Deliverable 1

Data Processing, Description, Validation and Profiling

Pere Arnau Alegre & Andrés Jiménez González

May 31, 2022

Contents

1 Data description

- Description https://www.kaggle.com/datasets/adityadesai13/used-car-dataset-ford-and-mercedes
- Data Dictionary - Scraped data of used cars, which have been separated into files corresponding to each car manufacturer (only Mercedes, BMW, Volkswagen and Audi cars are to be considered).

1.1 Variables

- Model
  - A string indicating the model of the car.
- Year
  - A discrete numeric variable to indicate the year the car was sold
- Price
  - Continuous variable indicating the price at which the car was sold
- Transmission
  - Categorical variable that indicates the type of transmission of the car
  - Values:
    - Automatic
    - Manual
    - Semi-Automatic
    - Other
- Mileage
  - A discrete numeric variable to indicate the number of miles the car had when it was sold
- Fuel Type
  - Categorical variable that indicates the type of fuel of the car
  - Values:
    - Diesel
    - Electric
    - Hybrid
    - Petrol
    - Other
- Tax
  - A discrete numeric variable to indicate the road tax of the vehicle.
- MPG
  - Continuous variable indicating the fuel consumption of the car
- Engine Size
  - Continuous variable indicating the size of the engine
- Manufacturer
  - Categorical variable that indicates the manufacturer brand of the car.
  - Values:
    - Mercedes
    - Audi
    - Volkswagen
    - BMW

2 Loading of Required Packages for the deliverable

We load the necessary packages and set the working directory

```
setwd("C:/Users/TOREROS-II/Documents/ANDRES/UNI/ADEI/trabajo/deliverable1")
# setwd('C:/Users/Arnau/Desktop/adei/deliverable1') Load Required
# Packages
options(contrasts = c("contr.treatment", "contr.treatment"))
requiredPackages <- c("missMDA", "chemometrics", "mvoutlier", "effects",
    "FactoMineR", "car", "factoextra", "RColorBrewer", "dplyr", "ggmap",
    "ggthemes", "knitr", "corrplot")
missingPackages <- requiredPackages[!(requiredPackages %in% installed.packages()[,
    "Package"])]
if (length(missingPackages)) install.packages(missingPackages)
lapply(requiredPackages, require, character.only = TRUE)
```

2.1 Select a sample of 5000 records

From the proposed database, we need to select a sample of 5000 records randomly so we can start analyzing our data.

```
if (!is.null(dev.list())) dev.off()  # Clear plots
rm(list = ls())  # Clean workspace
```

Data: used_car_dataset.csv

```
filepath <- "C:/Users/TOREROS-II/Documents/GitHub/adei/adei"
# filepath<-'C:/Users/Arnau/Desktop/adei/deliverable1'
df <- read.table(paste0(filepath, "/sample_5000.csv"), header = T, sep = ",")[c(-1)]

# dim(df) # Displays the sample size names(df) # Displays the names
# of the sample variables summary(df)
```

tinytex::install_tinytex() ## Some useful functions

```
calcQ <- function(x) {
    # Function to calculate the different quartiles
    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) {
    # Function to count the NA values
    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) {
    # Function to count a specific number of appearences
    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)
}
```

Univariate Description and Preprocessing

2.2 Variable initialization of missings, outliers and errors

```
jmis <- rep(0, ncol(df))  # columns - variables

mis1 <- countNA(df)
# mis1$mis_ind # Number of missings for the current set of cars
# (observations) mis1$mis_col # Number of missings for the current
# set of variables

jouts <- rep(0, ncol(df))  # columns - variables

jerrs <- rep(0, ncol(df))  # columns - variables

imis <- rep(0, nrow(df))  # rows - cars

iouts <- rep(0, nrow(df))  # rows - cars

ierrs <- rep(0, nrow(df))  # rows - cars
```

2.3 Preprocessing of Qualitative/Categorical & Numerical variables

Description: We need to do an analysis of all the variables to be able to identify missings, errors and outliers. We will also try to factorize each variable to make it easier to understand the sample.

2.3.1 Model

This variable indicates the model of the car.

```
df$model <- factor(paste0(df$manufacturer, "-", df$model))
# levels(df$model)
summary(df$model)
```

```
##            Audi- A1            Audi- A3            Audi- A4            Audi- A5
##                 130                 196                 143                  84
##            Audi- A6            Audi- A7            Audi- A8            Audi- Q2
##                  89                   8                  12                  74
##            Audi- Q3            Audi- Q5            Audi- Q7            Audi- Q8
##                 155                  95                  42                   8
##            Audi- R8           Audi- RS3           Audi- RS4           Audi- RS5
##                   6                   3                   1                   1
##           Audi- RS6           Audi- S3            Audi- S4            Audi- S8
##                   7                   1                   1                   1
##           Audi- SQ5           Audi- SQ7            Audi- TT       BMW- 1 Series
##                   2                   2                  28                 190
##       BMW- 2 Series       BMW- 3 Series       BMW- 4 Series       BMW- 5 Series
##                 129                 251                 113                  94
```

```
##       BMW- 6 Series       BMW- 7 Series       BMW- 8 Series            BMW- i3
##                  16                  12                   5                   5
##             BMW- M3             BMW- M4             BMW- M5             BMW- X1
##                   2                  15                   5                  81
##             BMW- X2             BMW- X3             BMW- X4             BMW- X5
##                  30                  50                  21                  42
##             BMW- X6             BMW- X7             BMW- Z3             BMW- Z4
##                   7                   6                   2                   8
##   Mercedes- A Class   Mercedes- B Class   Mercedes- C Class  Mercedes- CL Class
##                 266                  60                 395                  57
## Mercedes- CLA Class Mercedes- CLS Class   Mercedes- E Class  Mercedes- GL Class
##                   7                  25                 199                  12
## Mercedes- GLA Class Mercedes- GLB Class Mercedes- GLC Class Mercedes- GLE Class
##                  69                   1                  82                  39
## Mercedes- GLS Class   Mercedes- M Class   Mercedes- S Class  Mercedes- SL CLASS
##                   6                   9                  21                  29
##       Mercedes- SLK   Mercedes- V Class   Mercedes- X-CLASS        Mercedes-180
##                  10                  23                  10                   1
##          VW- Amarok          VW- Arteon          VW- Beetle           VW- Caddy
##                  10                  25                   4                   1
##      VW- Caddy Life VW- Caddy Maxi Life      VW- California       VW- Caravelle
##                   1                   4                   2                   9
##              VW- CC             VW- Eos             VW- Fox            VW- Golf
##                   8                   1                   1                 488
##         VW- Golf SV           VW- Jetta          VW- Passat            VW- Polo
##                  21                   1                  89                 330
##        VW- Scirocco          VW- Sharan         VW- Shuttle         VW- T-Cross
##                  27                  25                   6                  22
##           VW- T-Roc          VW- Tiguan VW- Tiguan Allspace         VW- Touareg
##                  64                 184                  12                  39
##          VW- Touran              VW- Up
##                  32                 100
```

```
# Too many models to represent them in a graph The is not missing
# data or erroneous data, so we will not make any change in the model
# column
```

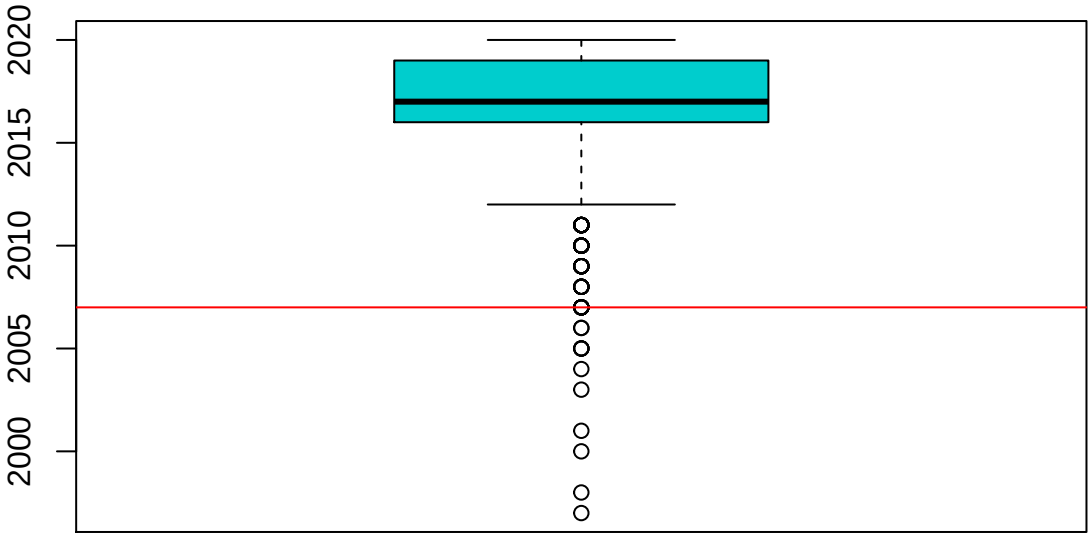
2.3.2 Year

A discrete numeric variable to indicate the year the car was sold, ranging from 1970 to 2020

```
boxplot(df$year, main = "Boxplot of sold year", col = "cyan3")
# df$year <- factor(df$year) We can see that there are outliers in
# the dataset, so we will treat them.
summary(df$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1997    2016    2017    2017    2019    2020
```

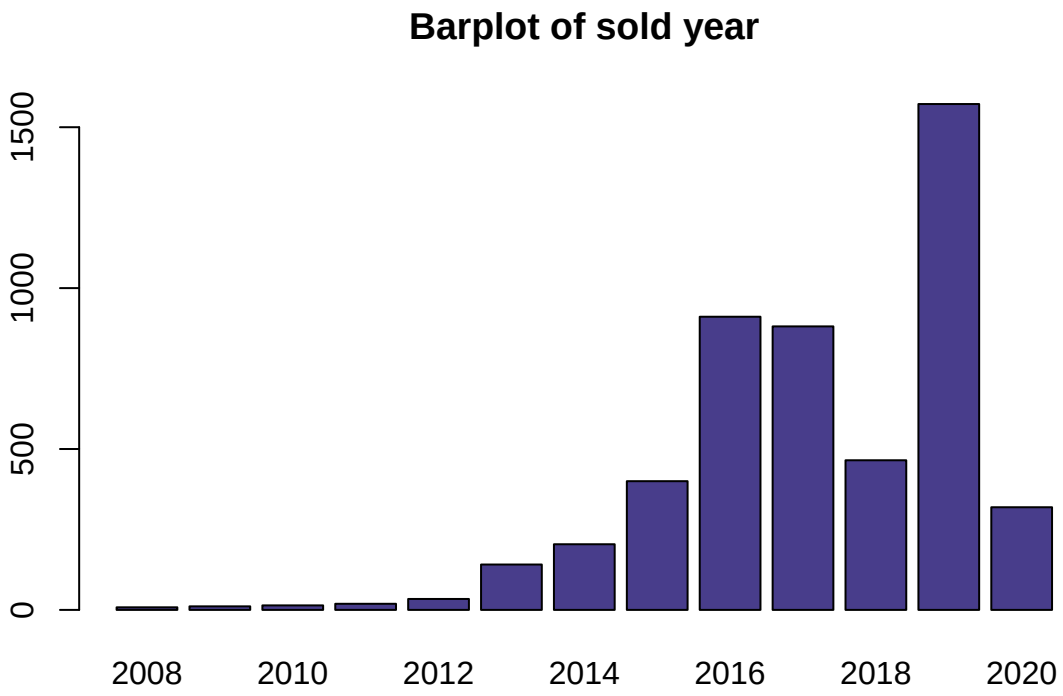
```
var_out <- calcQ(df$year)
abline(h = var_out$souts, col = "red")
abline(h = var_out$souti, col = "red")
```




```
llout <- which((df$year <= var_out$souti))
iouts[llout] <- iouts[llout] + 1
jouts[which(colnames(df) == "year")] <- length(llout)

# We will group all the inferior outliers into one variable
df[llout, "year"] <- NA
summary(df$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2008    2016    2017    2017    2019    2020      21
```

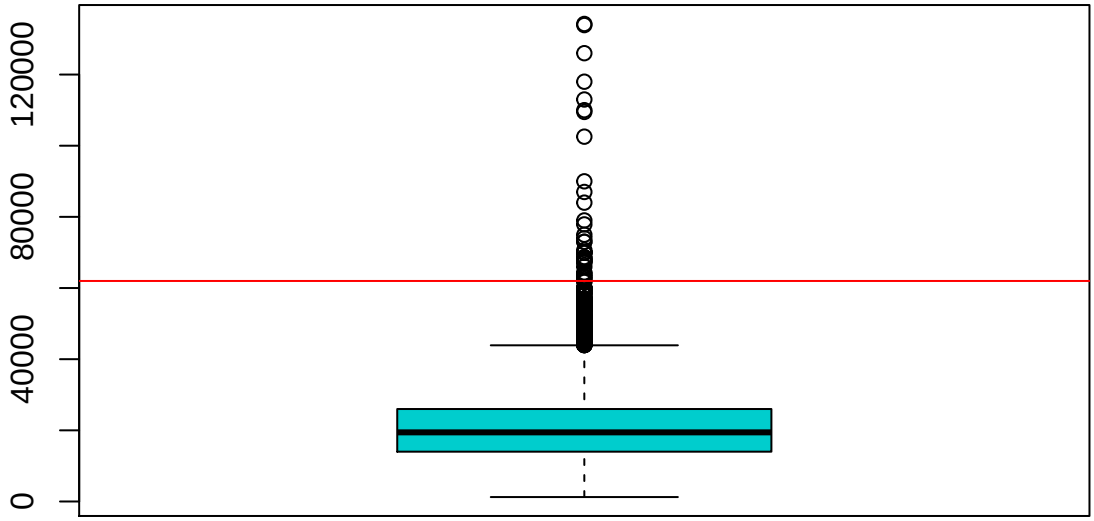
```
# df[which(df$year<=var_out$souti),'year'] <- paste0(var_out$souti, '
# or before')
barplot(table(df$year), main = "Barplot of sold year", col = "darkslateblue")
```



2.3.3 Price

In orther to better analyze the price of the cars and to group them, we will create a categorical variable representing the price of the car.

```
df$f.price <- factor(cut(df$price/1000, breaks = c(quantile(df$price/1000,
    seq(0, 1, 0.2), na.rm = TRUE)), labels = c("super cheap", "cheap",
    "expensive", "very expensive", "extremely expensive"), include.lowest = T))

sel <- which(df$price <= 0)
jerrs[which(colnames(df) == "price")] <- length(sel)

df[which(df$price < 0), ] <- NA

boxplot(df$price, main = "Boxplot of price", col = "cyan3")
var_out <- calcQ(df$price)
abline(h = var_out$souts, col = "red")
abline(h = var_out$souti, col = "red")
```




```
llout_price <- which((df$price > var_out$souts) | (df$price < var_out$souti))
jouts[which(colnames(df) == "price")] <- length(llout_price)
iouts[llout_price] <- iouts[llout_price] + 1
```

2.3.4 Transmission

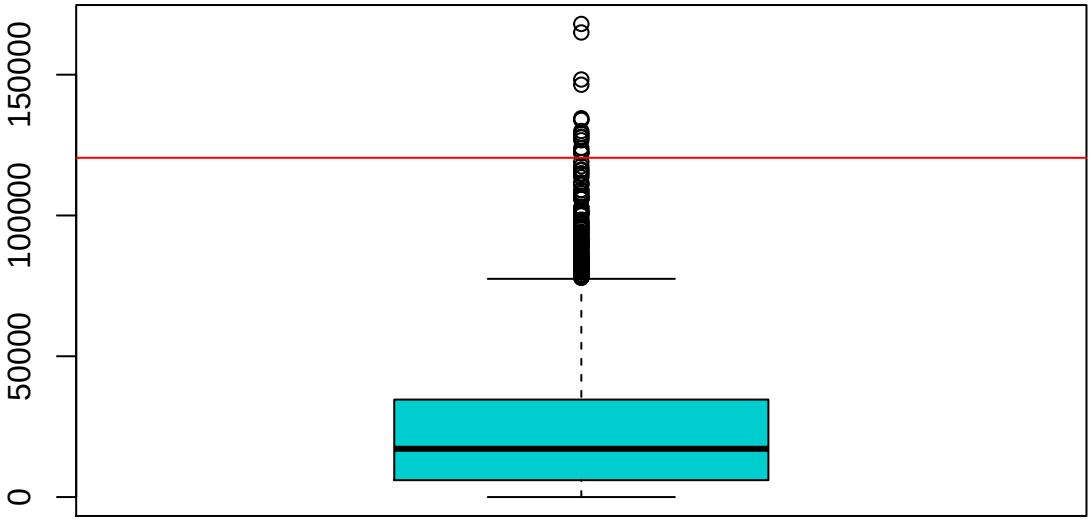
```
df$transmission <- factor(df$transmission)
levels(df$transmission)
```

```
## [1] "Automatic" "Manual"    "Other"     "Semi-Auto"
```

```
df$transmission <- factor(df$transmission, levels = c("Manual", "Semi-Auto",
    "Automatic"), labels = paste0("f.Trans-", c("Manual", "SemiAuto", "Automatic")))
# All transmission not listed above have been replaced as NA
```

2.3.5 Mileage

```
boxplot(df$mileage, col = "cyan3")
var_out <- calcQ(df$mileage)
abline(h = var_out$souts, col = "red")
abline(h = var_out$souti, col = "red")
```




```
llout_mil <- which((df$mileage < var_out$souti) | (df$mileage > var_out$souts))
iouts[llout_mil] <- iouts[llout_mil] + 1
df[llout_mil, "mileage"] <- NA
```

2.3.6 fuelType

```
df$fuelType <- factor(df$fuelType)
levels(df$fuelType)
```

```
## [1] "Diesel" "Hybrid" "Other"  "Petrol"
```

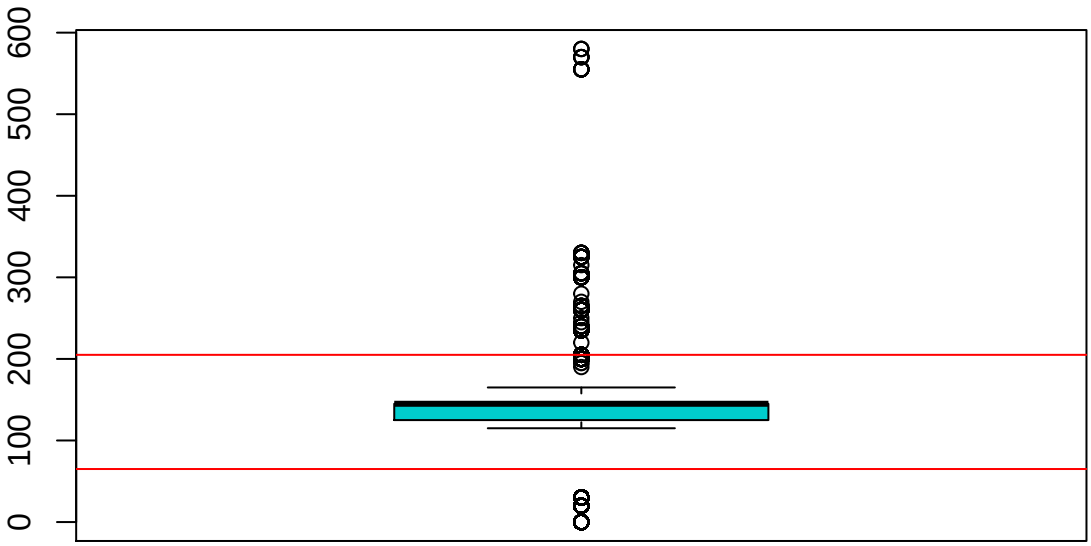
```
df$fuelType <- factor(df$fuelType, levels = c("Diesel", "Petrol", "Hybrid"),
    labels = paste0("f.Fuel-", c("Diesel", "Petrol", "Hybrid")))
# All fuelTypes not listed above have been replaced as NA
```

2.3.7 Tax

```
boxplot(df$tax, main = "Boxplot of tax", col = "cyan3")
# df$year <- factor(df$year) We can see that there are outliers in
# the dataset, so we will treat them.
summary(df$tax)
```

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

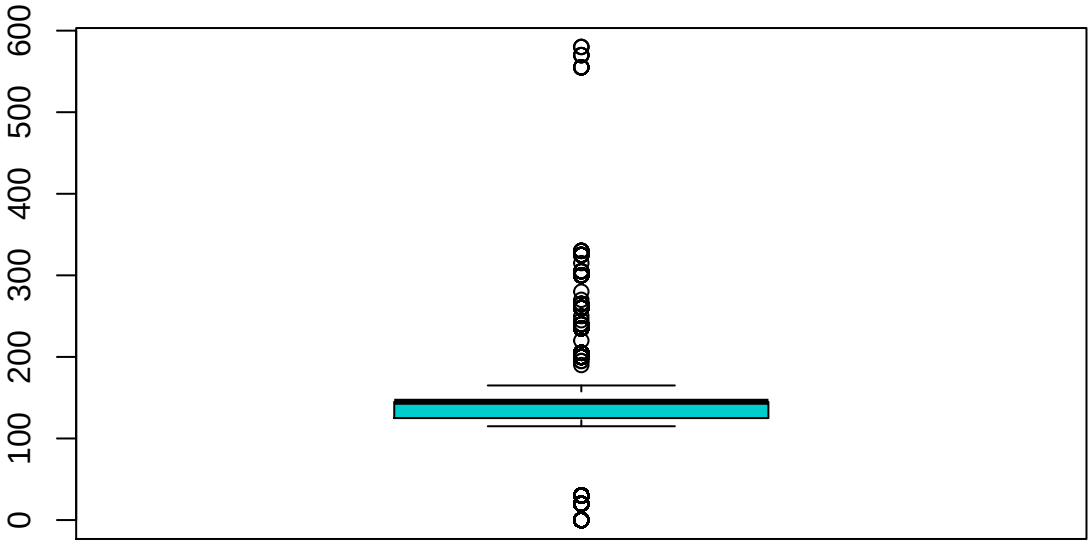
```
var_out <- calcQ(df$tax)
abline(h = var_out$souts, col = "red")
abline(h = var_out$souti, col = "red")
```




```
llout <- which((df$tax <= var_out$souti & df$tax >= var_out$souts))
iouts[llout] <- iouts[llout] + 1
jouts[which(colnames(df) == "tax")] <- length(llout)
df[llout, "tax"] <- NA

summary(df$tax)
```

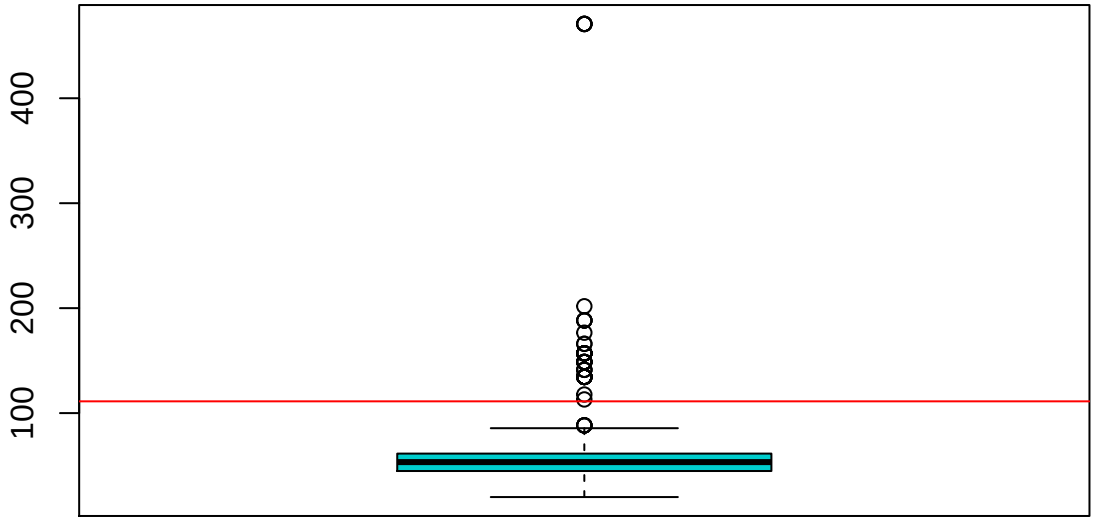
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   125.0   145.0   122.9   145.0   580.0
```

```
boxplot(df$tax, col = "cyan3")
```



2.3.8 MPG

```
# Outliers are replaced by NA
boxplot(df$mpg, col = "cyan3")
var_out <- calcQ(df$mpg)
abline(h = var_out$souts, col = "red")
abline(h = var_out$souti, col = "red")
```




```
llout_mpg <- which((df$mpg < var_out$souti) | (df$mpg > var_out$souts))
iouts[llout_mpg] <- iouts[llout_mpg] + 1
jouts[which(colnames(df) == "mpg")] <- length(llout)
df[llout_mpg, "mpg"] <- NA
```

2.3.9 EngineSyze

```
df$engineSize <- factor(df$engineSize)
levels(df$engineSize)
```

```
##  [1] "0"   "1"   "1.2" "1.3" "1.4" "1.5" "1.6" "1.8" "1.9" "2"   "2.1" "2.2"
## [13] "2.3" "2.5" "2.9" "3"   "3.2" "3.5" "3.7" "4"   "4.1" "4.2" "4.4" "4.7"
## [25] "5.2" "5.5" "6.2" "6.6"
```

```
df[which(df[, "engineSize"] == 0), ]
```

```
##                 model year price      transmission mileage      fuelType tax
## 777           Audi- Q3 2020 33333 f.Trans-Automatic    1500 f.Fuel-Diesel 145
## 789           Audi- Q2 2020 24990    f.Trans-Manual    1500 f.Fuel-Petrol 145
## 795          Audi- SQ5 2020 56450 f.Trans-Automatic    1500 f.Fuel-Diesel 145
## 796           Audi- Q3 2020 33990 f.Trans-Automatic    4000 f.Fuel-Diesel 145
## 812           Audi- Q3 2017 19300    f.Trans-Manual   16051 f.Fuel-Diesel 150
## 815           Audi- TT 2016 22500 f.Trans-Automatic   45182 f.Fuel-Petrol 200
## 821           Audi- Q3 2020 32000 f.Trans-Automatic    1500 f.Fuel-Petrol 145
## 1356           BMW- i3 2016 19490 f.Trans-Automatic    8421 f.Fuel-Hybrid   0
## 1450           BMW- i3 2016 16482 f.Trans-Automatic   43695 f.Fuel-Hybrid   0
## 1687           BMW- i3 2014 14182 f.Trans-Automatic   37161 f.Fuel-Hybrid   0
## 1710           BMW- i3 2017 23751 f.Trans-Automatic   28169 f.Fuel-Hybrid   0
## 1803           BMW- i3 2017 19948 f.Trans-Automatic   20929 f.Fuel-Hybrid 135
## 3144 Mercedes- A Class 2016 17800 f.Trans-Automatic   21913 f.Fuel-Diesel  20
## 3302 Mercedes- E Class 2018 22738 f.Trans-Automatic   24000 f.Fuel-Diesel 150
## 3552         VW- T-Roc 2019 22000 f.Trans-Automatic    2009 f.Fuel-Petrol 145
```

```
## 3965          VW- Golf 2017 12600    f.Trans-Manual   20340 f.Fuel-Diesel   0
##       mpg engineSize manufacturer             f.price
## 777  47.1          0         Audi extremely expensive
## 789  43.5          0         Audi      very expensive
## 795  34.5          0         Audi extremely expensive
## 796  47.1          0         Audi extremely expensive
## 812  52.3          0         Audi           expensive
## 815  40.9          0         Audi      very expensive
## 821  31.4          0         Audi extremely expensive
## 1356   NA          0          BMW           expensive
## 1450   NA          0          BMW               cheap
## 1687   NA          0          BMW               cheap
## 1710   NA          0          BMW      very expensive
## 1803   NA          0          BMW           expensive
## 3144 68.9          0     Mercedes           expensive
## 3302 61.4          0     Mercedes      very expensive
## 3552 39.8          0           VW      very expensive
## 3965 74.3          0           VW         super cheap
```

```
# It is a quantitive variable Non-possible values will be recoded to
# NA
sel <- which(df$engineSize == 0)
ierrs[sel] <- ierrs[sel] + 1  #Vector of errors per individual update
sel  #### sel contains the rownames of the individuals with 'O'
```

```
##  [1]  777  789  795  796  812  815  821 1356 1450 1687 1710 1803 3144 3302 3552
## [16] 3965
```

```
# as value for engineSize We should update jerrs vector: errors per
# variable

# df[sel,'engineSize']<-3 # non-possible values are replaced by NA,
# missing value symbol in R NA assignment for forward imputation:
df[sel, "engineSize"] <- NA
```

3 Imputation

What we do with imputation is be able to eliminate all those values that may be missings, outliers or errors to turn them into values that can be realistic within our sample.

3.1 Imputation of numeric variables

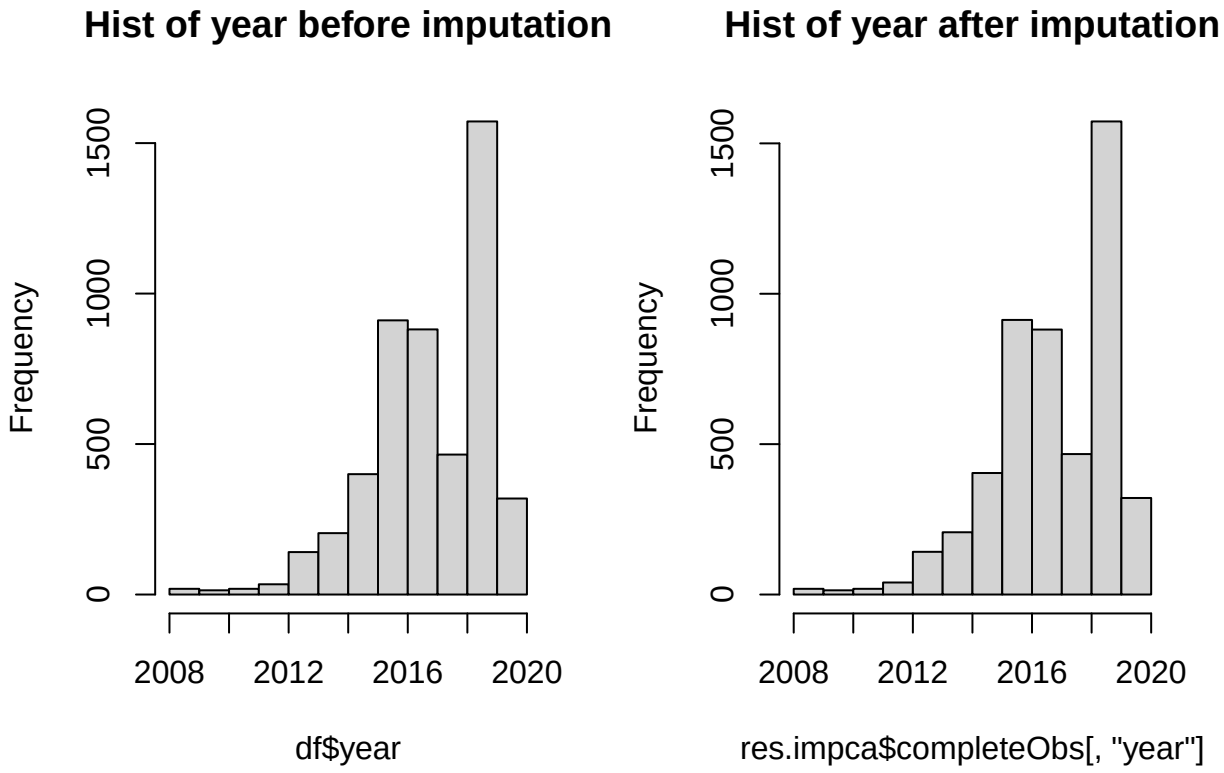
```
library(missMDA)
# Now one by one describe vars and put them on lists
vars_con <- c("year", "mileage", "tax", "mpg")
vars_res <- c("price")

summary(df[, vars_con])
```

```
##       year         mileage            tax             mpg
##  Min.   :2008   Min.   :     1   Min.   :  0.0   Min.   :20.00
##  1st Qu.:2016   1st Qu.:  5986   1st Qu.:125.0   1st Qu.:44.80
##  Median :2017   Median : 17007   Median :145.0   Median :53.30
##  Mean   :2017   Mean   : 23228   Mean   :122.9   Mean   :52.93
##  3rd Qu.:2019   3rd Qu.: 34403   3rd Qu.:145.0   3rd Qu.:61.40
##  Max.   :2020   Max.   :119000   Max.   :580.0   Max.   :88.30
##  NA's   :21     NA's   :17                       NA's   :54
```

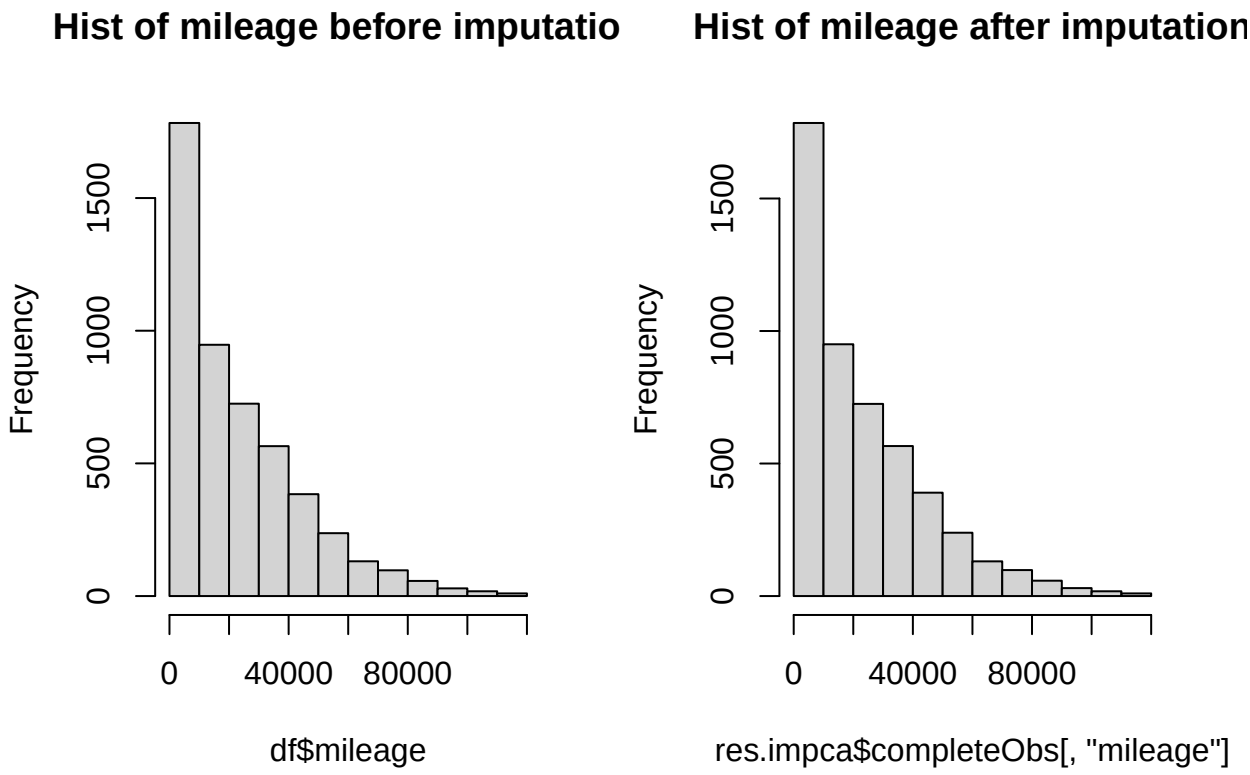
```
res.impca <- imputePCA(df[, vars_con], ncp = 3)
summary(res.impca$completeObs)
```

```
##       year          mileage            tax             mpg
##  Min.   :2008   Min.   :     1   Min.   :  0.0   Min.   :20.0
##  1st Qu.:2016   1st Qu.:  5991   1st Qu.:125.0   1st Qu.:44.8
##  Median :2017   Median : 17065   Median :145.0   Median :53.3
##  Mean   :2017   Mean   : 23290   Mean   :122.9   Mean   :53.0
##  3rd Qu.:2019   3rd Qu.: 34514   3rd Qu.:145.0   3rd Qu.:61.4
##  Max.   :2020   Max.   :119000   Max.   :580.0   Max.   :88.3
```

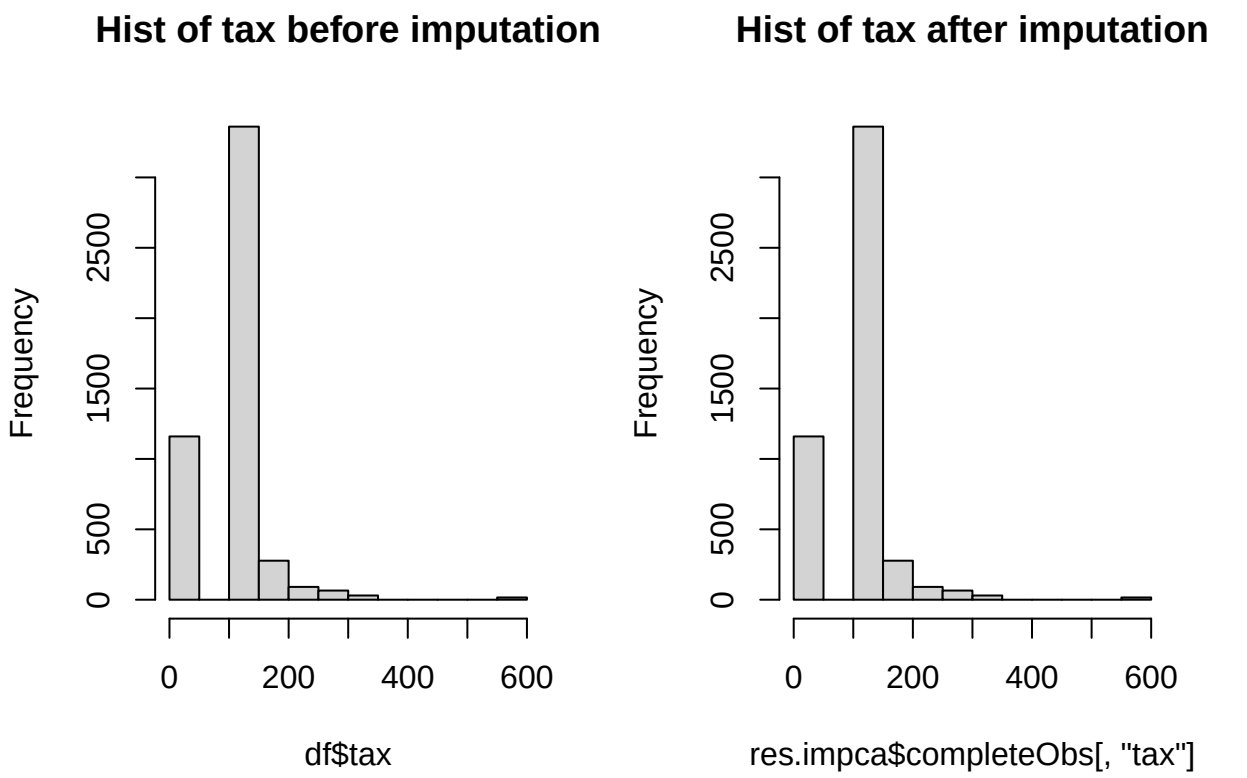
```
# Check one by one:
par(mfrow = c(1, 2))
hist(df$year, main = "Hist of year before imputation")
hist(res.impca$completeObs[, "year"], main = "Hist of year after imputation")
```



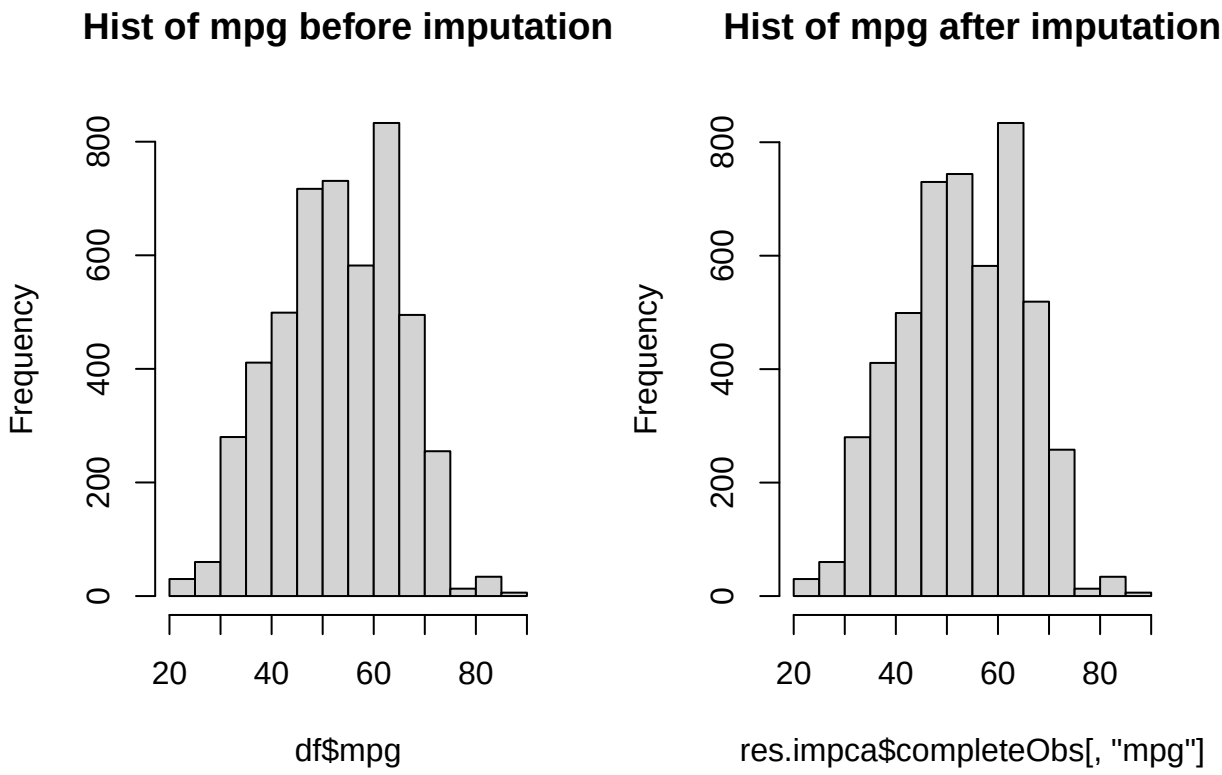

```
hist(df$mileage, main = "Hist of mileage before imputation")
hist(res.impca$completeObs[, "mileage"], main = "Hist of mileage after imputation")
```




```r
hist(df$tax, main = "Hist of tax before imputation")
hist(res.impca$completeObs[, "tax"], main = "Hist of tax after imputation")
```




```r
hist(df$mpg, main = "Hist of mpg before imputation")
hist(res.impca$completeObs[, "mpg"], main = "Hist of mpg after imputation")
```




```
# Once you have validated the process:
df[, vars_con] <- res.impca$completeObs
```

3.2 Imputation of qualitative variables

```
vars_dis <- c("model", "transmission", "fuelType", "engineSize", "manufacturer")
summary(df[, vars_dis])
```

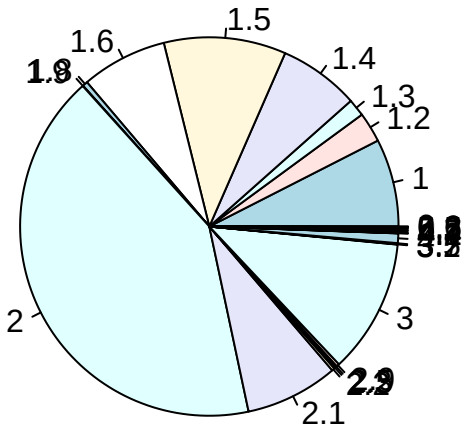
```
##                model                 transmission           fuelType
##  VW- Golf          : 488   f.Trans-Manual    :1741   f.Fuel-Diesel:2851
##  Mercedes- C Class: 395   f.Trans-SemiAuto :1938   f.Fuel-Petrol:2070
##  VW- Polo          : 330   f.Trans-Automatic:1320   f.Fuel-Hybrid:  66
##  Mercedes- A Class: 266   NA's             :   1   NA's          :  13
##  BMW- 3 Series     : 251
##  Mercedes- E Class: 199
##  (Other)           :3071
##    engineSize   manufacturer
##  2      :2076   Length:5000
##  3      : 571   Class :character
##  1.5    : 520   Mode  :character
##  2.1    : 395
##  1      : 374
##  (Other):1048
##  NA's   :  16
```

```
res.immca <- imputeMCA(df[, vars_dis], ncp = 4)
summary(res.immca$completeObs)
```

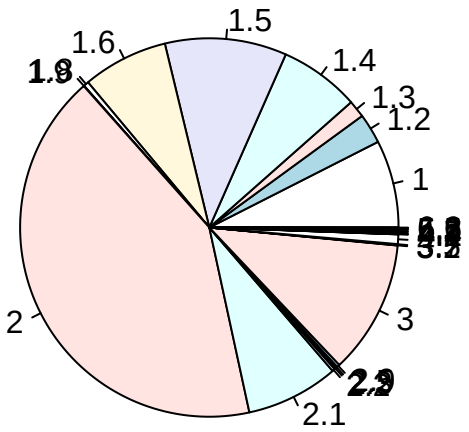
```
##                model                 transmission           fuelType
##  VW- Golf          : 488   f.Trans-Manual    :1741   f.Fuel-Diesel:2860
##  Mercedes- C Class: 395   f.Trans-SemiAuto :1939   f.Fuel-Petrol:2074
##  VW- Polo          : 330   f.Trans-Automatic:1320   f.Fuel-Hybrid:  66
```

```
##  Mercedes- A Class: 266
##  BMW- 3 Series    : 251
##  Mercedes- E Class: 199
##  (Other)          :3071
##    engineSize     manufacturer
##  2      :2092   Audi    :1089
##  3      : 571   BMW     :1084
##  1.5    : 520   Mercedes:1321
##  2.1    : 395   VW      :1506
##  1      : 374
##  1.6    : 365
##  (Other): 683
```

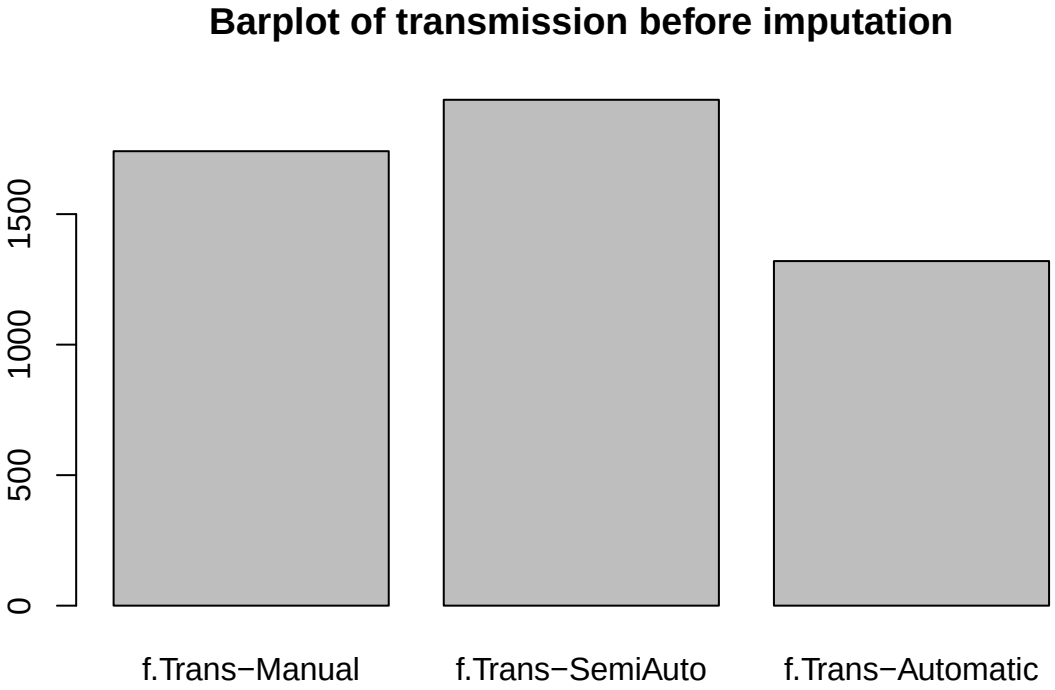
```
# Check one by one (we only have enginesize, transmission & fuelType)
pie(table(df$engineSize), main = "Piechart of engineSize before imputation")
```



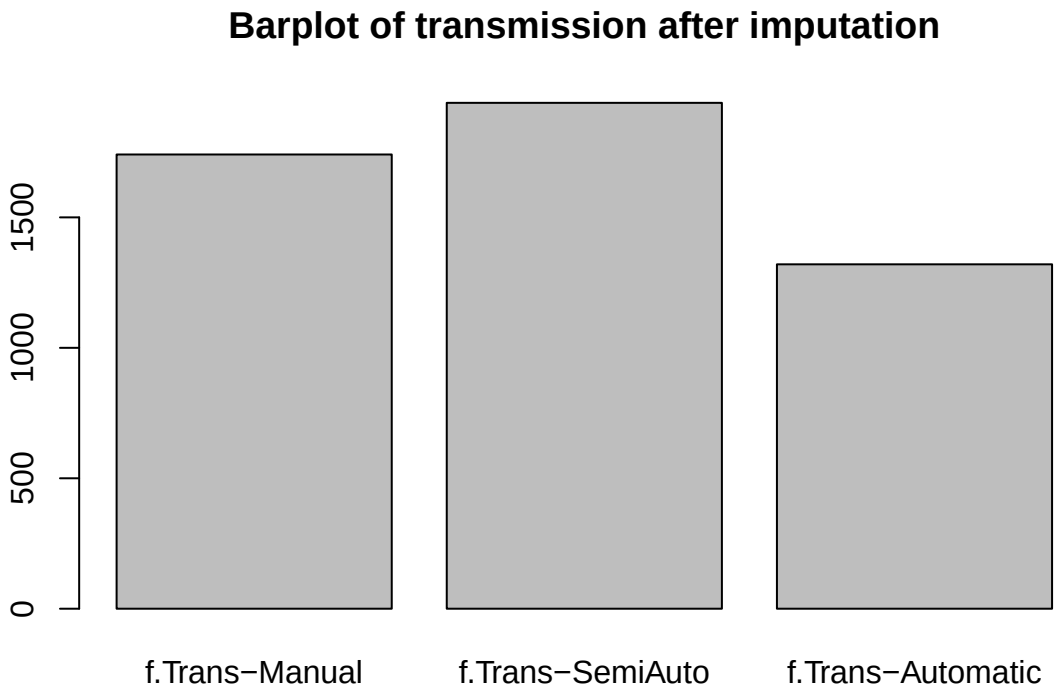

```
pie(table(res.immca$completeObs[, "engineSize"]), main = "Piechart of engineSize after imputation")
```



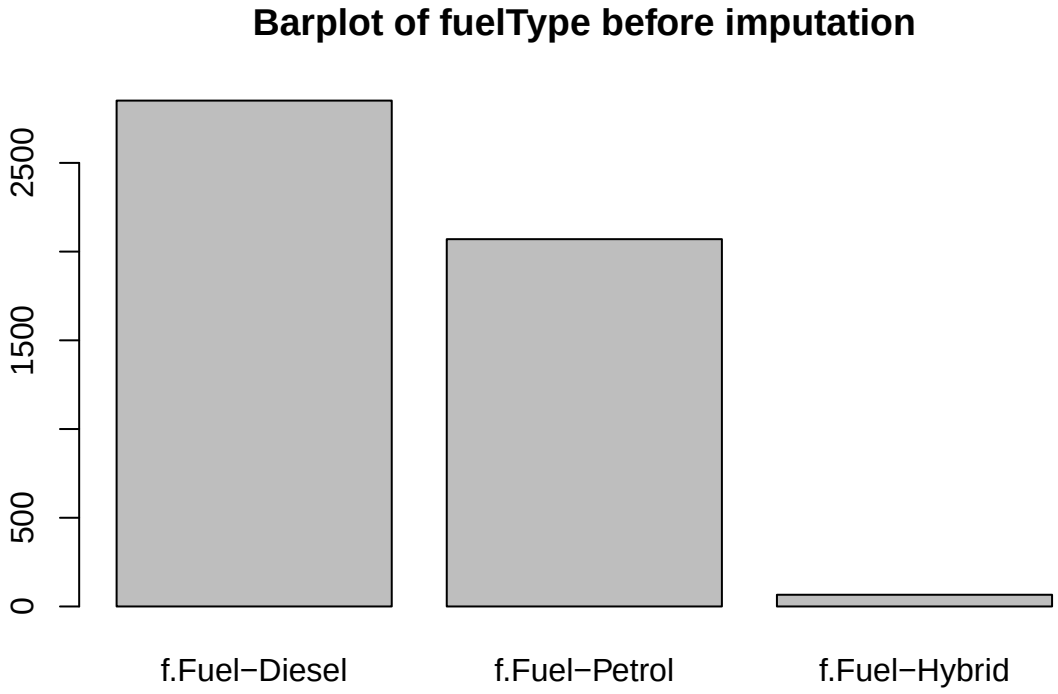

```r
barplot(table(df$transmission), main = "Barplot of transmission before imputation")
```



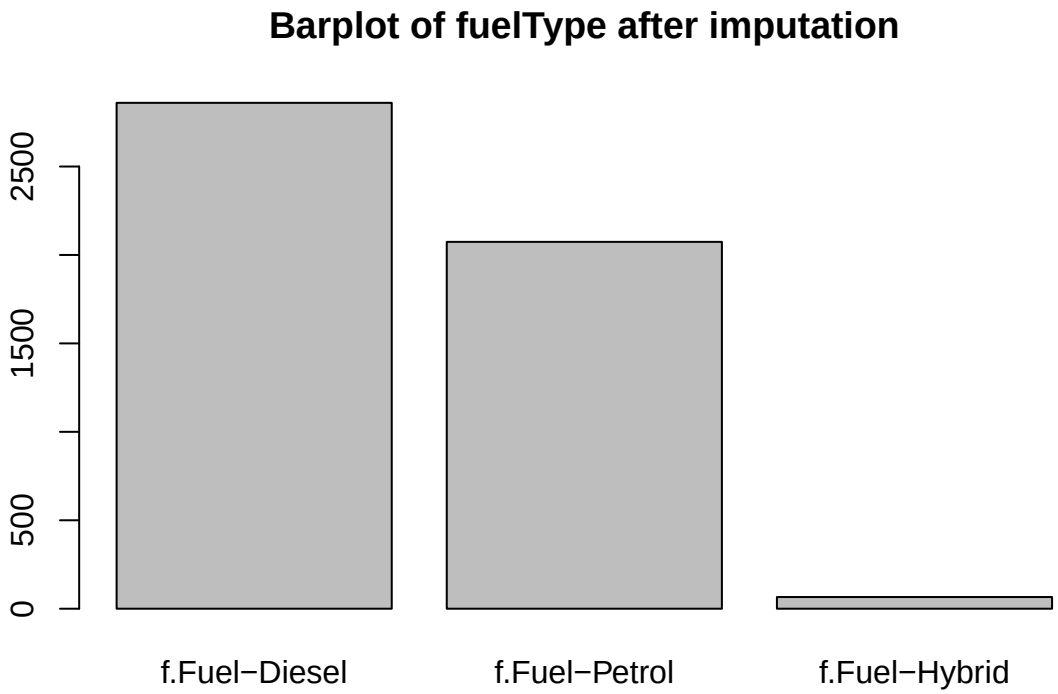
```r
barplot(table(res.immca$completeObs[, "transmission"]), main = "Barplot of transmission after imputation
```




```r
barplot(table(df$fuelType), main = "Barplot of fuelType before imputation")
```



```r
barplot(table(res.immca$completeObs[, "fuelType"]), main = "Barplot of fuelType after imputation")
```




```r
# Once you have validated the process
df[, vars_dis] <- res.immca$completeObs

# Are there NA?
sum(countNA(df)$mis_ind) == 0
```

```
## [1] TRUE
```

```r
par(mfrow = c(1, 1))
```

4 Creation and discretization of new variables

4.1 New variable: Audi/Not Audi

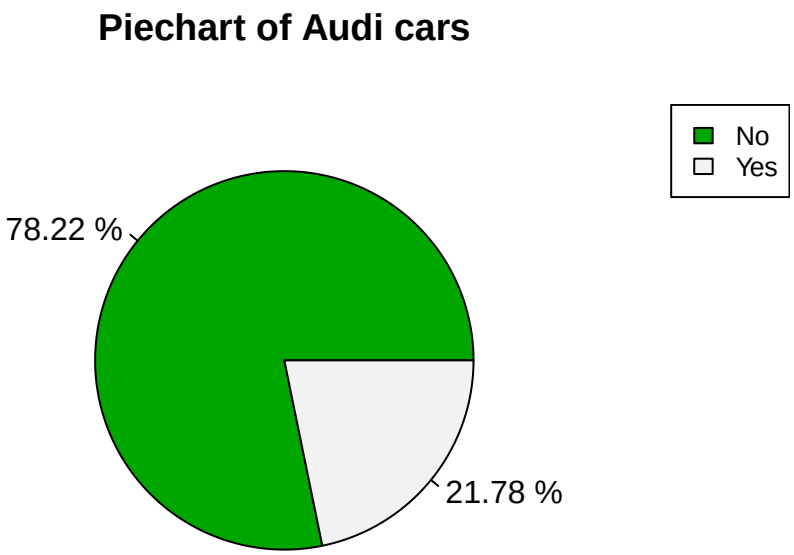
```r
# Binary Target: Audi?

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

```
##   No  Yes
## 3911 1089
```

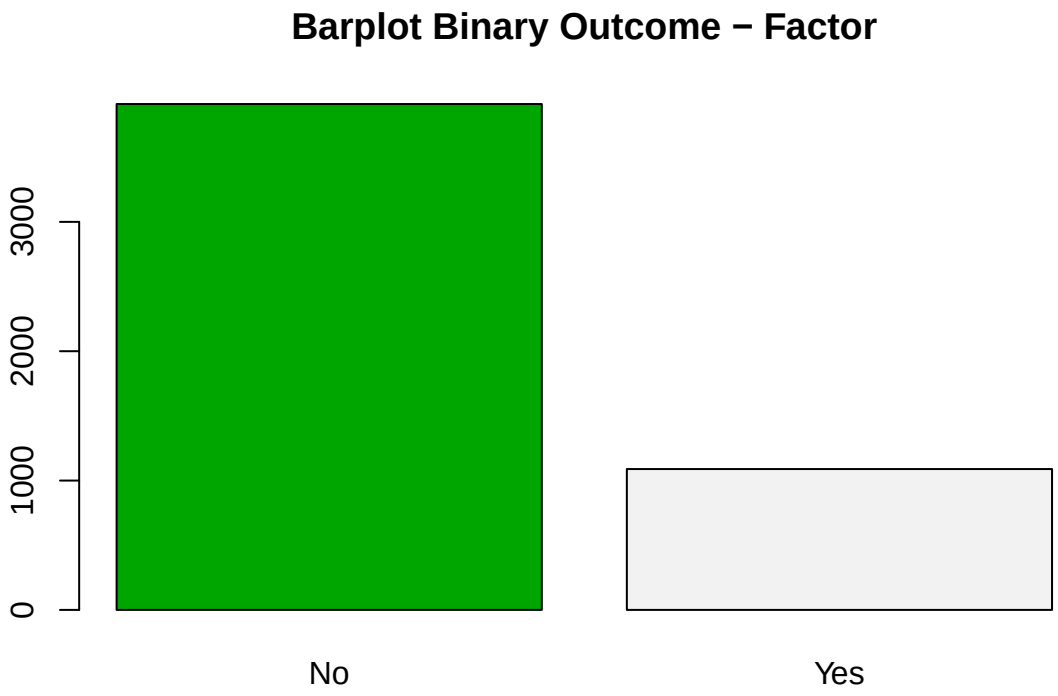
```r
# Pie
piepercent <- round(100 * (table(df$Audi)/nrow(df)), dig = 2)
piepercent
```

```
##
##    No   Yes
## 78.22 21.78
```

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



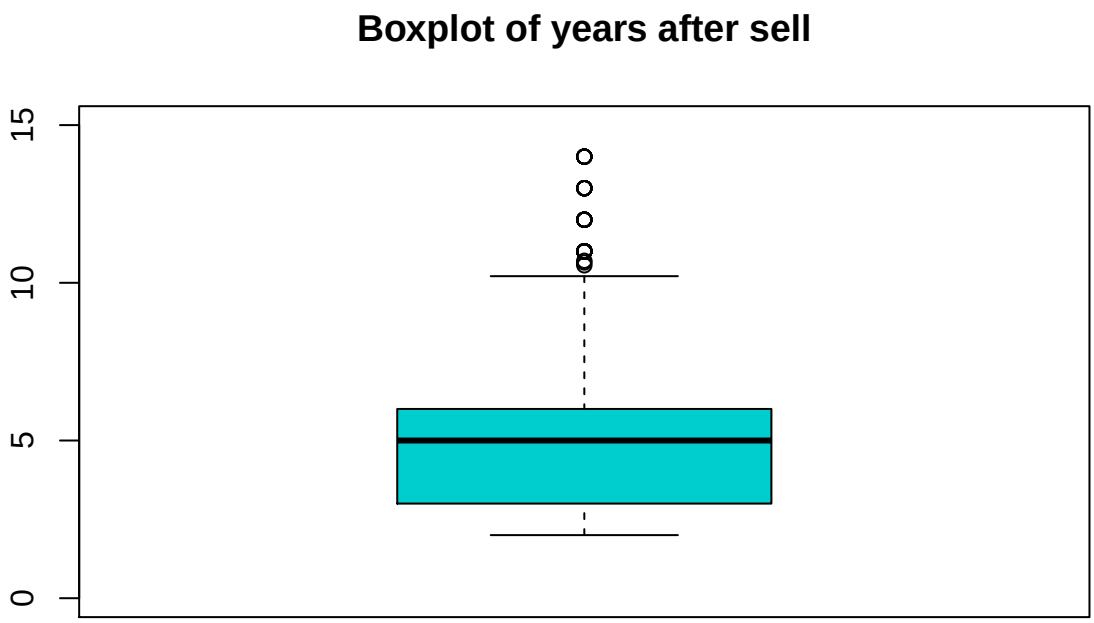

```
# Bar Chart
barplot(table(df$Audi), main = "Barplot Binary Outcome - Factor", col = terrain.colors(2))
```



4.2 New variable: yearsAferSell

A discrete numeric variable to indicate how many years have passed from when the car was sold since 2022.

```
df$years_after_sell <- 2022 - df$year
boxplot(df$years_after_sell, main = "Boxplot of years after sell", col = "cyan3",
    ylim = c(0, 15))
```



```
# There are no extreme outliers in the variable because we treated
# outliers in the variable year.
summary(df$years_after_sell)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   5.000   4.784   6.000  14.000
```

4.3 Discretization of the variable Tax

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

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

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

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    0   20   30  145  145  145  145  145  145  150  580
```

```
quants <- calcQ(df$tax)

# df$aux<-factor(cut(df$tax,breaks=quantile(df$tax,seq(0,1,0.25),na.rm=TRUE),include.lowest
# = T )) # Does not work Reconsiderations of limits bc mean and 3rd
# quantile are the same aux<-factor(cut(df$tax,breaks=c(0, 125, 145,
# quants),include.lowest = T )) summary(aux)
# tapply(df$tax,aux,median)
df$f.tax <- factor(cut(df$tax, breaks = c(quants$min, quants$q1, quants$q2,
    quants$q3 + 10, quants$max), include.lowest = T))
levels(df$f.tax) <- paste("f.tax-", levels(df$f.tax), sep = "")
table(df$f.tax, useNA = "always")
```

```
##
##   f.tax-[0,125] f.tax-(125,145] f.tax-(145,155] f.tax-(155,580]            <NA>
##            1447            2568             507             478               0
```

4.4 Discretization of the variable mileage

```
df$f.mileage <- factor(cut(df$mileage, breaks = c(quantile(df$mileage,
    seq(0, 1, 0.25), na.rm = TRUE)), include.lowest = T))
levels(df$f.mileage) <- paste("f.mileage-", levels(df$f.mileage), sep = "")
table(df$f.mileage, useNA = "always")
```

```
##
##        f.mileage-[1,5.99e+03] f.mileage-(5.99e+03,1.71e+04]
##                          1250                           1250
## f.mileage-(1.71e+04,3.45e+04] f.mileage-(3.45e+04,1.19e+05]
##                          1250                           1250
##                          <NA>
##                             0
```

4.5 Discretization of the variable mpg

```
df$f.mpg <- factor(cut(df$mpg, breaks = c(quantile(df$mpg, seq(0, 1, 0.25),
    na.rm = TRUE)), include.lowest = T))
levels(df$f.mpg) <- paste("f.mpg-", levels(df$f.mpg), sep = "")
table(df$f.mpg, useNA = "always")
```

```
##
##   f.mpg-[20,44.8] f.mpg-(44.8,53.3] f.mpg-(53.3,61.4] f.mpg-(61.4,88.3]
##              1280              1328              1208              1184
##              <NA>
##                 0
```

4.6 Discretization of the variable year

```
df$f.year <- factor(cut(df$year, breaks = c(quantile(df$year, seq(0, 1,
    0.25), na.rm = TRUE)), include.lowest = T))
levels(df$f.year) <- paste("f.mpg-", levels(df$f.year), sep = "")
table(df$f.year, useNA = "always")
```

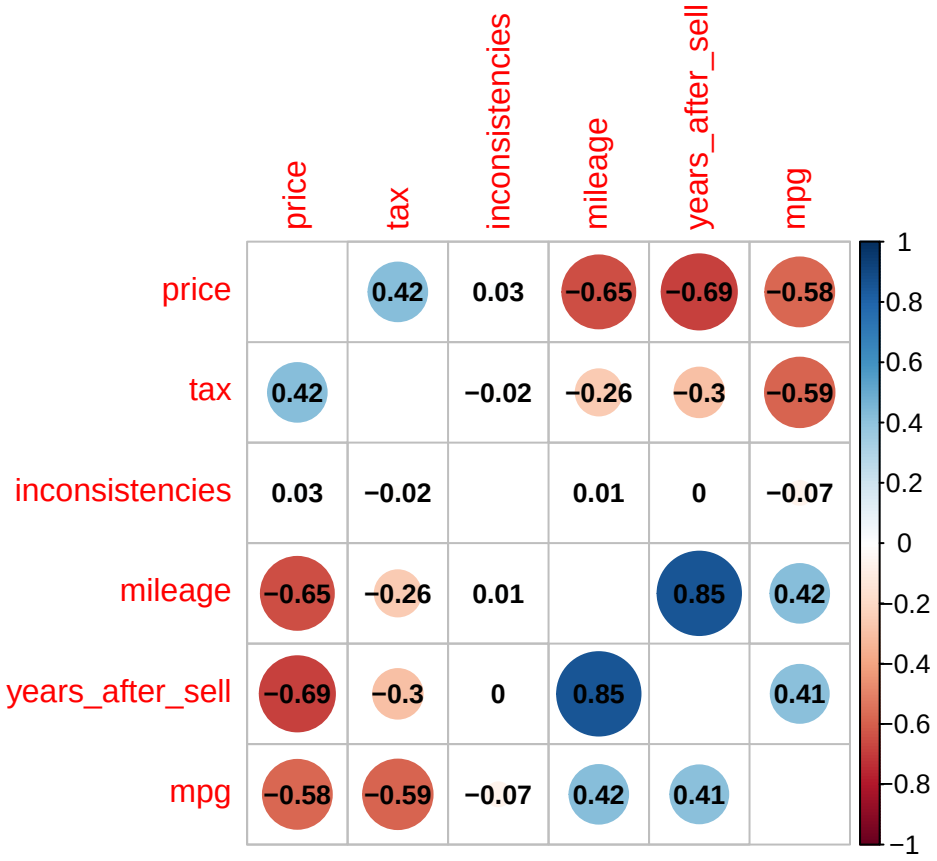
```
##
## f.mpg-[2008,2016] f.mpg-(2016,2017] f.mpg-(2017,2019] f.mpg-(2019,2020]
##              1758               881              2040               321
##              <NA>
##                 0
```

5 Create variable adding the total number missing values, outliers and errors.

Describe these variables, to which other variables exist higher associations.

5.1 Compute the correlation with all other variables. Rank these variables according the correlation

```
df$inconsistencies <- imis + iouts + ierrs
vars_con <- c(3, 5, 7, 8, 13, 18)
M = round(cor(df[, c(vars_con)], method = "spearman"), dig = 2)
corrplot(M, method = "circle", insig = "blank", addCoef.col = "black",
    number.cex = 0.8, order = "AOE", diag = FALSE)
```



The variable year is correlated negatively with the variables years_after_sell and mileage, so it indicates that cars are cheaper when they are older or they are more used.

Variable tax is positively correlated to the variable price and negatively with the variable mpg. Thus a car will have more taxes if it is expensive and less if spends low mpg.

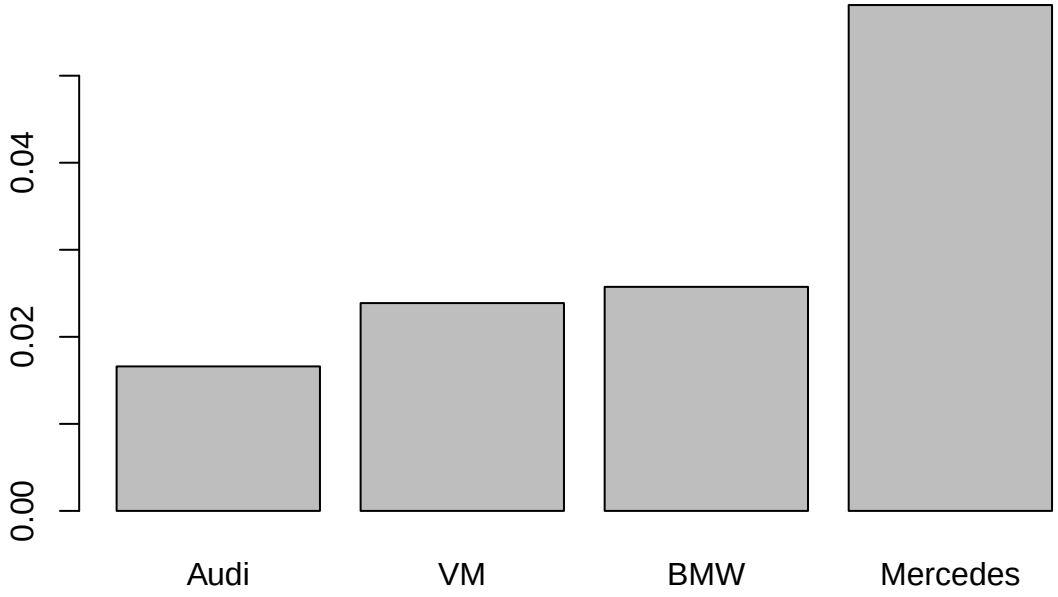
Inconsistencies is not correlated with any variable.

The variable mpg is negatively correlated with price and tax, as we had seen. Moreover, is positively correlated with the variables mileage and years_after_sell. We can deduce that older cars and more used has a lower consume.

5.2 Mean of missing/outliers/errors per groups

Compute for every group of individuals (group of age, etc, ...) the mean of missing/outliers/errors values. Rank the groups according the computed mean.

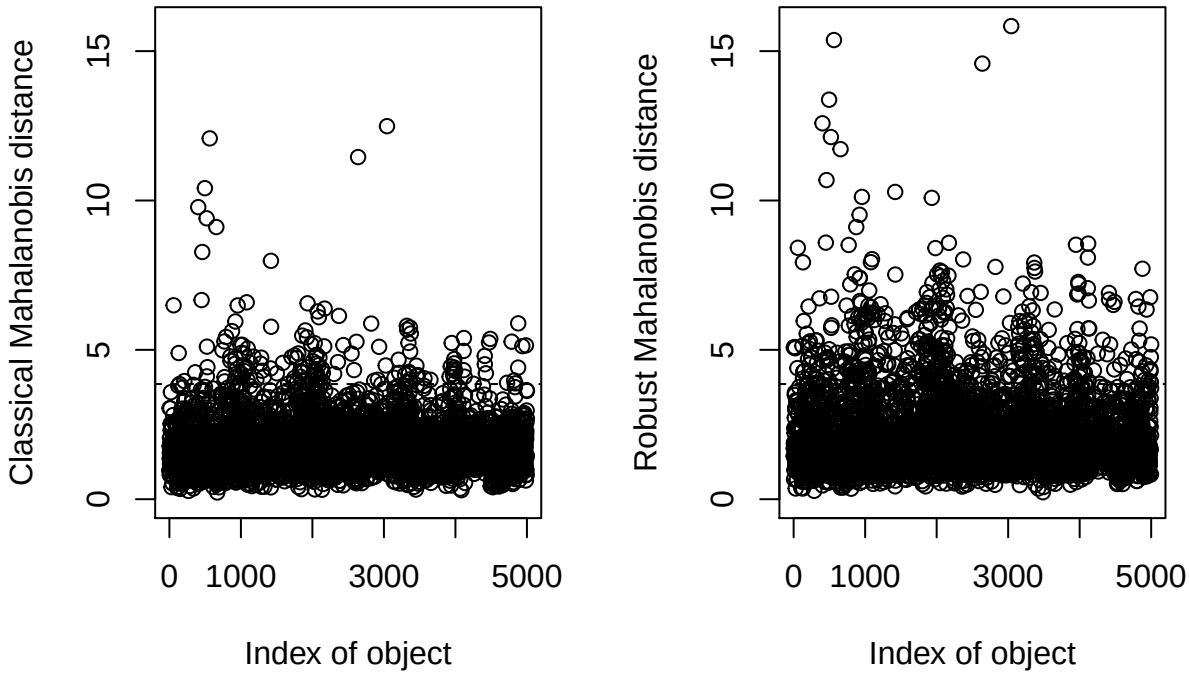
```
dfInconsistencias <- data.frame(Modelo = c("Audi", "VM", "Mercedes", "BMW"))
vw_inconsis <- mean(df$inconsistencies[which(df$manufacturer == "VW")])
audi_inconsis <- mean(df$inconsistencies[which(df$manufacturer == "Audi")])
bmw_inconsis <- mean(df$inconsistencies[which(df$manufacturer == "BMW")])
merc_inconsis <- mean(df$inconsistencies[which(df$manufacturer == "Mercedes")])
dfInconsistencias$incons <- c(vw_inconsis, audi_inconsis, bmw_inconsis,
    merc_inconsis)
dfInconsistencias <- dfInconsistencias[order(dfInconsistencias$incons),
    ]
barplot(dfInconsistencias$incons, names.arg = dfInconsistencias$Modelo)
```



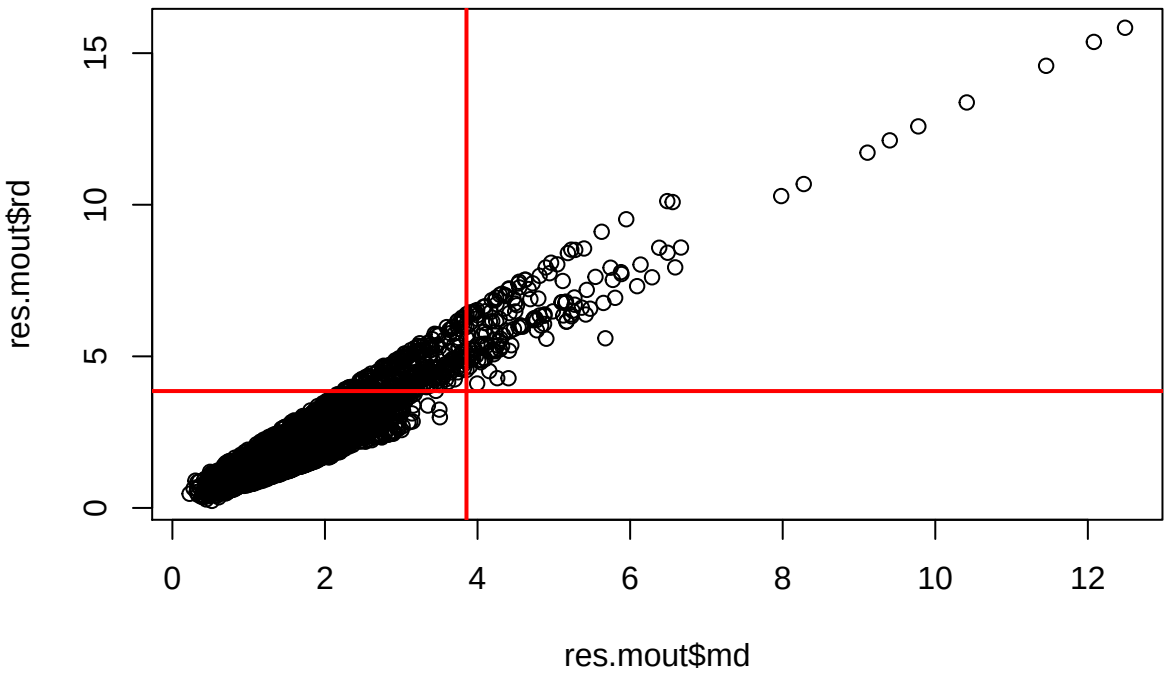
6 Multivariant outliers

We don't use the variable tax for the searching of multivariant outliers because it is a column linearly dependent with other column.

```
res.mout <- Moutlier(df[, c(2, 3, 5, 8)], quantile = 0.995)
```



```
par(mfrow = c(1, 1))
plot(res.mout$md, res.mout$rd)
abline(h = res.mout$cutoff, lwd = 2, col = "red")
abline(v = res.mout$cutoff, lwd = 2, col = "red")
```




```
llmout <- which((res.mout$md > res.mout$cutoff) & (res.mout$rd > res.mout$cutoff))
llmout
```

```
##   [1]   59  130  141  209  361  403  450  460  496  521  524  525  532  564  656
##  [16]  741  770  771  773  792  852  855  876  902  921  922  926  927  940  955
##  [31]  991  992  994 1008 1009 1020 1045 1046 1049 1057 1081 1082 1094 1096 1130
##  [46] 1176 1216 1275 1284 1296 1323 1416 1420 1422 1503 1588 1661 1709 1738 1748
##  [61] 1754 1782 1785 1802 1830 1832 1842 1845 1857 1863 1864 1901 1924 1932 1942
##  [76] 1953 1983 1994 2006 2029 2031 2032 2035 2041 2046 2060 2061 2067 2068 2071
##  [91] 2074 2075 2088 2090 2098 2103 2135 2142 2146 2165 2171 2304 2357 2371 2433
## [106] 2548 2582 2616 2639 2822 2864 2915 2933 3025 3043 3148 3205 3213 3295 3317
## [121] 3321 3325 3343 3349 3351 3352 3360 3361 3365 3368 3376 3454 3455 3475 3567
## [136] 3651 3918 3935 3948 3966 3969 3977 3981 3983 3984 3990 3995 4114 4115 4119
## [151] 4129 4253 4410 4411 4426 4470 4487 4722 4786 4822 4839 4877 4878 4936 4986
```

```
df$mout <- 0
df$mout[llmout] <- 1
df$mout <- factor(df$mout, labels = c("MvOut.No", "MvOut.Yes"))
res.mout$cutoff
```

```
## [1] 3.854901
```

```
res.cat <- catdes(df[, c(2:8, 10, 18:19)], 10)
res.cat$category
```

```
## $MvOut.No
##                                 Cla/Mod  Mod/Cla Global      p.value    v.test
## transmission=f.Trans-SemiAuto  97.98865 39.29679  38.78 2.987720e-05  4.174400
```

```
## manufacturer=VW                97.94157 30.50672  30.12 8.309633e-04  3.342270
## transmission=f.Trans-Automatic 94.84848 25.89452  26.40 2.746633e-05 -4.193516
## manufacturer=BMW               94.55720 21.19959  21.68 2.603442e-05 -4.205640
##
## $MvOut.Yes
##                                 Cla/Mod  Mod/Cla Global      p.value    v.test
## manufacturer=BMW               5.442804 35.75758  21.68 2.603442e-05  4.205640
## transmission=f.Trans-Automatic 5.151515 41.21212  26.40 2.746633e-05  4.193516
## manufacturer=VW                2.058433 18.78788  30.12 8.309633e-04 -3.342270
## transmission=f.Trans-SemiAuto  2.011346 23.63636  38.78 2.987720e-05 -4.174400
```

The cars with Automatic transmission are overrepresented in multivariant outliers. And also there is a high percentage of automatic cars that are outliers (6.1%) in comparison to cars with other types of transmission. There is a relative low amount of semiautomatic cars that are outliers (20.81%) compared to the global amount of semiautomatic cars (38.45%).

```
summary(df[df$mout == "MvOut.Yes", ])
```

```
##                model          year          price
##  VW- Golf          : 13   Min.   :2008   Min.   :  1450
##  BMW- 3 Series     : 12   1st Qu.:2011   1st Qu.:  7500
##  Audi- A3          : 10   Median :2015   Median : 12500
##  Mercedes- C Class:  7   Mean   :2015   Mean   : 29100
##  Audi- A6          :  6   3rd Qu.:2018   3rd Qu.: 58990
##  Audi- R8          :  6   Max.   :2020   Max.   :134219
##  (Other)           :111
##             transmission    mileage                fuelType       tax
##  f.Trans-Manual    :58    Min.   :    10   f.Fuel-Diesel:93   Min.   :  0.0
##  f.Trans-SemiAuto :39    1st Qu.: 10000   f.Fuel-Petrol:69   1st Qu.:125.0
##  f.Trans-Automatic:68    Median : 73872   f.Fuel-Hybrid: 3   Median :145.0
##                          Mean   : 58934                      Mean   :172.2
##                          3rd Qu.: 94000                      3rd Qu.:220.0
##                          Max.   :119000                      Max.   :580.0
##
##       mpg          engineSize   manufacturer                f.price    Audi
##  Min.   :20.00   2      :58   Audi    :42    super cheap        :88   No :123
##  1st Qu.:30.10   3      :34   BMW     :59    cheap              :19   Yes: 42
##  Median :41.50   4      :18   Mercedes:33    expensive          : 4
##  Mean   :45.03   1.6    :11   VW      :31    very expensive     : 3
##  3rd Qu.:60.10   1.4    : 6                  extremely expensive:51
##  Max.   :88.30   4.4    : 6
##                  (Other):32
##  years_after_sell             f.tax                              f.mileage
##  Min.   : 2.000   f.tax-[0,125]  :49   f.mileage-[1,5.99e+03]       : 32
##  1st Qu.: 4.000   f.tax-(125,145]:45   f.mileage-(5.99e+03,1.71e+04]: 19
##  Median : 7.000   f.tax-(145,155]:14   f.mileage-(1.71e+04,3.45e+04]:  3
##  Mean   : 7.341   f.tax-(155,580]:57   f.mileage-(3.45e+04,1.19e+05]:111
##  3rd Qu.:10.995
##  Max.   :14.000
##
##                f.mpg                   f.year    inconsistencies
##  f.mpg-[20,44.8]  :93   f.mpg-[2008,2016]:107   Min.   :0.0000
##  f.mpg-(44.8,53.3]:17   f.mpg-(2016,2017]: 14   1st Qu.:0.0000
##  f.mpg-(53.3,61.4]:19   f.mpg-(2017,2019]: 32   Median :0.0000
##  f.mpg-(61.4,88.3]:36   f.mpg-(2019,2020]: 12   Mean   :0.3576
##                                                 3rd Qu.:1.0000
##                                                 Max.   :2.0000
##
##         mout
##  MvOut.No :  0
##  MvOut.Yes:165
##
##
```

```
##
##
##
```

```
summary(df)
```

```
##                model           year          price
##  VW- Golf          : 488   Min.   :2008   Min.   :  1250
##  Mercedes- C Class: 395   1st Qu.:2016   1st Qu.: 14000
##  VW- Polo          : 330   Median :2017   Median : 19430
##  Mercedes- A Class: 266   Mean   :2017   Mean   : 21419
##  BMW- 3 Series     : 251   3rd Qu.:2019   3rd Qu.: 25995
##  Mercedes- E Class: 199   Max.   :2020   Max.   :134219
##  (Other)           :3071
##             transmission     mileage                fuelType         tax
##  f.Trans-Manual   :1741   Min.   :     1   f.Fuel-Diesel:2860   Min.   :  0.0
##  f.Trans-SemiAuto :1939   1st Qu.:  5991   f.Fuel-Petrol:2074   1st Qu.:125.0
##  f.Trans-Automatic:1320   Median : 17065   f.Fuel-Hybrid:  66   Median :145.0
##                           Mean   : 23290                        Mean   :122.9
##                           3rd Qu.: 34514                        3rd Qu.:145.0
##                           Max.   :119000                        Max.   :580.0
##
##       mpg          engineSize     manufacturer                  f.price
##  Min.   :20.0   2      :2092   Audi    :1089   super cheap        :1000
##  1st Qu.:44.8   3      : 571   BMW     :1084   cheap              :1002
##  Median :53.3   1.5    : 520   Mercedes:1321   expensive          : 999
##  Mean   :53.0   2.1    : 395   VW      :1506   very expensive     :1004
##  3rd Qu.:61.4   1      : 374                   extremely expensive: 995
##  Max.   :88.3   1.6    : 365
##                 (Other): 683
##   Audi       years_after_sell             f.tax
##  No :3911   Min.   : 2.000   f.tax-[0,125]  :1447
##  Yes:1089   1st Qu.: 3.000   f.tax-(125,145]:2568
##             Median : 5.000   f.tax-(145,155]: 507
##             Mean   : 4.784   f.tax-(155,580]: 478
##             3rd Qu.: 6.000
##             Max.   :14.000
##
##                           f.mileage                  f.mpg
##  f.mileage-[1,5.99e+03]       :1250   f.mpg-[20,44.8]  :1280
##  f.mileage-(5.99e+03,1.71e+04]:1250   f.mpg-(44.8,53.3]:1328
##  f.mileage-(1.71e+04,3.45e+04]:1250   f.mpg-(53.3,61.4]:1208
##  f.mileage-(3.45e+04,1.19e+05]:1250   f.mpg-(61.4,88.3]:1184
##
##
##
##                f.year     inconsistencies         mout
##  f.mpg-[2008,2016]:1758   Min.   :0.0000   MvOut.No :4835
##  f.mpg-(2016,2017]: 881   1st Qu.:0.0000   MvOut.Yes: 165
##  f.mpg-(2017,2019]:2040   Median :0.0000
##  f.mpg-(2019,2020]: 321   Mean   :0.0296
##                           3rd Qu.:0.0000
##                           Max.   :2.0000
##
```

The cars that are outliers tend to be more expensive, have more mileage, have to pay more tax. The manufacturers Mercedes and VW have a low percentage of outliers cars.

7 Profiling with FactoMineR

7.1 Profiling of the numeric target variable "price"

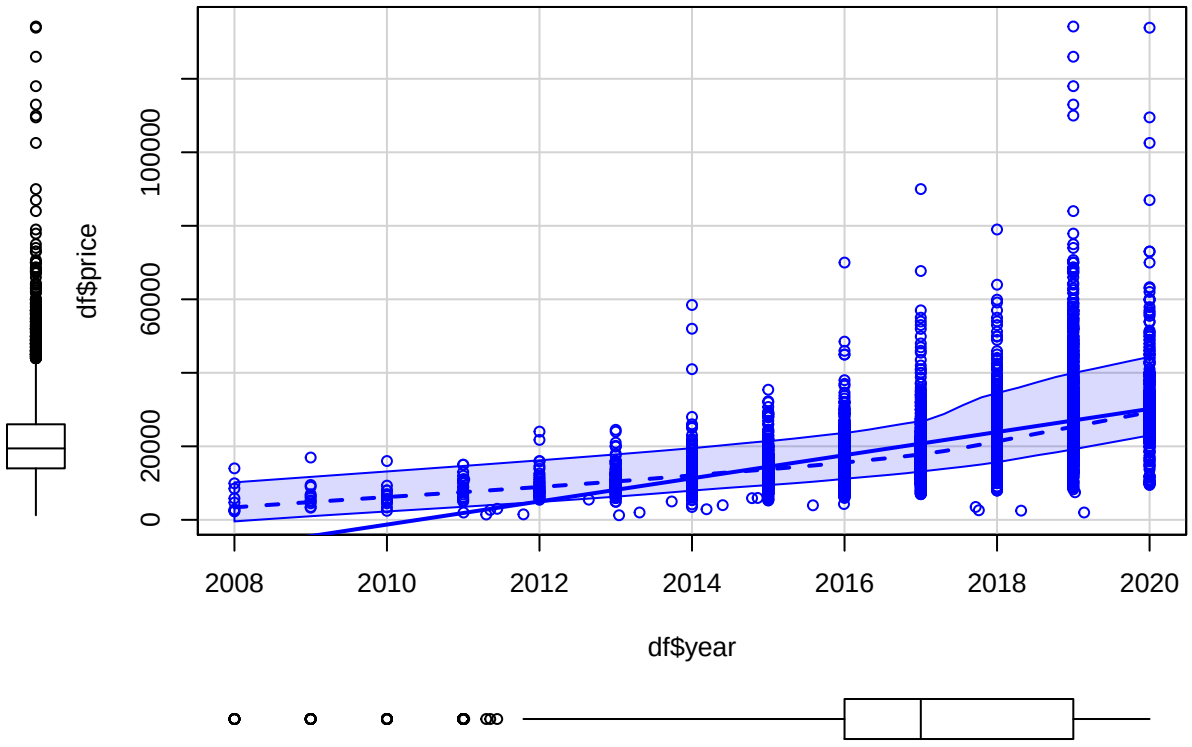
```
summary(df$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1250   14000   19430   21419   25995  134219
```

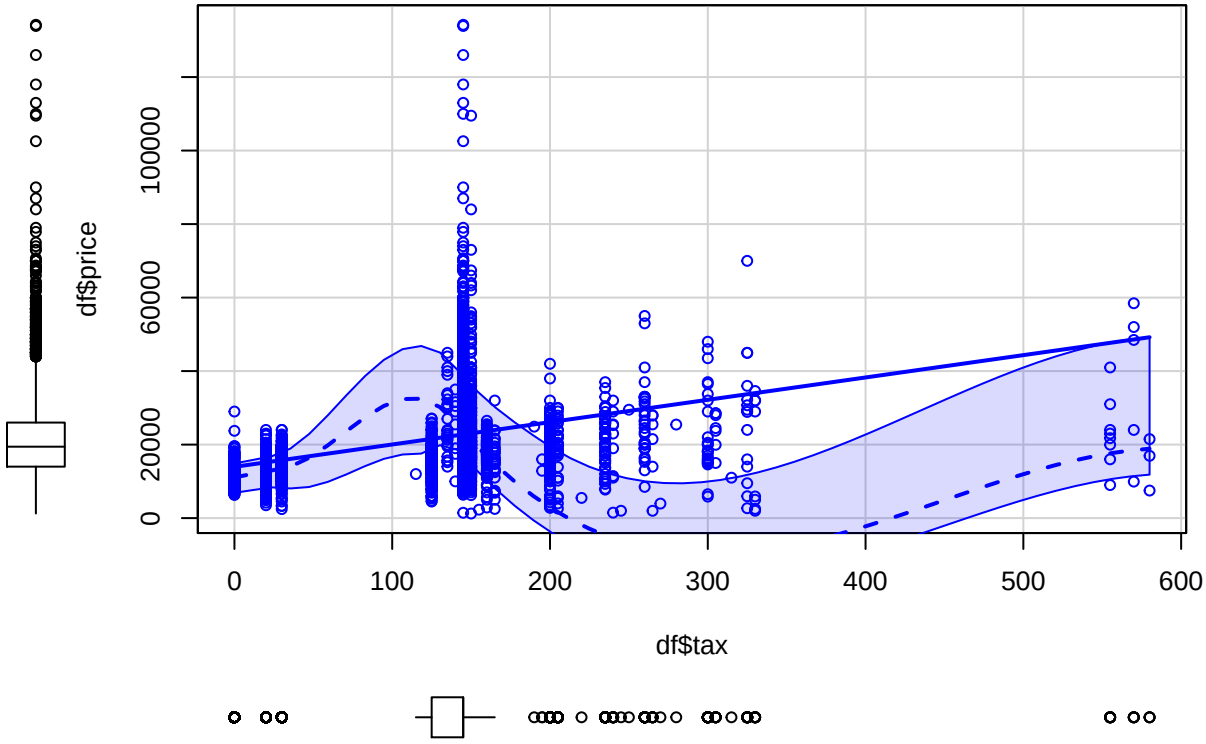
```
# The 'variable to describe cannot have NA
res.condes <- condes(df, 3, proba = 0.01)

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

```
##                   correlation       p.value
## year                0.5613989  0.000000e+00
## tax                 0.3583712 2.056371e-151
## inconsistencies     0.1774299  1.215367e-36
## mileage            -0.5045927 2.687717e-321
## years_after_sell   -0.5613989  0.000000e+00
## mpg                -0.5957920  0.000000e+00
```

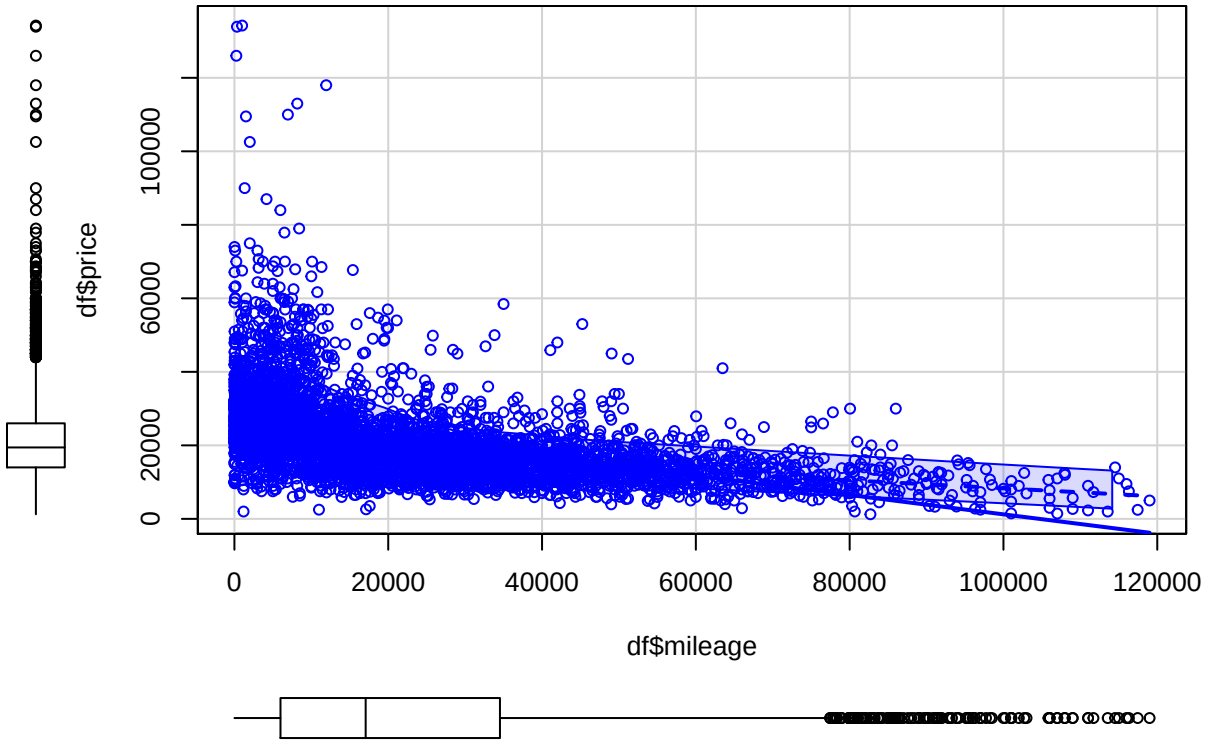
```
# The response variable has a strong correlation with the following
# variables: year, tax, mileage and mpg.
scatterplot(df$year, df$price)
```



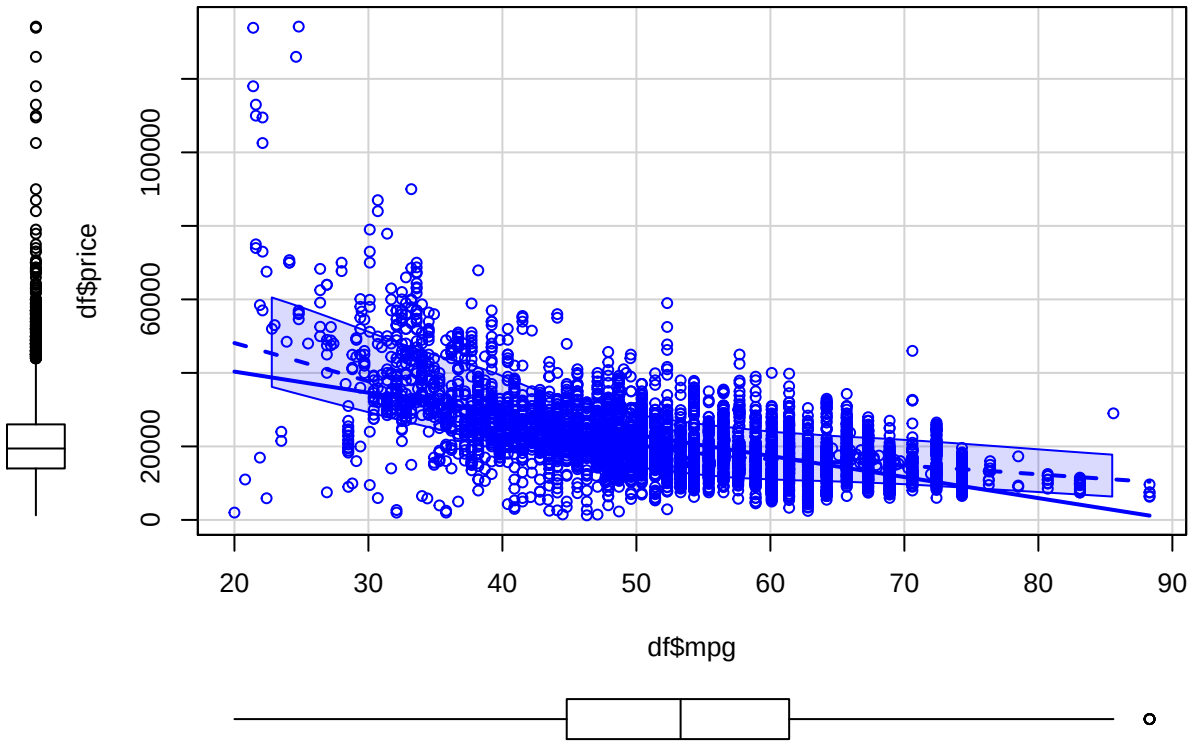

```
scatterplot(df$tax, df$price)
```




```
scatterplot(df$mileage, df$price)
```



```r
scatterplot(df$mpg, df$price)
```




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

```
##                       R2       p.value
## model        0.506310837  0.000000e+00
## engineSize   0.445356046  0.000000e+00
## f.price      0.740211567  0.000000e+00
## f.mileage    0.280287483  0.000000e+00
## f.mpg        0.303581211  0.000000e+00
## f.year       0.311671285  0.000000e+00
## f.tax        0.246203333 6.733681e-306
## transmission 0.214844139 3.525807e-263
## manufacturer 0.076387577  9.712025e-86
## mout         0.016136544  1.939148e-19
## Audi         0.007427251  1.035841e-09
## fuelType     0.002042108  6.051651e-03
```

```r
# P-values indicate whether the correlation is statistically
# different from 0 or not. p-values < 0.05 reject the null hypothesis
# (correlation statistically equal to 0).  The response variable has
# a strong correlation with the following variables: model,
# engineSize and transmission, according to the R2 stadistic value.

res.condes$category  # Partial association to significative levels in factors
```

```
##                                          Estimate       p.value
## f.mpg=f.mpg-[20,44.8]                   10134.5537  0.000000e+00
## f.price=extremely expensive             16706.3876  0.000000e+00
## f.tax=f.tax-(125,145]                    5718.0342 9.351134e-226
## f.year=f.mpg-(2017,2019]                 4257.8887 1.310858e-219
## f.mileage=f.mileage-[1,5.99e+03]         7868.8842 1.601333e-198
## engineSize=3                             3823.8262 2.336421e-158
```

```
## transmission=f.Trans-SemiAuto             4395.6513 5.287880e-117
## engineSize=4                              31033.9583 5.743782e-115
## engineSize=5.2                            77681.7632  3.028801e-67
## f.year=f.mpg-(2019,2020]                   8613.1641  1.583633e-64
## model=Audi- R8                            69823.5872  2.248929e-64
## model=Audi- Q7                            14965.6110  8.569659e-36
## f.mileage=f.mileage-(5.99e+03,1.71e+04]    3169.6962  2.033865e-31
## model=Mercedes- GLC Class                  6640.1156  1.895944e-26
## model=BMW- X7                             41524.4205  5.009171e-26
## model=Mercedes- GLE Class                 12369.9718  5.670081e-26
## transmission=f.Trans-Automatic             2678.2834  2.417659e-25
## manufacturer=Mercedes                      2417.0701  3.657767e-25
## model=Audi- Q8                            34193.6705  5.968125e-25
## model=Audi- RS6                           36545.0634  1.830541e-24
## f.price=very expensive                     3042.7908  1.839697e-22
## model=BMW- X5                              9849.1824  3.256878e-21
## model=Audi- Q5                             4093.3310  3.545023e-20
## mout=MvOut.Yes                             3971.7044  1.939148e-19
## engineSize=4.4                            24596.0132  2.964706e-16
## model=BMW- 8 Series                       34241.5205  3.750036e-16
## model=BMW- M4                             16597.0538  1.494936e-15
## model=BMW- M5                             32003.1205  1.426681e-14
## engineSize=2.9                            15686.5132  4.622873e-13
## model=BMW- 7 Series                       16061.2538  3.381104e-12
## model=VW- Touareg                          5520.8692  2.873120e-11
## model=BMW- X3                              3554.9405  3.243856e-10
## Audi=Yes                                   1166.1867  1.035841e-09
## manufacturer=Audi                          1523.3835  1.035841e-09
## model=Mercedes- S Class                    7952.7300  4.307614e-09
## model=Mercedes- GLS Class                 19097.7538  2.377482e-08
## model=BMW- X4                              5620.6348  9.143424e-07
## model=BMW- X6                             14269.3491  1.049275e-06
## engineSize=5.5                            18749.7632  2.391084e-06
## model=VW- Caravelle                       10840.8094  3.951980e-06
## manufacturer=BMW                           1001.3887  1.421101e-05
## model=BMW- X2                              2204.4872  2.790794e-05
## model=VW- California                      26266.9205  3.666024e-05
## engineSize=4.7                            11160.7632  3.941937e-05
## model=Audi- A8                             6739.5872  5.037590e-05
## model=Mercedes- SL CLASS                   1585.3688  1.340124e-04
## model=Mercedes- V Class                    1709.4857  5.565380e-04
## model=Audi- SQ7                           19261.9205  1.199686e-03
## engineSize=6.6                            28017.7632  1.440295e-03
## model=Audi- RS5                           26771.9205  3.052873e-03
## model=VW- Tiguan Allspace                  2963.8372  3.980291e-03
## model=Mercedes- X-CLASS                    3486.5205  5.497394e-03
## model=Mercedes- SLK                      -15719.8795  7.654442e-03
## fuelType=f.Fuel-Diesel                     -177.4435  7.057873e-03
## model=VW- Arteon                           -292.4795  6.926527e-03
## model=BMW- 3 Series                       -8166.4221  6.879860e-03
## model=VW- CC                             -18098.8295  2.807267e-03
## fuelType=f.Fuel-Petrol                    -1111.5447  2.552597e-03
## model=Mercedes- C Class                   -4633.8263  1.887334e-03
## model=VW- Scirocco                       -13696.9684  5.670642e-04
## model=VW- Passat                         -10395.3829  4.957409e-04
## f.tax=f.tax-(155,580]                      -663.5443  4.933855e-04
## engineSize=1.5                            -9219.2080  3.546677e-04
## engineSize=1.8                           -16776.5550  1.049266e-04
## engineSize=2.1                            -9951.5102  9.418664e-06
## engineSize=2                              -6703.3816  3.784604e-06
## model=Mercedes- E Class                   -2078.0292  4.756013e-08
## model=Audi- A3                           -10852.9111  5.969868e-09
## Audi=No                                   -1166.1867  1.035841e-09
## f.price=expensive                         -2073.3541  7.088486e-11
```

```
## model=BMW- 1 Series                       -12027.6426  5.558526e-13
## model=Audi- A1                            -13475.5180  1.086479e-13
## f.year=f.mpg-(2016,2017]                   -4341.1419  3.705028e-17
## mout=MvOut.No                              -3971.7044  1.939148e-19
## engineSize=1.6                            -13421.5409  1.316980e-25
## model=VW- Golf                            -11558.5426  4.236098e-28
## engineSize=1.4                            -13984.6275  1.312162e-28
## model=VW- Up                              -19465.2095  4.564383e-33
## engineSize=1.2                            -19174.8306  3.712678e-33
## f.mileage=f.mileage-(1.71e+04,3.45e+04]    -3647.7614  3.123053e-41
## f.mpg=f.mpg-(53.3,61.4]                    -4091.1056  2.356120e-53
## engineSize=1                              -16773.0657  2.190482e-63
## model=VW- Polo                            -16497.1492  8.427752e-68
## manufacturer=VW                            -4941.8422  5.149754e-86
## f.price=cheap                              -6170.8477  5.385902e-88
## f.mpg=f.mpg-(61.4,88.3]                    -5637.7271  5.626745e-97
## f.mileage=f.mileage-(3.45e+04,1.19e+05]    -7390.8190 1.916704e-173
## transmission=f.Trans-Manual                -7073.9347 1.010238e-259
## f.tax=f.tax-[0,125]                        -7084.9630 6.959088e-265
## f.year=f.mpg-[2008,2016]                   -8529.9109 5.950397e-267
## f.price=super cheap                       -11504.9766  0.000000e+00
```

With this output we can see from different categories the mean difference in price compared to the mean price of the dataset * The cars that have low mpg are more expensive. * We can also see that the cars with an engine size = 4 has an estimate of +19400$ * We can also see that the cars with an engine size = 2.9 has an estimate of +19179$ * We can also see that the cars with an model=BMW- 7 Series has an estimate of +18054$ * We can also see that the cars with an model=Audi- Q8 has an estimate of +25943$ * We can also see that the cars with an engine size = 5.2 has an estimate of +32960$ * We can also see that the cars with an model=VW- Up has an estimate of -17472$ * We can also see that the cars with an engine size = 1.2 has an estimate of -15682$

7.2 Profiling of the categorical target variable "Audi"

```
summary(df$Audi)
```

```
##   No  Yes
## 3911 1089
```

```
# The 'variable to describe cannot have NA
res.catdes <- catdes(df[, -c(1)], 11, proba = 0.01)
# We exclude the model of the car from the analysis because it
# doesn't bring useful information.
res.catdes$quanti.var  # Global association to numeric variables
```

```
##                Eta2      P-value
## mpg     0.012792186 1.046255e-15
## price   0.007427251 1.035841e-09
## mileage 0.002465673 4.439689e-04
```

Miles per galon (mpg), price and mileage are statistically significant variables as they have a p-value less than 0.01. Despite that fact, the effect size associated with them is quite small as they have a small Eta2 value. This means that these variables are not quite significant at predicting if a car is an Audi or not.

```
res.catdes$quanti  # Partial association of numeric variables to levels of outcome factor
```

```
## $No
##            v.test Mean in category Overall mean sd in category  Overall sd
## mpg      7.996758         53.69565     53.00322       11.41652    11.60192
## mileage -3.510826      22728.49600  23289.51910    21127.23916 21411.29702
## price   -6.093343      20910.54487  21418.53580    10559.77573 11170.48060
```

```
##              p.value
## mpg     1.277380e-15
## mileage 4.467167e-04
## price   1.105767e-09
##
## $Yes
##            v.test Mean in category Overall mean sd in category  Overall sd
## price    6.093343      23242.91827  21418.53580    12968.60553 11170.48060
## mileage  3.510826      25304.35963  23289.51910    22285.63765 21411.29702
## mpg     -7.996758         50.51647     53.00322       11.91745    11.60192
##              p.value
## price   1.105767e-09
## mileage 4.467167e-04
## mpg     1.277380e-15
```

With this output we can see that Audi cars have a little more price and mileage than the global average and have fewer mpg than the global average. The opposite is true for cars that are not Audi.

```
# mean(df$tax[which(df$Audi=='No')])-mean(df$tax[which(df$Audi=='Yes')])
res.catdes$test.chi2  # Global association to factors
```

```
##                   p.value df
## manufacturer 0.000000e+00  3
## engineSize   3.983094e-89 26
## f.mpg        7.074125e-18  3
## f.price      1.377668e-06  4
## fuelType     2.207182e-06  2
## transmission 2.560144e-05  2
## f.mileage    1.760678e-04  3
## f.year       8.509511e-03  3
```

```
res.catdes$category  # Partial association to significative levels in factors
```

```
## $No
##                                           Cla/Mod     Mod/Cla Global
## manufacturer=VW                         100.00000 38.50677576  30.12
## manufacturer=Mercedes                   100.00000 33.77652774  26.42
## manufacturer=BMW                        100.00000 27.71669650  21.68
## engineSize=2.1                          100.00000 10.09971874   7.90
## engineSize=1.2                           98.43750  3.22168243   2.56
## f.mpg=f.mpg-(61.4,88.3]                  84.37500 25.54333930  23.68
## engineSize=1.5                           87.88462 11.68499105  10.40
## engineSize=1.3                          100.00000  1.89209921   1.48
## fuelType=f.Fuel-Hybrid                   96.96970  1.63641013   1.32
## f.price=super cheap                      82.90000 21.19662490  20.00
## f.mileage=f.mileage-(5.99e+03,1.71e+04]  82.08000 26.23369982  25.00
## engineSize=1                             85.29412  8.15648172   7.48
## transmission=f.Trans-SemiAuto            80.66013 39.98977244  38.78
## f.year=f.mpg-(2017,2019]                 80.53922 42.00971619  40.80
## f.mpg=f.mpg-(53.3,61.4]                  81.29139 25.10866786  24.16
## fuelType=f.Fuel-Diesel                   79.72028 58.29711071  57.20
## engineSize=2.5                           16.66667  0.02556891   0.12
## engineSize=5.2                            0.00000  0.00000000   0.10
## fuelType=f.Fuel-Petrol                   75.55448 40.06647916  41.48
## engineSize=4                             46.34146  0.48580926   0.82
## transmission=f.Trans-Manual              74.61229 33.21401176  34.82
## f.price=extremely expensive              72.56281 18.46075173  19.90
## engineSize=2                             74.61759 39.91306571  41.84
## f.mpg=f.mpg-[20,44.8]                    70.07812 22.93531066  25.60
## engineSize=1.4                           46.93878  4.11659422   6.86
## manufacturer=Audi                         0.00000  0.00000000  21.78
##                                              p.value     v.test
```

```
## manufacturer=VW                            2.534150e-197  29.968395
## manufacturer=Mercedes                      3.047886e-168  27.646817
## manufacturer=BMW                           1.900767e-133  24.583411
## engineSize=2.1                              7.203244e-45  14.054753
## engineSize=1.2                              1.042675e-11   6.800485
## f.mpg=f.mpg-(61.4,88.3]                     1.657751e-09   6.028220
## engineSize=1.5                              2.439106e-09   5.965484
## engineSize=1.3                              1.095102e-08   5.715301
## fuelType=f.Fuel-Hybrid                      1.702797e-05   4.300674
## f.price=super cheap                         4.359026e-05   4.087577
## f.mileage=f.mileage-(5.99e+03,1.71e+04]     1.082646e-04   3.871282
## engineSize=1                                3.521010e-04   3.573604
## transmission=f.Trans-SemiAuto               8.318966e-04   3.341958
## f.year=f.mpg-(2017,2019]                    9.325096e-04   3.310135
## f.mpg=f.mpg-(53.3,61.4]                     2.694697e-03   3.000576
## fuelType=f.Fuel-Diesel                      3.042950e-03   2.963366
## engineSize=2.5                              2.497156e-03  -3.023686
## engineSize=5.2                              4.865917e-04  -3.488031
## fuelType=f.Fuel-Petrol                      1.283354e-04  -3.829631
## engineSize=4                                8.984095e-06  -4.440282
## transmission=f.Trans-Manual                 7.461294e-06  -4.480086
## f.price=extremely expensive                 2.222765e-06  -4.732038
## engineSize=2                                1.886058e-07  -5.210231
## f.mpg=f.mpg-[20,44.8]                       1.404785e-15  -7.985038
## engineSize=1.4                              8.785423e-40 -13.199898
## manufacturer=Audi                           0.000000e+00       -Inf
##
## $Yes
##                                               Cla/Mod     Mod/Cla Global
## manufacturer=Audi                          100.000000 100.0000000  21.78
## engineSize=1.4                              53.061224  16.7125803   6.86
## f.mpg=f.mpg-[20,44.8]                       29.921875  35.1698806  25.60
## engineSize=2                                25.382409  48.7603306  41.84
## f.price=extremely expensive                 27.437186  25.0688705  19.90
## transmission=f.Trans-Manual                 25.387708  40.5876951  34.82
## engineSize=4                                53.658537   2.0202020   0.82
## fuelType=f.Fuel-Petrol                      24.445516  46.5564738  41.48
## engineSize=5.2                             100.000000   0.4591368   0.10
## engineSize=2.5                              83.333333   0.4591368   0.12
## fuelType=f.Fuel-Diesel                      20.279720  53.2598714  57.20
## f.mpg=f.mpg-(53.3,61.4]                     18.708609  20.7529844  24.16
## f.year=f.mpg-(2017,2019]                    19.460784  36.4554637  40.80
## transmission=f.Trans-SemiAuto               19.339866  34.4352617  38.78
## engineSize=1                                14.705882   5.0505051   7.48
## f.mileage=f.mileage-(5.99e+03,1.71e+04]     17.920000  20.5693297  25.00
## f.price=super cheap                         17.100000  15.7024793  20.00
## fuelType=f.Fuel-Hybrid                       3.030303   0.1836547   1.32
## engineSize=1.3                               0.000000   0.0000000   1.48
## engineSize=1.5                              12.115385   5.7851240  10.40
## f.mpg=f.mpg-(61.4,88.3]                     15.625000  16.9880624  23.68
## engineSize=1.2                               1.562500   0.1836547   2.56
## engineSize=2.1                               0.000000   0.0000000   7.90
## manufacturer=BMW                             0.000000   0.0000000  21.68
## manufacturer=Mercedes                        0.000000   0.0000000  26.42
## manufacturer=VW                              0.000000   0.0000000  30.12
##                                                  p.value     v.test
## manufacturer=Audi                           0.000000e+00        Inf
## engineSize=1.4                              8.785423e-40  13.199898
## f.mpg=f.mpg-[20,44.8]                       1.404785e-15   7.985038
## engineSize=2                                1.886058e-07   5.210231
## f.price=extremely expensive                 2.222765e-06   4.732038
## transmission=f.Trans-Manual                 7.461294e-06   4.480086
## engineSize=4                                8.984095e-06   4.440282
## fuelType=f.Fuel-Petrol                      1.283354e-04   3.829631
```

```
## engineSize=5.2                              4.865917e-04   3.488031
## engineSize=2.5                              2.497156e-03   3.023686
## fuelType=f.Fuel-Diesel                      3.042950e-03  -2.963366
## f.mpg=f.mpg-(53.3,61.4]                     2.694697e-03  -3.000576
## f.year=f.mpg-(2017,2019]                    9.325096e-04  -3.310135
## transmission=f.Trans-SemiAuto               8.318966e-04  -3.341958
## engineSize=1                                3.521010e-04  -3.573604
## f.mileage=f.mileage-(5.99e+03,1.71e+04]     1.082646e-04  -3.871282
## f.price=super cheap                         4.359026e-05  -4.087577
## fuelType=f.Fuel-Hybrid                      1.702797e-05  -4.300674
## engineSize=1.3                              1.095102e-08  -5.715301
## engineSize=1.5                              2.439106e-09  -5.965484
## f.mpg=f.mpg-(61.4,88.3]                     1.657751e-09  -6.028220
## engineSize=1.2                              1.042675e-11  -6.800485
## engineSize=2.1                              7.203244e-45 -14.054753
## manufacturer=BMW                           1.900767e-133 -24.583411
## manufacturer=Mercedes                      3.047886e-168 -27.646817
## manufacturer=VW                            2.534150e-197 -29.968395
```

With this final categorical analysis we can see that: For cars that are not Audi: *We have smaller engine sizes overall.* The percentage of cars with diesel and hybrid engines is slightly higher than the global mean. *We have more cars with a lower mileage.

For cars that are Audi: *The percentage of engines with a size of 1.4 is higher than the global mean (16.9 vs 6.9).* The percentage of Audis with a manual transmission is higher than the global mean (41 vs 35).