## 79.42 20.58
```

```
pie(table(df$Audi),col=heat.colors(2),labels=paste(piepercent,"%"))
legend("topright", levels(df$Audi), cex = 0.8, fill = heat.colors(2))
```



3.2 Description of numeric variables that represent qualitative concepts

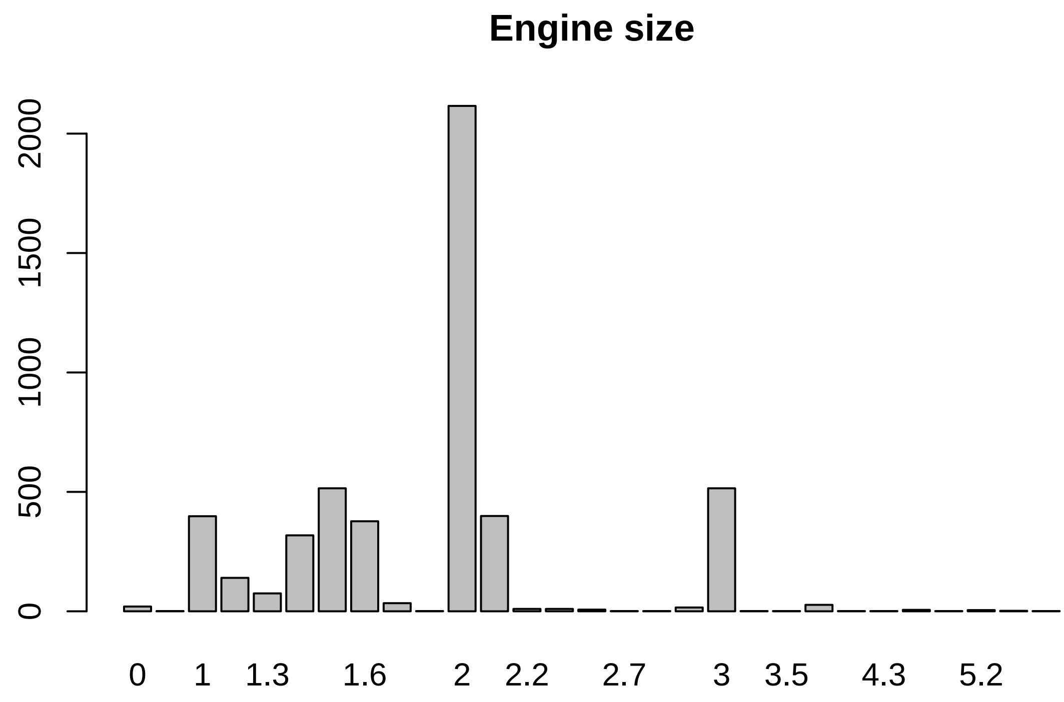
There are 4 Original numeric variables corresponding to qualitative concepts. We will describe them but we will not factorize them yet because first we want to treat all the errors, and out layers that they contain.

3.2.1 Enigine Size

```
summary(df$engineSize)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.500   2.000   1.895   2.000   6.600
```

```
barplot(table(df$engineSize), main="Engine size")
```



In first place we can find engine size. It is a numerical variable that represents a finite number of different engine sizes. For our analysis it is not very interesting to know exact size of an engine. For this reason we will group all size in 3 different categories. Category "Petit" = (0, 2), "Mitjà" = [2, 3) and "Gran" [3, infinite]. We will do this factorized process one we have treated errors and out liers

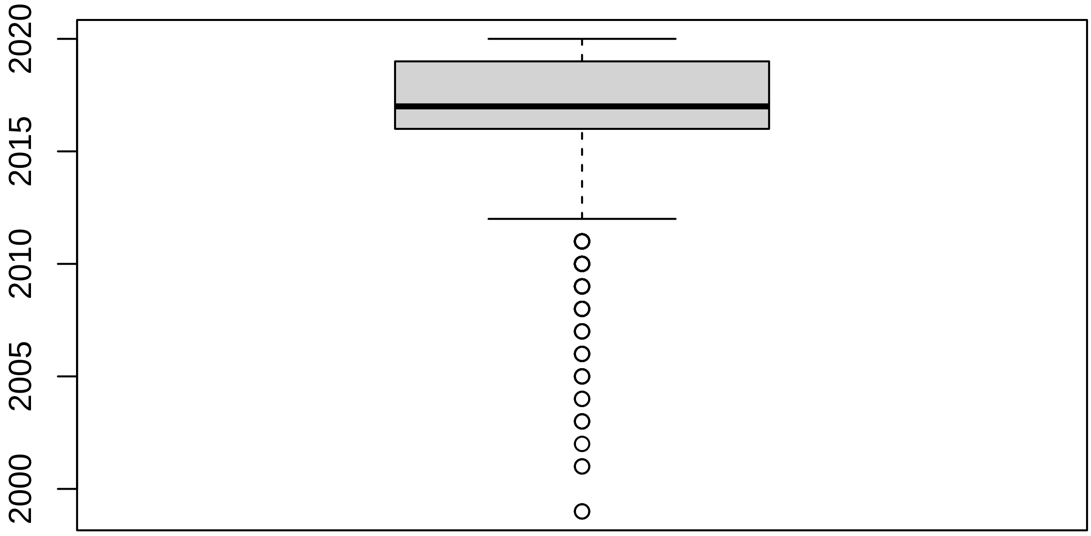
In the plot we can see that a big number of values are concentrated in the size 2. This will affect our final factorization because the group that contains this value will be much bigger than the others.

3.2.2 Year of purchase / years sell

```
summary(df$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1999    2016    2017    2017    2019    2020
```

```
boxplot(df$year)
```



The variable year of purchase is discrete because only contains 21 different values. For this reason we will group it in groups because the information that it represents is qualitative. We can see that the numbers of cars that appear before the year 2013 doesn't is very significant so we will group all of them in only one category. The variable years sell has the objective to classification the cars in a more general way. "Molt nou" < 3, "Semi nou" <=6, "Semi vell" <=10 o "Vell" if they are older than 10 year since the year 2020.

By the way we will do the classification after we treat the outliers.

A new variable derived from this one called years_sold will be created too.

3.3 Description of numeric variables that represent cuantitative concepts

In this section we will descrive numeric variables that represent directly cuantitative concepts but we will not factorize them until the next section after we have treated errors and outliers.

3.3.1 Price

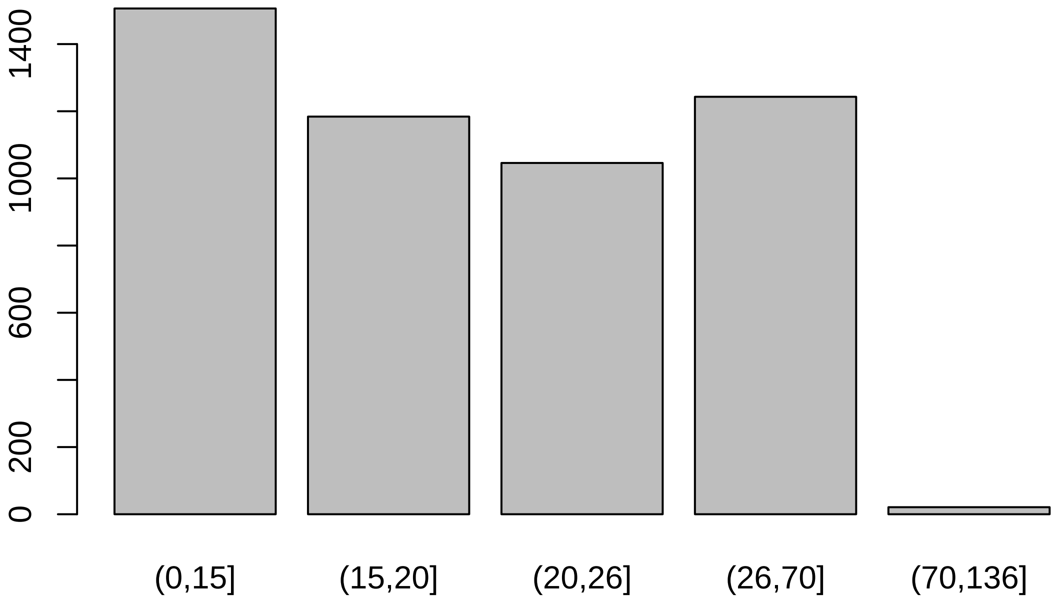
```
summary(df$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     899   13991   19498   21459   26299  135124
```

```
quantile(df$price,seq(0,1,0.25),na.rm=TRUE)
```

```
##        0%       25%       50%       75%      100%
##    899.00  13990.75  19498.00  26299.00 135124.00
```

```
barplot(table(factor(cut(df$price/1000,breaks=c(0,15,20,26, 70, 136), include.lowest = F ))))
```



Price is a numeric variable that has a lot of different values. We can see that the mean of the price is 21459 and that the prices fluctuate between 899 and 135124. The lowest values don't show us extreme cases that may be considered outliers or errors but the boxplot shows that there exist some outliers for the highest valued cars (90+) so we will treat them before factorizing the variable.

3.3.2 Mileage

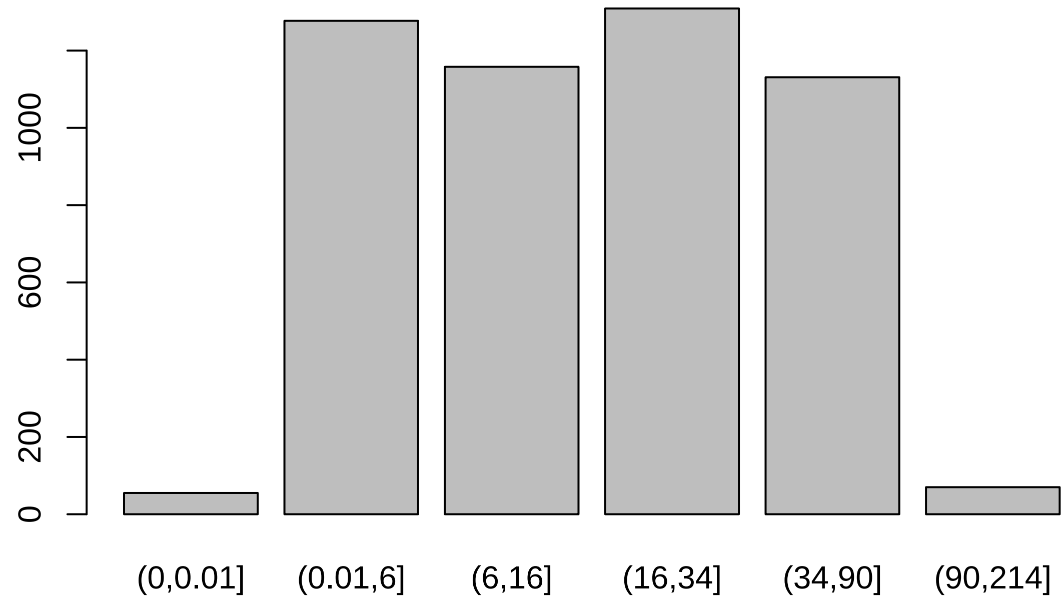
```
summary(df$mileage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1    5758   16144   22775   33187  214000
```

```
quantile(df$mileage,seq(0,1,0.25),na.rm=TRUE)
```

```
##        0%       25%       50%       75%      100%
##      1.00   5758.50  16143.50  33186.75 214000.00
```

```
barplot(table(factor(cut(df$mileage/1000,breaks=c(0,0.01, 6,16,34,90, max(df$mileage/1000)), include.low
```



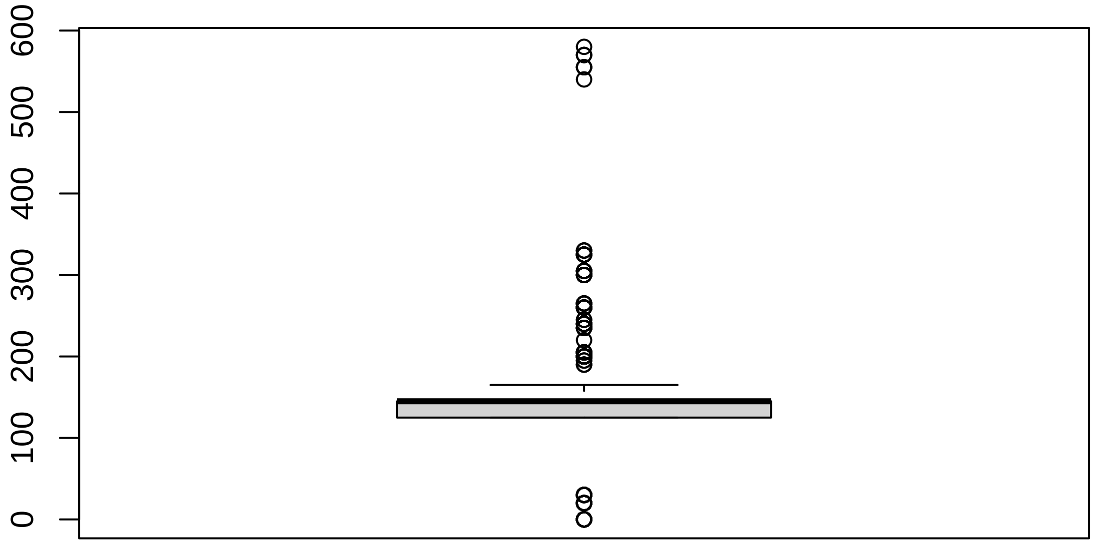
All values in the barplot are divided by 1000 to make them more legible. We will classify all the cars that has more than 10 km and less than 90. We will consider this two groups as errors and outliers. We can see that nearly 50% of the cars have less than 16.000km and the majority of them less than 90.000km.

3.3.3 Tax

```
summary(df$tax)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   125.0   145.0   122.9   145.0   580.0
```

```
boxplot(df$tax)$stats[c(1, 5), ]
```




```
## [1] 125 165
```

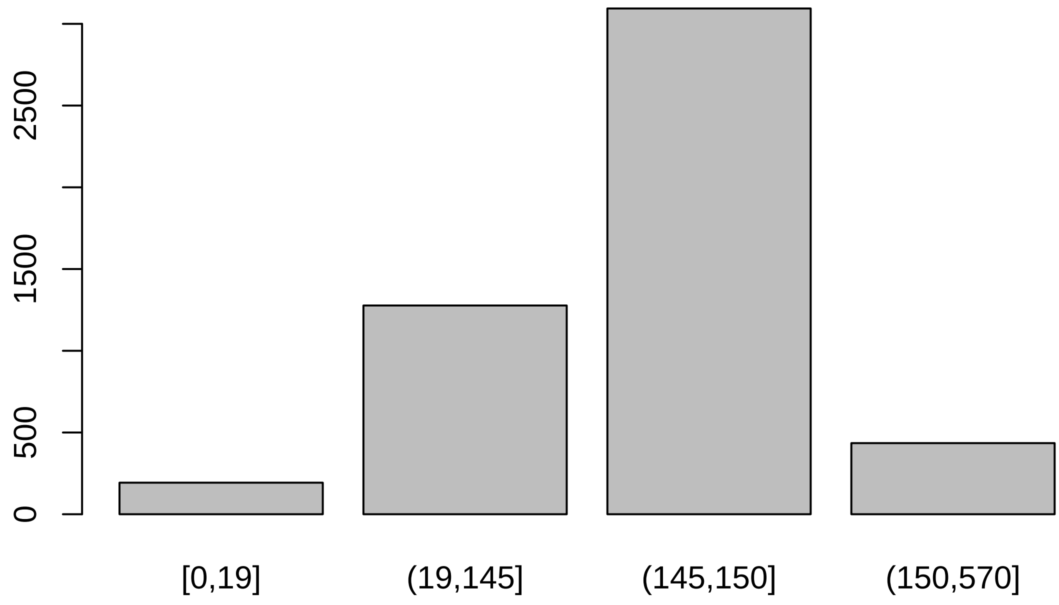
```
sort(df$tax)[194]
```

```
## [1] 20
```

```
quantile(df$tax,seq(0,1,0.25),na.rm=TRUE)
```

```
##   0%  25%  50%  75% 100%
##    0  125  145  145  580
```

```
barplot(table(factor(cut(df$tax,breaks=c(0,19, 144.9,150.1, 570), include.lowest = T ))))
```



We see that the intervals are not equally distributed for the tax variable, because there is a concentration of the values at the 150 value.

We consider that values under 20 for the variable tax are errors because are too low. By the way the only value in this interval is the 0. The next value after it is number 20.

3.3.4 mpg

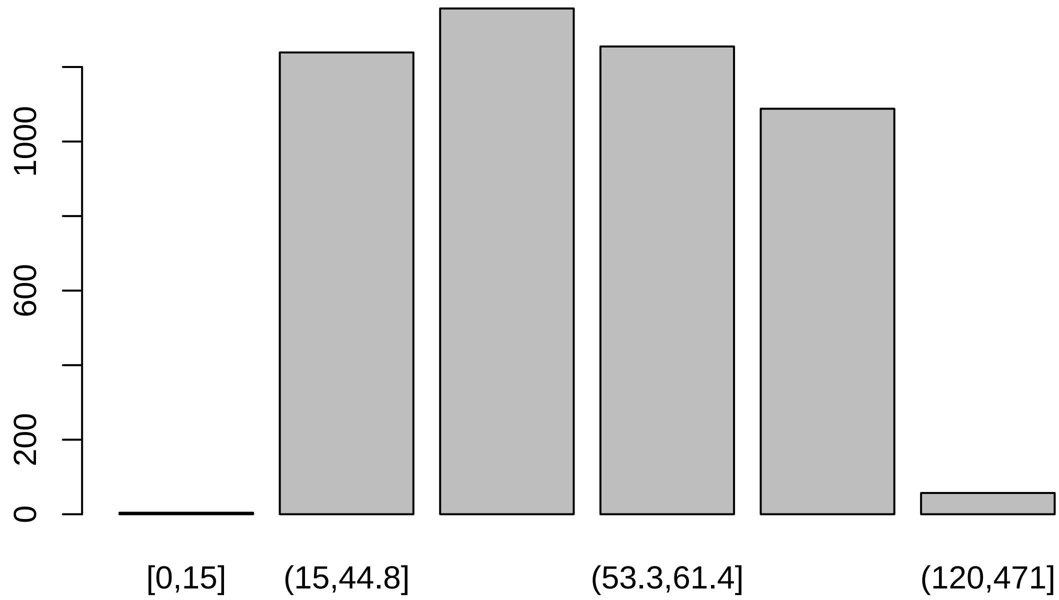
```
summary(df$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.10   45.60   53.30   54.62   61.40  470.80
```

```
quantile(df$mpg,seq(0,1,0.25),na.rm=TRUE)
```

```
##    0%   25%   50%   75%  100%
##   1.1  45.6  53.3  61.4 470.8
```

```
barplot(table(factor(cut(df$mpg,breaks=c(0, 15, 44.8,53.3,61.4, 120, 470.8), include.lowest = T ))))
```



We can see that exists an equal distribution of samples between the frour groups. The problem is that the are some cars that have too high consume values and others very low consume values. We will analyse this two cases in more detail in the next section.

4 Data quality report

```
########################################################
iouts<-rep(0,nrow(df))  # rows - cars
jouts<-rep(0,ncol(df))  # columns - variables
#######################################################
```

4.1 Missing values per variable

```
########################################################
imis<-rep(0,nrow(df))  # rows - cars
jmis<-rep(0,ncol(df))  # columns - variables
#######################################################
mis1<-countNA(df) # Counts the errors for each variable
#mis1$mis_ind # Number of missings for the current set of cars
mis1$mis_col # Number of missings for the current set of variables
```

```
##              mis_x
## model            0
## year             0
## price            0
## transmission     0
## mileage          0
## fuelType        16
## tax              0
## mpg              0
## engineSize       0
```

```
## manufacturer      0
## Audi              0
```

Doing the analysis of the missing values per variable, we can see that the one that contains missing values is the fuelType one that has 16.

4.2 Errors per variable

```
########################################################
ierrs<-rep(0,nrow(df))  # rows - cars
jerrs<-rep(0,ncol(df))  # columns - variables
#######################################################
```

4.2.1 EngineSize == 0

We see that there are some cars that have an engine size of 0. These are errors and we will transform them into NAs to avoid using them in our tasks.

```
sel<-which(df$engineSize==0)# captures the number of the row

ierrs[sel]<-ierrs[sel]+1
jerrs[9]<-length(sel) #jerrs gives us the total number of errors in the column
df[sel,"engineSize"]<- NA
#We replaced 0 by NA in order to avoid taking into account these values
jerrs
```

```
##  [1]  0  0  0  0  0  0  0  0 20  0  0
```

4.2.2 Tax == 0

Cars which pay 0 in Tax are also viewed as errors

```
sel<-which(df$tax==0)
ierrs[sel]<-ierrs[sel]+1
jerrs[7]<-length(sel)
df[sel,"tax"]<- NA
jerrs
```

```
##  [1]   0   0   0   0   0   0 193   0  20   0   0
```

There are 193 cars that didn't pay taxes, which is not normal

4.2.3 Mileage

We will also add as errors the cars that are bought for more than 1 year and have recorded less than 5000km and the cars that has 10km or less recorded.

```
sel<-which((df$mileage<=5000 & df$year<2019)|(df$mileage<=10))
ierrs[sel]<-ierrs[sel]+1
jerrs[5]<-length(sel)
df[sel,"mileage"]<- NA
jerrs
```

```
##  [1]   0   0   0   0 101   0 193   0  20   0   0
```

4.2.4 Milles per gallon

The consumption of 15 mpg equals to a consumption of aprox 16 liters every 100km. Which now a days is a value too high for comercial vehicles. We consider that all values under 16 are errors.

We have looked for non hybrid cars with big size engines and very low consumption values (less than 4.7l/100km). We thought that if there were cars with this properties they will be errors but as we can see there not exist samples of this type.

```
df[which(df[,"mpg"]<15),]
```

```
##                   model year price      transmission mileage      fuelType tax
## 1164           BMW- X3 2020 52910 f.Trans-SemiAuto     101 f.Fuel-Hybrid 135
## 1570     BMW- 3 Series 2019 33999 f.Trans-SemiAuto    8680 f.Fuel-Hybrid 135
## 1714     BMW- 3 Series 2019 35995 f.Trans-SemiAuto    2166 f.Fuel-Hybrid 135
## 2813 Mercedes- A Class 2020 31500 f.Trans-SemiAuto    1000 f.Fuel-Hybrid 135
##      mpg engineSize   manufacturer Audi
## 1164 5.5        2.0      f.Man-BMW   No
## 1570 8.8        2.0      f.Man-BMW   No
## 1714 8.8        2.0      f.Man-BMW   No
## 2813 1.1        1.3 f.Man-Mercedes   No
```

```
count(df[which((df[,"mpg"]>50)&((df[,"fuelType"]!="f.Fuel-Hybrid"))&(df[,"engineSize"]>3)),])
```

```
##   n
## 1 0
```

As we can see there are 4 vehicles with so high values and what is more the have relatively small engineSizes(mitjà or small). Looking for the real consumption values in the internet we have confirmed that these are errors.

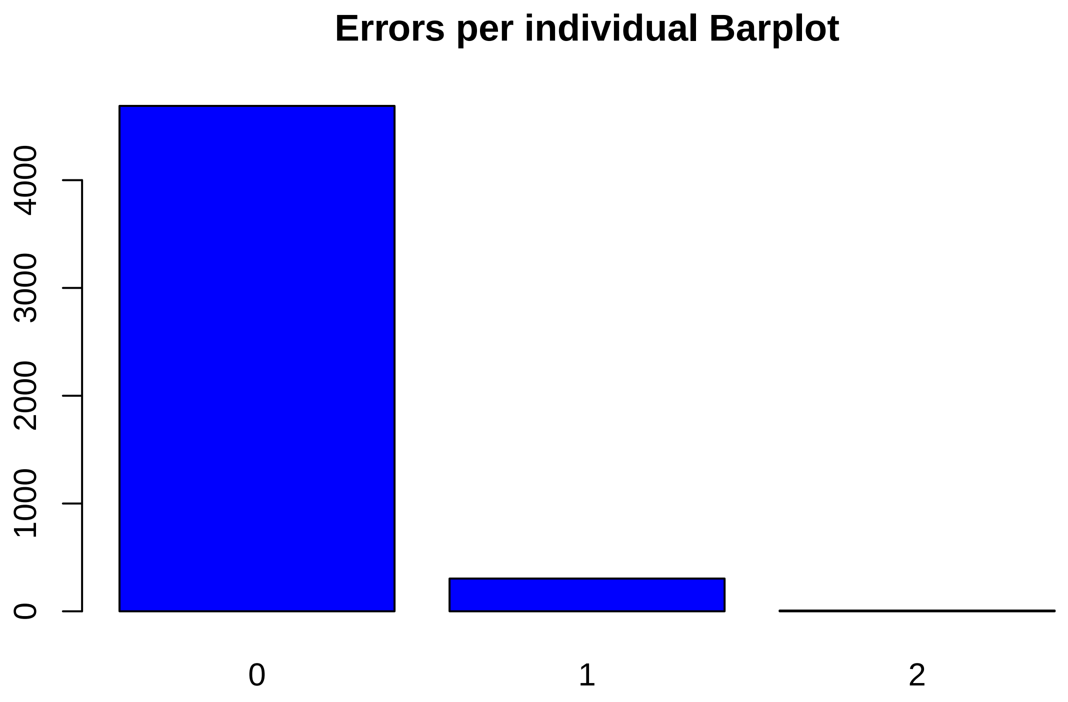
```
sel<-which(df$mpg<=15)
ierrs[sel]<-ierrs[sel]+1
jerrs[8]<-length(sel)
df[sel,"mpg"]<- NA
jerrs
```

```
##  [1]   0   0   0   0 101   0 193   4  20   0   0
```

4.2.5 Total errors

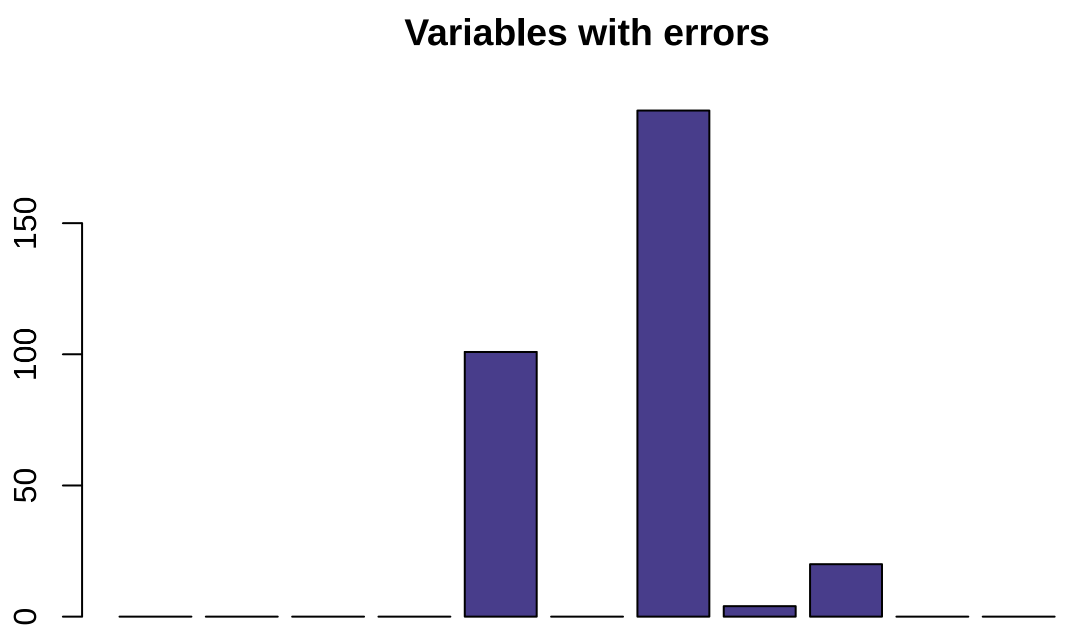
As a summary we can se that the variable engine size has 20 errors corresponding to all the engines that has a size of 0. The variable tax contains 193 errors that correspond to the values 0 or what is the same, the ones that doesn't pay taxes. There are 101 cars that have less than 10km or that have been in circulation for more than one year and have less than 5000km. Finally there are 4 cars with extremely high consume values.

```
barplot(table(ierrs),main="Errors per individual Barplot",col = "Blue")
```



The majority of the cars don't have more than an error

```
barplot(jerrs,main="Variables with errors",col = "DarkSlateBlue")
```




```
jerrs
```

```
## [1]   0   0   0   0 101   0 193   4  20   0   0
```

4.3 Outliers per variable

To end with the analysis of the quality of the different variables, we will check the outliers for all of them.

4.3.1 Price

```
# We will exclude the cars whose price is more than 70 000
var_out<-calcQ(df$price)
llout<-which(df$price>70000)
length(llout)
```

```
## [1] 21
```

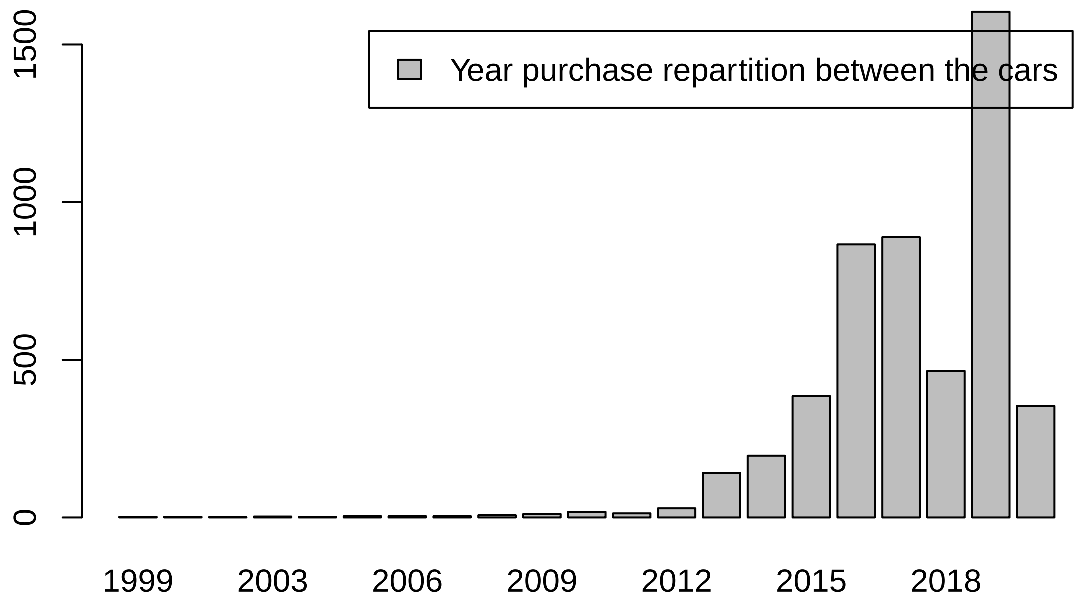
```
iouts[llout]<-iouts[llout]+1
jouts[3]<-length(llout)
df[llout,"price"]<- NA
jouts
```

```
##  [1]  0  0 21  0  0  0  0  0  0  0  0
```

We consider as outlyers all cars that are more expensive than 70000 as we have explained in the first section.

4.3.2 Year

```
barplot(table(df$year),legend.text = "Year purchase repartition between the cars")
```

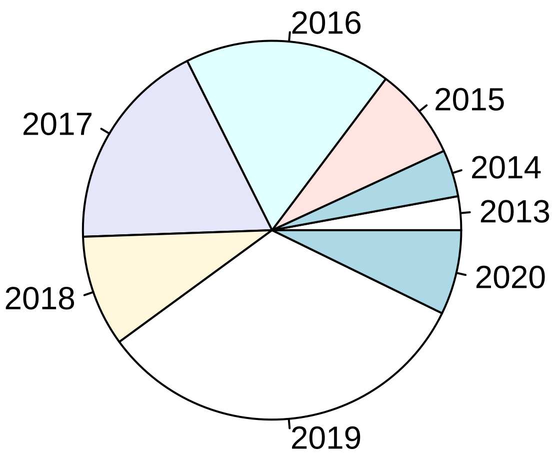


We see that there are practically no cars purchased before 2013, we will consider the cars bought before as outliers and make them NA. HAs we can see the variable jouts shows us that there are 100 cars affected by this decision.

```
set<-which(df$year<2013)
length(set)
```

```
## [1] 100
```

```
iouts[set]<-iouts[set]+1
jouts[2]<-length(set)
df[set,"year"]<- NA
pie(table(df$year))
```




```
jouts
```

```
##  [1]   0 100  21   0   0   0   0   0   0   0   0
```

4.3.3 mpg

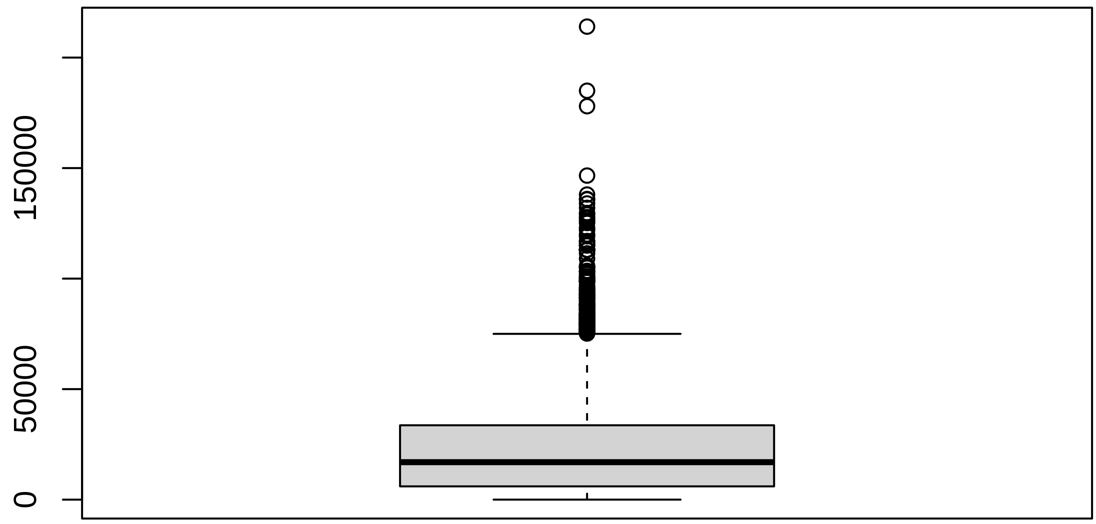
```
df[which((df[,"mpg"]>100)&((df[,"fuelType"]!="f.Fuel-Hybrid"))&(df[,"engineSize"]>3)),]
```

```
##  [1] model        year         price        transmission mileage
##  [6] fuelType     tax          mpg          engineSize   manufacturer
## [11] Audi
## <0 rows> (or 0-length row.names)
```

We see that there are some unusual values (those with mpg>100), we thought that they were out liers, but we have discovered that they correspond to the cars with hybrid engines. They might seem outliers because there are few samples with this engine type.

4.3.4 mileage

```
boxplot(df$mileage)
```



We see also that the cars with more than 150 000 km are minoritary, we will consider them as outliers

```
set<-which(df$mileage>150000)
length(set)
```

```
## [1] 3
```

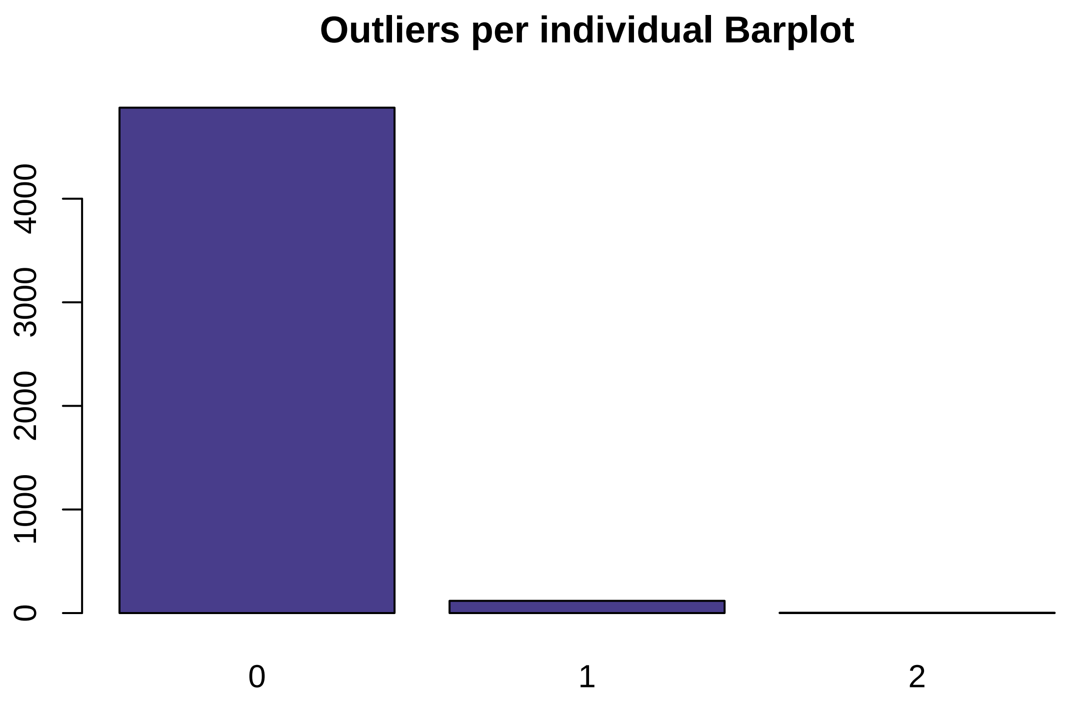
```
iouts[set]<-iouts[set]+1
jouts[5]<-length(set)
df[set,"mileage"]<- NA
jouts
```

```
##  [1]   0 100  21   0   3   0   0   0   0   0   0
```

4.3.5 Total outliers

Table of Outliers per individual Barplot

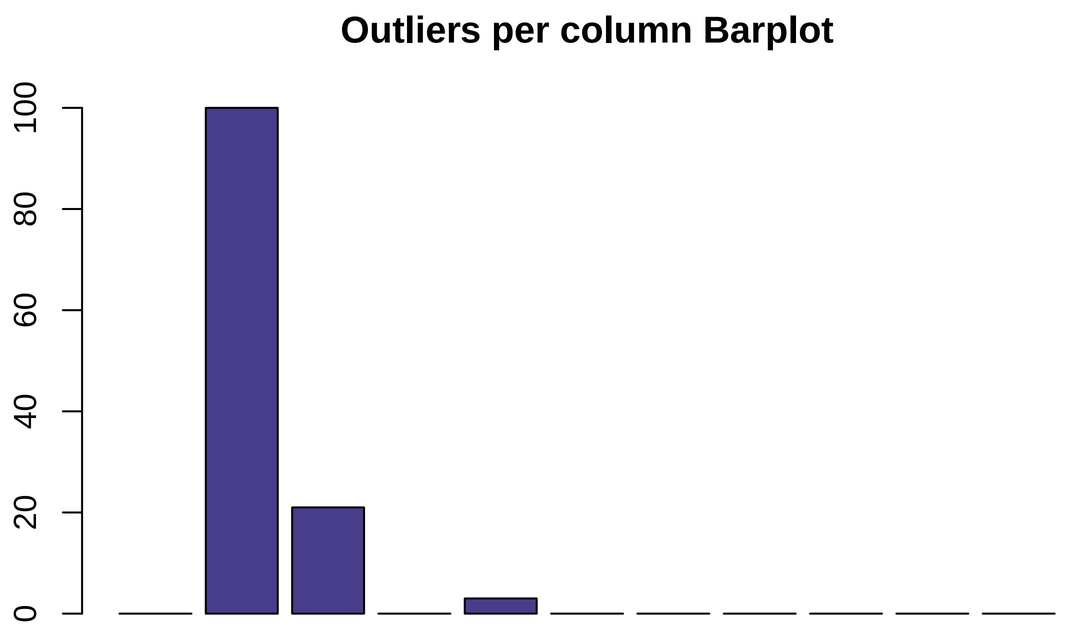
```r
barplot(table(iouts),main="Outliers per individual Barplot",col = "DarkSlateBlue")
```




```r
jouts
```

```
## [1]   0 100  21   0   3   0   0   0   0   0   0
```

```r
barplot(jouts,main="Outliers per column Barplot",col = "DarkSlateBlue")
```



As

we can see there are three variables with outliers. Rows with year<2013 (100), price > 70000 (21) and milleage >150000.

4.4 Errors, missings and oultiers summary

4.4.1 Number of missing values of each variable (with ranking)

```
missings_ranking_sortlist <- sort.list(mis1$mis_col, decreasing = TRUE)
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
for (j in missings_ranking_sortlist) {
  print(paste(names(df)[j], " : ", mis1$mis_col$mis_x[j]))
}
```

```
## [1] "fuelType  :  16"
## [1] "model  :  0"
## [1] "year  :  0"
## [1] "price  :  0"
## [1] "transmission  :  0"
## [1] "mileage  :  0"
## [1] "tax  :  0"
## [1] "mpg  :  0"
## [1] "engineSize  :  0"
## [1] "manufacturer  :  0"
## [1] "Audi  :  0"
```

4.4.2 Number of outliers per each variable

```
errors_ranking_sortlist <- sort.list(jouts, decreasing = TRUE)
for (j in errors_ranking_sortlist) {
  if(!is.na(names(df)[j])) print(paste(names(df)[j], " : ", jouts[j]))
}
```

```
## [1] "year  :  100"
## [1] "price  :  21"
## [1] "mileage  :  3"
## [1] "model  :  0"
## [1] "transmission  :  0"
## [1] "fuelType  :  0"
## [1] "tax  :  0"
## [1] "mpg  :  0"
## [1] "engineSize  :  0"
## [1] "manufacturer  :  0"
## [1] "Audi  :  0"
```

4.4.3 Number of errors per each variable

```
errors_ranking_sortlist <- sort.list(jerrs, decreasing = TRUE)
for (j in errors_ranking_sortlist) {
  if(!is.na(names(df)[j])) print(paste(names(df)[j], " : ", jerrs[j]))
}
```

```
## [1] "tax  :  193"
## [1] "mileage  :  101"
## [1] "engineSize  :  20"
```

```
## [1] "mpg  :  4"
## [1] "model  :  0"
## [1] "year  :  0"
## [1] "price  :  0"
## [1] "transmission  :  0"
## [1] "fuelType  :  0"
## [1] "manufacturer  :  0"
## [1] "Audi  :  0"
```

4.4.4 Total Errors, outliers and NA per individual

```
mis <- 0; out <- 0; err <- 0;
for (m in mis1$mis_ind) {mis <- mis + m}
for (o in iouts) {out  <- out + o}
for (e in ierrs) {err <- err + e}

mis
```

```
## [1] 16
```

```
out
```

```
## [1] 124
```

```
err
```

```
## [1] 318
```

4.4.5 Creating a new variable total with the total missing, outliers and error values for each individual

```
countNA_row <- function(x) {
  mis_i <- rep(0,ncol(x))
  for (j in 1:nrow(x)) {mis_i[j]<- sum(is.na(x[j,])) }
  mis_i}

mis1<-countNA_row(df)
#mis1=countNA_row(df)[, 1]
df$total<-factor(mis1)
```

As all the previous errors, outliers and initial missing values have been converted to missing values by addition, we will just count the number of missing values in each row in order to find the total number of (errors,outliers and initial missing values). We will then create the factor total that indicates this number

```
mis1<-countNA_row(df)
df$total<-mis1
```

We consequently added the factor total that indicates the amount of errors, outliers and missing values

```
#vars_quantitatives<-names(df)[c(1:,4:7,18)]
data<- df[,c(2,3,5,7,8, 12)]
res <- cor(data, use = "complete.obs")
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(res)
```

There is a high correlation between, mileage and year, year and price.

There is practically no correlation between mpg and mileage, tax and mileage.

5 Data Imputation

Impute realistic values to all NA values in the dataset (errors + outliers). In this section we will impute values for all 6 numeric variables. Year and engineSize are the qualitative ones and the other four are the ones that represent quantitative data.

Remove the samples that has NA as price because it is the numeric target variable

```
is.integer0 <- function(x)
{
  is.integer(x) && length(x) == 0L
}
sel <- which(is.na( df$price ))
if (!is.integer0(sel)){
  df <- df[-sel,]
}
```

```
library(missMDA)
```

```
#selection of numeric values
vars<-names(df)[c(2,3,5, 7, 8, 9)]
res.imputation<-imputePCA(df[,vars],ncp=5)
summary(res.imputation$completeObs)
```

```
##       year          price          mileage            tax
##  Min.   :2011   Min.   :  899   Min.   :-10228   Min.   :-310.2
##  1st Qu.:2016   1st Qu.:13990   1st Qu.:   6000   1st Qu.: 125.0
##  Median :2017   Median :19495   Median : 16876   Median : 145.0
```

```
##  Mean   :2017   Mean   :21166   Mean   : 22977   Mean   : 125.4
##  3rd Qu.:2019   3rd Qu.:26000   3rd Qu.: 33315   3rd Qu.: 145.0
##  Max.   :2020   Max.   :70000   Max.   :146604   Max.   : 580.0
##       mpg          engineSize
##  Min.   : 17.80   Min.   :0.600
##  1st Qu.: 45.60   1st Qu.:1.500
##  Median : 53.30   Median :2.000
##  Mean   : 54.77   Mean   :1.892
##  3rd Qu.: 61.40   3rd Qu.:2.000
##  Max.   :470.80   Max.   :6.600
```

Now we have to correct all the errors that have been created by the procedure.

5.1 year

```
ll<-which(res.imputation$completeObs[,"year"] < 2013)
res.imputation$completeObs[ll,"year"] <- 2013
```

5.2 mileage

```
ll<-which(res.imputation$completeObs[,"mileage"] <= 0)
res.imputation$completeObs[ll,"mileage"] <- 11
```

5.3 tax

```
ll<-which(res.imputation$completeObs[,"tax"] <= 0)
res.imputation$completeObs[ll,"tax"] <- 20
```

5.4 Aplly changes to dataset

```
df[,vars] <- res.imputation$completeObs
summary(df)
```

```
##                 model           year          price
##  VW- Golf           : 477   Min.   :2013   Min.   :  899
##  Mercedes- C Class: 385   1st Qu.:2016   1st Qu.:13990
##  VW- Polo           : 368   Median :2017   Median :19495
##  Mercedes- A Class: 265   Mean   :2017   Mean   :21166
##  BMW- 3 Series      : 247   3rd Qu.:2019   3rd Qu.:26000
##  Mercedes- E Class: 201   Max.   :2020   Max.   :70000
##  (Other)            :3036
##            transmission     mileage                fuelType         tax
##  f.Trans-Manual   :1835   Min.   :     11   f.Fuel-Diesel:2818   Min.   : 20.0
##  f.Trans-SemiAuto :1892   1st Qu.:   6000   f.Fuel-Petrol:2062   1st Qu.:125.0
##  f.Trans-Automatic:1252   Median : 16876   f.Fuel-Hybrid:   83   Median :145.0
##                           Mean   : 22987   NA's         :   16   Mean   :125.8
##                           3rd Qu.: 33315                         3rd Qu.:145.0
##                           Max.   :146604                         Max.   :580.0
##
##       mpg          engineSize           manufacturer    Audi
##  Min.   : 17.80   Min.   :0.600   f.Man-Audi     :1020   No :3959
##  1st Qu.: 45.60   1st Qu.:1.500   f.Man-BMW      :1101   Yes:1020
##  Median : 53.30   Median :2.000   f.Man-Mercedes:1299
##  Mean   : 54.77   Mean   :1.892   f.Man-VW       :1559
```

```
##   3rd Qu.: 61.40   3rd Qu.:2.000
##   Max.   :470.80   Max.   :6.600
##
##       total
##   Min.   :0.00000
##   1st Qu.:0.00000
##   Median :0.00000
##   Mean   :0.08757
##   3rd Qu.:0.00000
##   Max.   :2.00000
##
```

6 Discretization

6.1 Numeric variables qualitative concepts

There are 4 Original numeric variables corresponding to qualitative concepts that will be converted to factors.

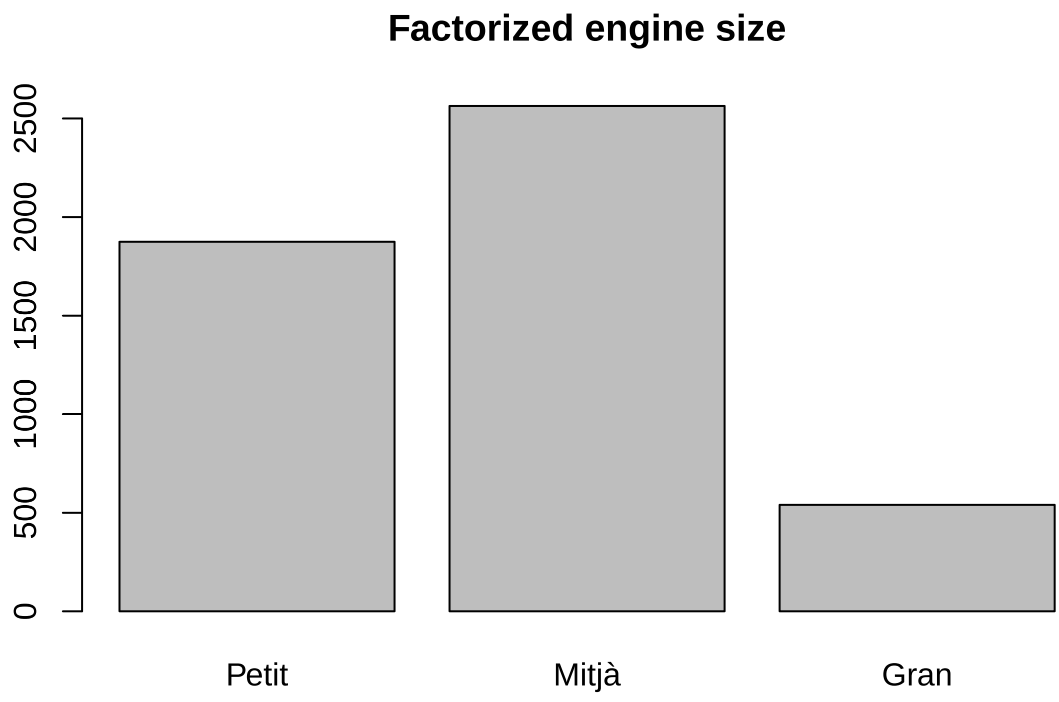
6.1.1 Enigine Size

```
table(df$engineSize)
```

```
##
##               0.6 0.732646797347417 0.788341767789462 0.791884621566836
##                 1                 1                 1                 1
## 0.876032342265793 0.916103042038359 0.964059108087171                 1
##                 1                 1                 1               398
##               1.2               1.3               1.4  1.43592464844068
##               140                75               318                 1
##  1.44883342455924               1.5  1.57544752263109               1.6
##                 1               515                 1               377
##  1.61790647300094  1.63916287299868  1.68149509853343  1.76325108180272
##                 1                 1                 1                 1
##               1.8  1.85655366779918  1.86148841411089               1.9
##                34                 1                 1                 1
##  1.95662548941302                 2   2.0883662007312               2.1
##                 1              2116                 1               399
##               2.2               2.3    2.311131276639  2.41201001221907
##                10                10                 1                 1
##               2.5   2.5283444906766               2.7               2.8
##                 7                 1                 1                 1
##               2.9                 3               3.2               3.5
##                16               511                 1                 1
##                 4               4.2               4.3               4.4
##                18                 1                 1                 3
##               4.7               6.2               6.6
##                 1                 2                 1
```

```
df[which(df$engineSize>0 & df$engineSize < 2), "engineSize"] <- 1
df[which(df$engineSize>=2 & df$engineSize < 3), "engineSize"]<- 2
df[which(df$engineSize>=3), "engineSize"]<- 3
df$engineSize<-factor(df$engineSize,labels=c("Petit","Mitjà","Gran"))

barplot(table(df$engineSize), main="Factorized engine size")
```




```
summary(df$engineSize)
```

```
## Petit Mitjà  Gran
##  1875  2564   540
```

In first place we can find engine size. It is a numerical variable that represents a finite number of different engine sizes. For our analysis it is not very interesting to know exact size of an engine. For this reason we will group all size in 3 different categories. Category "Petit" = (0, 2), "Mitjà" = [2, 3) and "Gran" [3, infinite]

We can see that the barplot of the factorized variable shows that the group "Mitjà" groups a big portion of the total number of cars. This is because the value "2" of the original variable, that is represented in the grup "Mitjà", groups on its own a total of 2116 cars.

6.1.2 Year of purchase / years sell

The variable year of purchase is discrete because only contains 21 different values. For this reason we will factorize it because the information that it represents is qualitative. We can see that the numbers of cars that appear before the year 2013 doesn't is very significant so we will group all of them in only one category. The variable years sell has the objectivo to classificate the cars in a more general way. "Molt nou" < 3, "Semi nou" <=6, "Vell" <=10.

```
summary(df$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2013    2016    2017    2017    2019    2020
```

```
str(df$year)
```

```
##  num [1:4979] 2017 2015 2017 2018 2016 ...
```

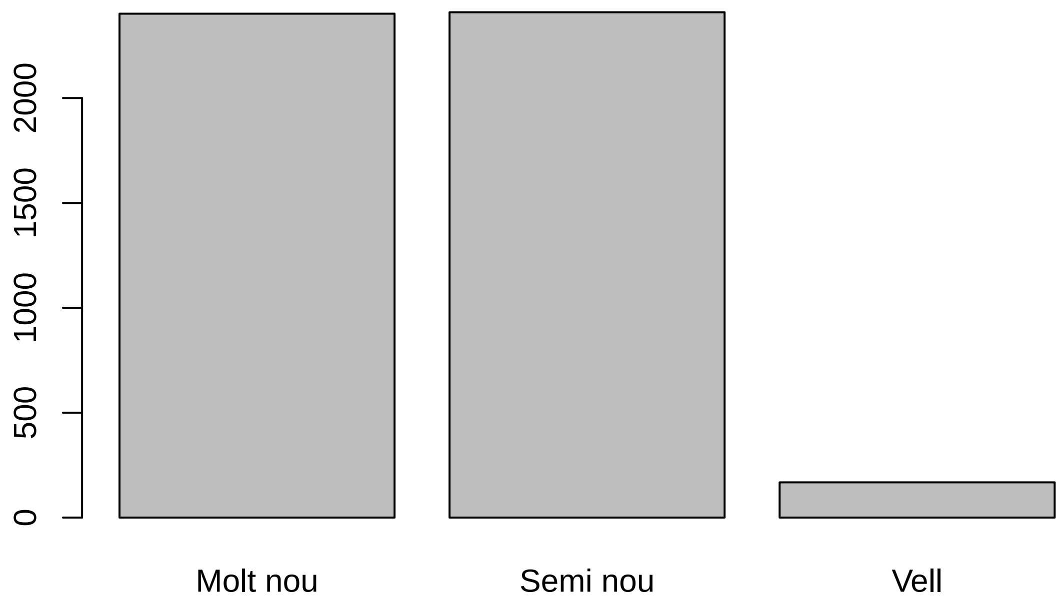
```
df$years_sell <-  as.integer(2020 - df$year)
df[which(df$years_sell < 3), "years_sell2"] <- 1
df[which(df$years_sell >= 3 & df$years_sell <= 6), "years_sell2"] <- 2
df[which(df$years_sell > 6 & df$years_sell <= 10), "years_sell2"] <- 3
df[which(df$years_sell > 10), "years_sell2"] <- 4
df$years_sell<-factor(df$years_sell2,labels=c("Molt nou","Semi nou","Vell"))

df[which(df$year<2013),"year"] <- "2012 or before"
df$year <- factor(df$year)

summary(df$years_sell)
```

```
## Molt nou Semi nou     Vell
##     2402     2409      168
```

```
barplot(table(df$years_sell))
```



6.2 Discretization of numeric variables cuantitative concepts

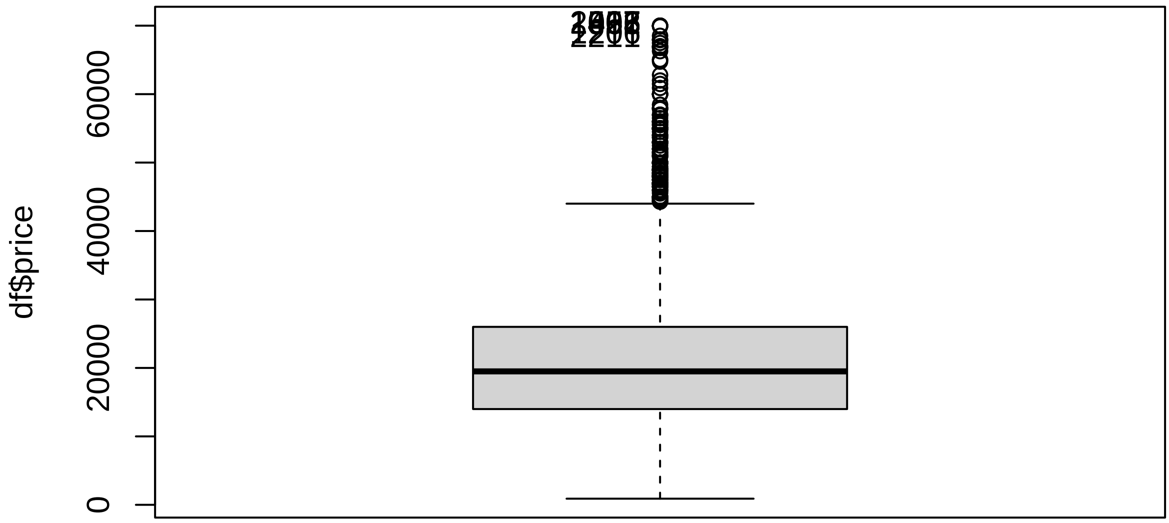
Original numeric variables corresponding to real quantitative concepts are kept as numeric but additional factors should also be created as a discretization of each numeric variable.

6.2.1 Price

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     899   13990   19495   21166   26000   70000
```

```r
Boxplot(df$price)
```




```
##  [1] 1003 1657 1515 1627 1617 2566 1491  968 1206 2211
```

```r
quantile(df$price,seq(0,1,0.25),na.rm=TRUE)
```

```
##    0%   25%   50%   75%  100%
##   899 13990 19495 26000 70000
```

```r
df$aux<-factor(cut(df$price,breaks=c(min(df$price),13995,19498,2690, max(df$price)),include.lowest = T )
summary(df$aux)
```

```
##     [899,2.69e+03] (2.69e+03,1.4e+04] (1.4e+04,1.95e+04]   (1.95e+04,7e+04]
##                  7               1253               1247               2472
```

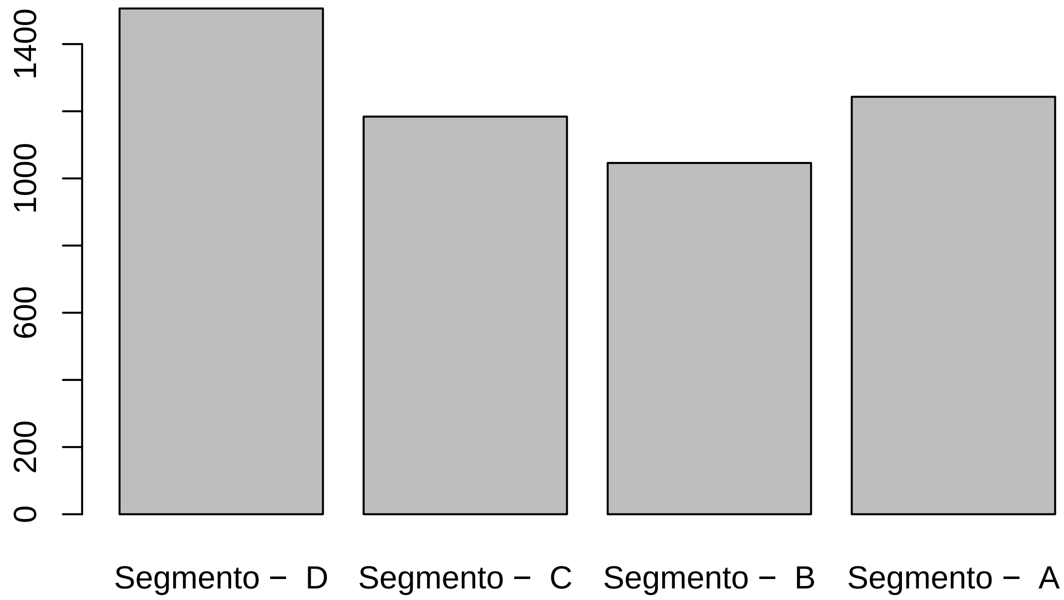
```r
tapply(df$price,df$aux,median)
```

```
##     [899,2.69e+03] (2.69e+03,1.4e+04] (1.4e+04,1.95e+04]   (1.95e+04,7e+04]
##               1595              10955              16990              26248
```

```r
df$f.price<-factor(cut(df$price/1000,breaks=c(0,15,20,26, 90)),labels=paste("Segmento - ",c("D","C","B",
table(df$f.price,useNA="always")
```

```
##
## Segmento -  D Segmento -  C Segmento -  B Segmento -  A          <NA>
##          1506          1184          1046          1243             0
```

```r
barplot(table(df$f.price))
```



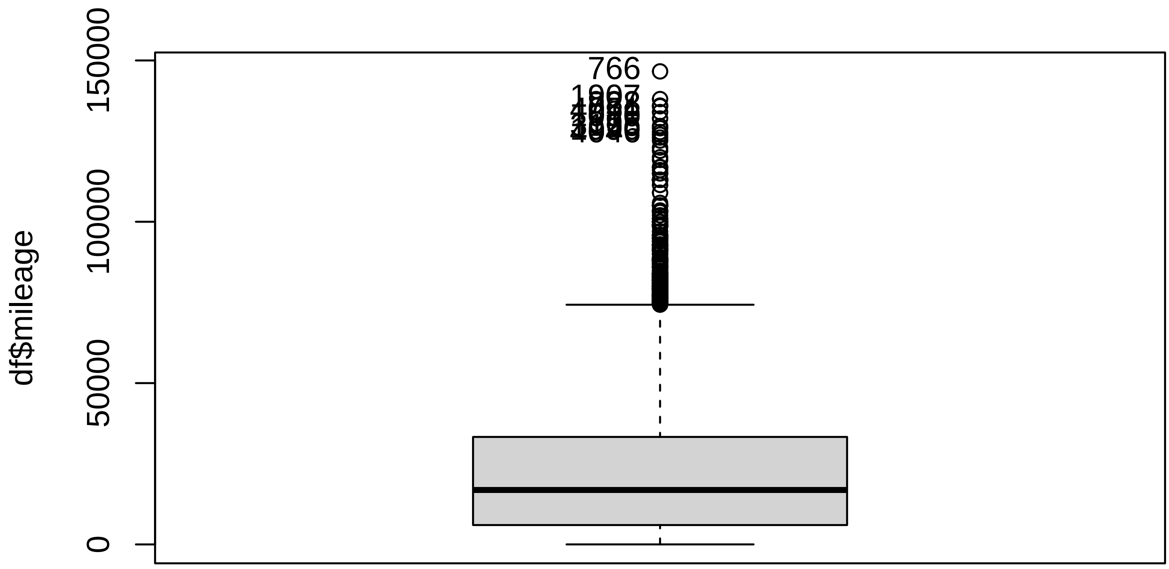
The discretization of the price value has been done in four groups beeing the "Segmento D" the cheapest ones and the "Segmento A" the most expensive ones.

6.2.2 Mileage

```
summary(df$mileage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      11    6000   16876   22987   33315  146604
```

```
Boxplot(df$mileage)
```




```
## [1]  766 1907  982  728 4051 1010  895 3125 1990 4046
```

```
quantile(df$mileage,seq(0,1,0.25),na.rm=TRUE)
```

```
##     0%    25%    50%    75%   100%
##     11   6000  16876  33315 146604
```

```
df$aux<-factor(cut(df$mileage,breaks=c(0,5891,16908,33981,323000),include.lowest = T ))
summary(df$aux) # We want to know the number of cars in each interval
```

```
##       [0,5.89e+03] (5.89e+03,1.69e+04]  (1.69e+04,3.4e+04]  (3.4e+04,3.23e+05]
##               1191                1303                1278                1207
```

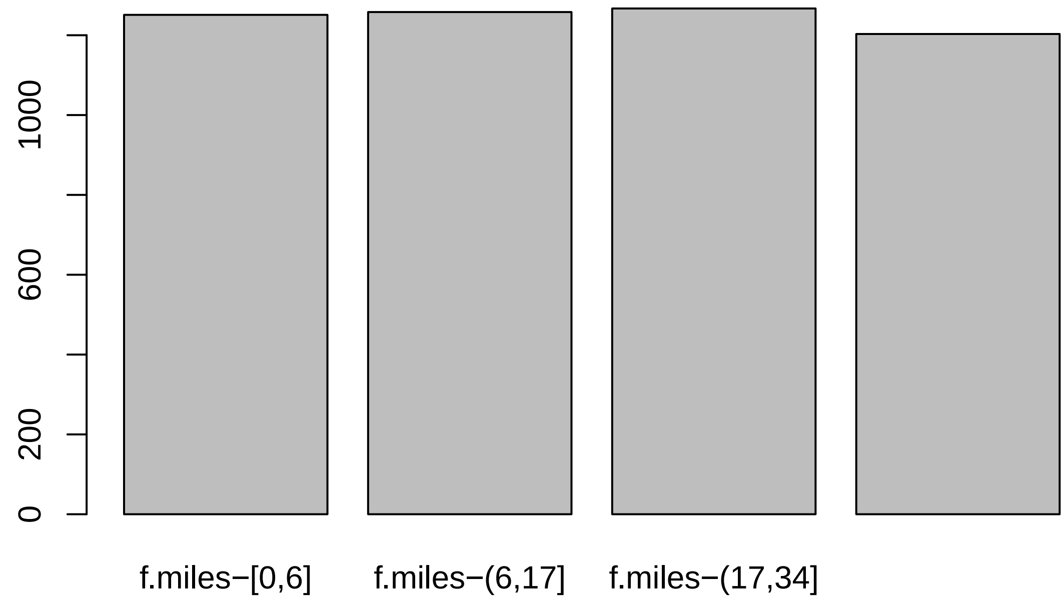
```
tapply(df$mileage,df$aux,median) #gives us the median value of the mileage of the car in the four interv
```

```
##       [0,5.89e+03] (5.89e+03,1.69e+04]  (1.69e+04,3.4e+04]  (3.4e+04,3.23e+05]
##           2943.919           10217.000           24667.000           48488.000
```

```
df$f.miles<-factor(cut(df$mileage/1000,breaks=c(0,6,17,34, 323),include.lowest = T )) # We divide by 100
levels(df$f.miles)<-paste("f.miles-",levels(df$f.miles),sep="")
table(df$f.miles,useNA="always")
```

```
##
##    f.miles-[0,6]    f.miles-(6,17]  f.miles-(17,34] f.miles-(34,323]
##             1251              1258             1267             1203
##             <NA>
##                0
```

```
barplot(table(df$f.miles))
```

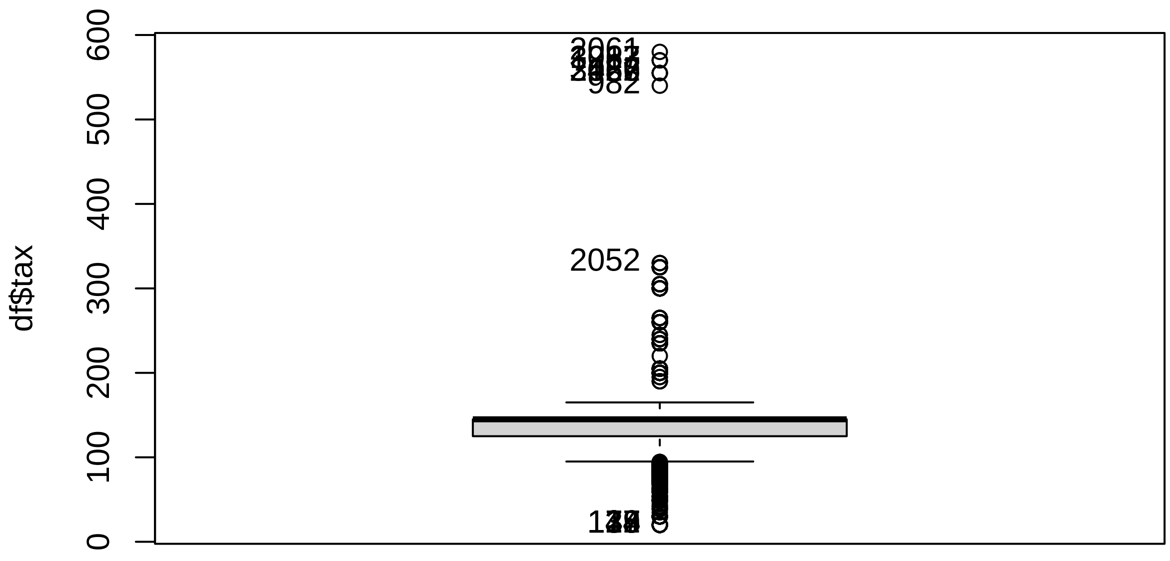


6.2.3 Tax

```
summary(df$tax)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    20.0   125.0   145.0   125.8   145.0   580.0
```

```
Boxplot(df$tax)
```




```
## [1]   17   29   32   34   78  134  135  138  145  147 2061 1013 3263 3287  466
## [16] 2069 2127 3082  982 2052
```

```r
sort(df$tax)[194]
```

```
## [1] 20
```

```r
sort(df$tax)[194]
```

```
## [1] 20
```

```r
quantile(df$tax,seq(0,1,0.25),na.rm=TRUE)
```

```
##   0%  25%  50%  75% 100%
##   20  125  145  145  580
```

```r
quantile(df$tax,seq(0,1,0.1),na.rm=TRUE)
```

```
##        0%       10%       20%       30%       40%       50%       60%       70%
##  20.00000  20.00000  78.22434 145.00000 145.00000 145.00000 145.00000 145.00000
##       80%       90%      100%
## 145.00000 150.00000 580.00000
```

```r
df$f.tax<-factor(cut(df$tax,breaks=c(0, 1, 125, 144.9,150.1, 570),include.lowest = T ))
summary(df$f.tax) # We want to know the number of cars in each interval
```

```
##   (1,125] (125,145] (145,150] (150,570]      NA's
##      1429        41      3073       435         1
```

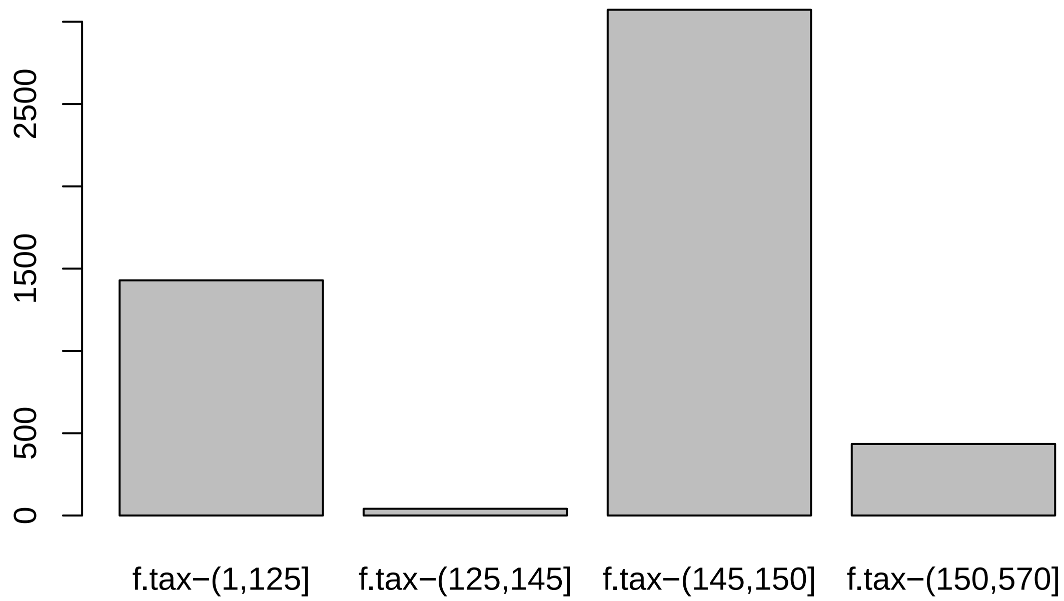
```r
tapply(df$tax,df$f.tax,median)#gives us the median value of the tax of the car in the four intervals
```

```
##   (1,125] (125,145] (145,150] (150,570]
##        30       135       145       200
```

```r
levels(df$f.tax)<-paste("f.tax-",levels(df$f.tax),sep="")
table(df$f.tax,useNA="always")
```

```
##
##   f.tax-(1,125] f.tax-(125,145] f.tax-(145,150] f.tax-(150,570]            <NA>
##            1429              41            3073             435               1
```

```r
barplot(table(df$f.tax))
```



We see that the intervals are not equally distributed for the tax variable, because there is a concentration of the values at the 150 value.

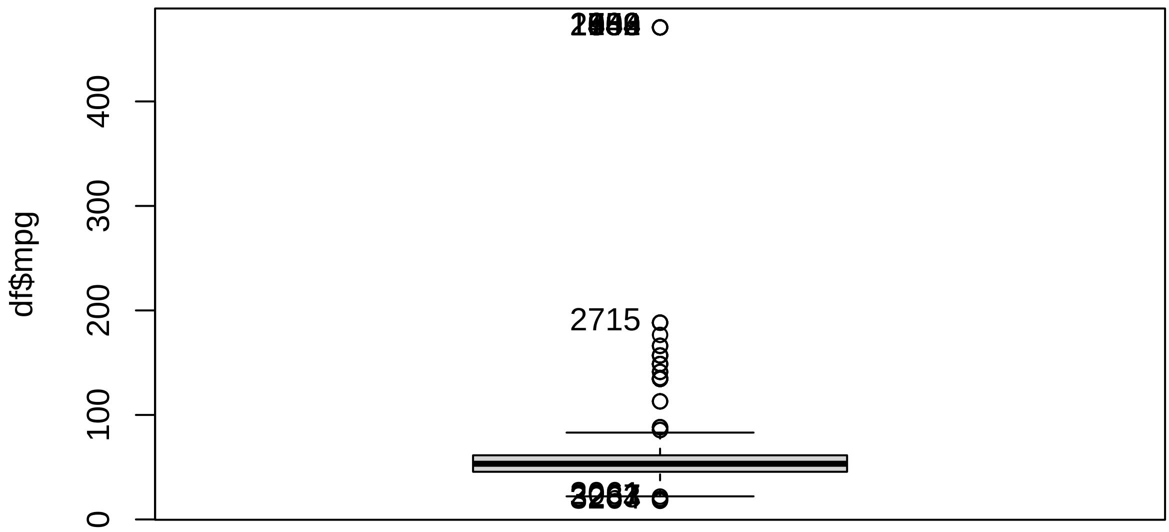
We consider that values under 1 for the variable tax are errors because are too low. By the way the only value in this interval is the 0. The next value after it is number 20.

6.2.4 mpg

```r
summary(df$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.80   45.60   53.30   54.77   61.40  470.80
```

```r
Boxplot(df$mpg)
```




```
##  [1] 2061 3263 3264 3287 1353 1446 1542 1659 1744 1836 1982 2059 2104 2715
```

```
quantile(df$mpg,seq(0,1,0.25),na.rm=TRUE)
```

```
##    0%   25%   50%   75%  100%
##  17.8  45.6  53.3  61.4 470.8
```

```
quantile(df$mpg,seq(0,1,0.1),na.rm=TRUE)
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##  17.8  37.7  42.8  47.1  50.4  53.3  56.5  60.1  64.2  68.9 470.8
```

```
df$mpg_d<-factor(cut(df$mpg,breaks=c(0,44.8,53.3,61.4 , 470.8),include.lowest = T ))
summary(df$mpg_d) # We want to know the number of cars in each interval
```

```
##    [0,44.8] (44.8,53.3] (53.3,61.4]   (61.4,471]
##        1219        1360        1255        1145
```

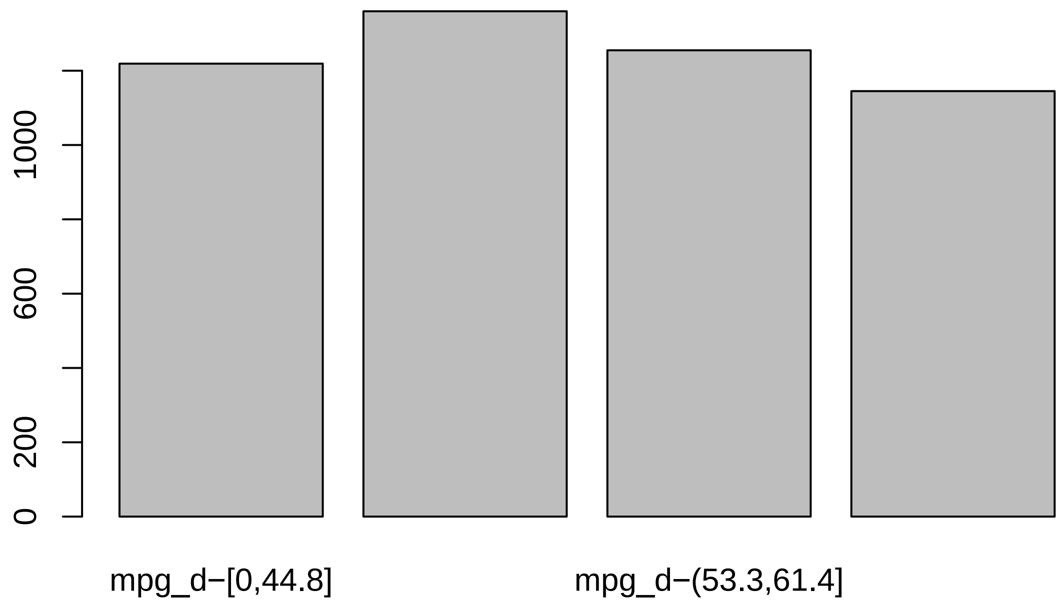
```
tapply(df$mpg,df$mpg_d,median)#gives us the median value of the tax of the car in the four intervals
```

```
##    [0,44.8] (44.8,53.3] (53.3,61.4]   (61.4,471]
##        39.2        49.6        58.9        67.3
```

```
levels(df$mpg_d)<-paste("mpg_d-",levels(df$mpg_d),sep="")
table(df$mpg_d,useNA="always")
```

```
##
##    mpg_d-[0,44.8] mpg_d-(44.8,53.3] mpg_d-(53.3,61.4]   mpg_d-(61.4,471]
##              1219              1360              1255               1145
##              <NA>
##                 0
```

```r
barplot(table(df$mpg_d))
```



7 Profiling

##Target price

```r
library(FactoMineR)
summary(df$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     899   13990   19495   21166   26000   70000
```

```r
res.condes<-condes(df,3)
```

7.0.1 Numeric variables

```r
res.condes$quanti  # Global association to numeric variables
```

```
##               correlation       p.value
## tax             0.3484210 4.350392e-142
## total          -0.1813625  4.430203e-38
## mpg            -0.2825608  4.726630e-92
## mileage        -0.5511147  0.000000e+00
## years_sell2    -0.5899469  0.000000e+00
```

We can see a relation between the price and mileage as the p-value is greater than 0.01, and the correlation is thus greater in absolute value than 0.5. We can see a relation between the price and years_sell as the p-value is greater than 0.01, and the correlation is thus greater in absolute value than 0.5.

7.0.2 Qualitative variables

```
res.condes$quali # Global association to factors
```

```
##                    R2       p.value
## model       0.517573372  0.000000e+00
## year        0.420349923  0.000000e+00
## transmission 0.259782914  0.000000e+00
## engineSize   0.267363533  0.000000e+00
## years_sell   0.355016952  0.000000e+00
## aux          0.337934639  0.000000e+00
## f.price      0.782680339  0.000000e+00
## f.miles      0.339303114  0.000000e+00
## f.tax        0.274692486  0.000000e+00
## mpg_d        0.309505916  0.000000e+00
## manufacturer 0.099023476 3.970378e-112
## fuelType     0.010172140  5.212557e-11
## Audi         0.003629495  2.101479e-05
```

We can see a relation between the price and model as the p-value is greater than 0.01, and the correlation is thus greater in absolute value than 0.5. We can see a relation between the price and year as the p-value is greater than 0.01, and the correlation is thus greater in absolute value than 0.5.

##Target Audi

```
library(FactoMineR)
summary(df$Audi)
```

```
##   No  Yes
## 3959 1020
```

```
# The "variable to describe cannot have NA ###################################
res.catdes<-catdes(df,11, proba = 0.50)
```

7.0.3 Numeric variables

```
res.catdes$quanti.var  # Global association to numeric variables
```

```
##                   Eta2      P-value
## mpg         0.0074500591 1.058502e-09
## price       0.0036294954 2.101479e-05
## mileage     0.0020681558 1.328260e-03
## years_sell2 0.0011754195 1.555127e-02
## tax         0.0005777409 8.991142e-02
```

We can see a relation between the Audi variable and tax as the p-value is greater than 0.01. We can see a relation between the Audi variable and price as the p-value is greater than 0.01.

7.0.4 Qualitative variables

```
res.catdes$test.chi2 # Global association to factors
```

```
##                 p.value df
## model       0.000000e+00 86
## manufacturer 0.000000e+00  3
## mpg_d       4.950957e-17  3
```

```
## fuelType     2.317869e-08  3
## f.price      3.397277e-05  3
## f.miles      4.248730e-04  3
## transmission 1.152834e-03  2
## aux          6.407798e-03  3
## f.tax        7.978105e-03  4
## years_sell   3.907109e-02  2
## engineSize   9.671930e-02  2
```

We can see a relation between the Audi variable and manufacturer as the p-value is greater than 0.01. We can see a relation between the Audi variable and model as the p-value is greater than 0.01. We can see a relation between the Audi variable and engineSize as the p-value is greater than 0.01.

```
#Save data
vars_con<-names(df)[c(5,7,8, 11)]
vars_dis<-names(df)[c(1:2, 4, 6, 9, 10, 12, 13, 14, 15, 16)]
var_mout <- names(df)[c(18)]
vars_res<-names(df)[c(3, 17)]
save( list = c( "vars_con","vars_dis", "vars_res", "var_mout", "df"), file = "Output-Othman-ELoi.RData"
```