Primera Práctica de ADEI

Laboratori 1 - Data Preparation

Alejandro Alarcón

10/19/2021

Contents

Configuración del environment

Limpiamos el environment.

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

```
## null device
##           1
```

```r
# Clean workspace
rm(list=ls())
```

Importamos las librerias y paquetes necesarios.

```r
options(contrasts=c("contr.treatment","contr.treatment"))

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

package.check <- lapply(requiredPackages, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})
```

```
## Loading required package: effects

## Loading required package: carData

## lattice theme set by effectsTheme()
## See ?effectsTheme for details.

## Loading required package: FactoMineR

## Loading required package: car

## Loading required package: factoextra

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

## Loading required package: RColorBrewer

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggmap

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.

## Loading required package: ggthemes

## Loading required package: missMDA
```

```
#verify they are loaded
search()
```

```
##  [1] ".GlobalEnv"          "package:missMDA"      "package:ggthemes"
##  [4] "package:ggmap"       "package:dplyr"        "package:RColorBrewer"
##  [7] "package:factoextra"  "package:ggplot2"      "package:car"
## [10] "package:FactoMineR"  "package:effects"      "package:carData"
## [13] "package:knitr"       "package:stats"        "package:graphics"
## [16] "package:grDevices"   "package:utils"        "package:datasets"
## [19] "package:methods"     "Autoloads"            "package:base"
```

Cargamos los datos

```
filepath<-"/Users/aleibz/ADEI/ADEI/ADEI/CarPrices/"
load(paste0(filepath,"MyOldCars-Raw.RData"))
```

Echamos un vistazo al dataset

```
summary( df )
```

```
##     model                year          price         transmission
##  Length:5000        Min.   :2001   Min.   :   899   Length:5000
##  Class :character   1st Qu.:2016   1st Qu.: 13995   Class :character
##  Mode  :character   Median :2017   Median : 19498   Mode  :character
##                     Mean   :2017   Mean   : 21207
##                     3rd Qu.:2019   3rd Qu.: 25980
##                     Max.   :2020   Max.   :109495
##     mileage          fuelType              tax              mpg
##  Min.   :      1   Length:5000        Min.   :  0.0   Min.   :  1.10
##  1st Qu.:   5815   Class :character   1st Qu.:125.0   1st Qu.: 45.60
##  Median : 17731   Mode  :character   Median :145.0   Median : 53.30
##  Mean   : 23590                      Mean   :122.8   Mean   : 53.93
##  3rd Qu.: 34130                      3rd Qu.:145.0   3rd Qu.: 61.40
##  Max.   :178000                      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   :1.909
##  3rd Qu.:2.000
##  Max.   :5.500
```

```
names( df )
```

```
##  [1] "model"        "year"         "price"        "transmission" "mileage"
##  [6] "fuelType"     "tax"          "mpg"          "engineSize"   "manufacturer"
```

0.1 Transformación de variables categóricas a factores

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

```
##  [1] "Audi- A1"             "Audi- A3"             "Audi- A4"
##  [4] "Audi- A5"             "Audi- A6"             "Audi- A7"
##  [7] "Audi- A8"             "Audi- Q2"             "Audi- Q3"
## [10] "Audi- Q5"             "Audi- Q7"             "Audi- Q8"
## [13] "Audi- R8"             "Audi- RS3"            "Audi- RS4"
## [16] "Audi- RS5"           "Audi- RS6"           "Audi- S3"
## [19] "Audi- S4"            "Audi- SQ5"           "Audi- TT"
## [22] "BMW- 1 Series"       "BMW- 2 Series"       "BMW- 3 Series"
## [25] "BMW- 4 Series"       "BMW- 5 Series"       "BMW- 6 Series"
## [28] "BMW- 7 Series"       "BMW- 8 Series"       "BMW- i3"
## [31] "BMW- i8"             "BMW- M2"             "BMW- M3"
## [34] "BMW- M4"             "BMW- M5"             "BMW- M6"
## [37] "BMW- X1"             "BMW- X2"             "BMW- X3"
## [40] "BMW- X4"             "BMW- X5"             "BMW- X6"
## [43] "BMW- X7"             "BMW- Z4"             "Mercedes- A Class"
## [46] "Mercedes- B Class"   "Mercedes- C Class"   "Mercedes- CL Class"
## [49] "Mercedes- CLA Class" "Mercedes- CLC Class" "Mercedes- CLS Class"
## [52] "Mercedes- E Class"   "Mercedes- G Class"   "Mercedes- GL Class"
## [55] "Mercedes- GLA Class" "Mercedes- GLB Class" "Mercedes- GLC Class"
## [58] "Mercedes- GLE Class" "Mercedes- GLS Class" "Mercedes- M Class"
## [61] "Mercedes- S Class"   "Mercedes- SL CLASS"  "Mercedes- SLK"
## [64] "Mercedes- V Class"   "Mercedes- X-CLASS"   "Mercedes-180"
## [67] "Mercedes-220"        "Mercedes-230"        "VW- Amarok"
## [70] "VW- Arteon"          "VW- Beetle"          "VW- Caddy Maxi Life"
## [73] "VW- California"      "VW- Caravelle"       "VW- CC"
## [76] "VW- Eos"             "VW- Fox"             "VW- Golf"
## [79] "VW- Golf SV"         "VW- Jetta"           "VW- Passat"
## [82] "VW- Polo"            "VW- Scirocco"        "VW- Sharan"
## [85] "VW- Shuttle"         "VW- T-Cross"         "VW- T-Roc"
## [88] "VW- Tiguan"          "VW- Tiguan Allspace" "VW- Touareg"
## [91] "VW- Touran"          "VW- Up"
```

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

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

```
df$transmission <- factor( df$transmission, levels = c("Manual","Semi-Auto","Automatic"),labels = paste(
head( df )
```

```
##       model year price      transmission mileage fuelType tax  mpg engineSize
## 1  Audi- A1 2017 12500    f.Trans-Manual   15735   Petrol 150 55.4        1.4
## 6  Audi- A1 2016 13900 f.Trans-Automatic   32260   Petrol  30 58.9        1.4
## 9  Audi- A3 2015 10200    f.Trans-Manual   46112   Petrol  20 60.1        1.4
## 23 Audi- A5 2017 22500 f.Trans-Automatic   21649   Diesel 145 58.9        3.0
## 25 Audi- Q5 2016 20000 f.Trans-Automatic   23789   Diesel 200 47.1        2.0
## 38 Audi- A6 2016 19400 f.Trans-Automatic   34030   Diesel 125 58.9        2.0
##    manufacturer
## 1          Audi
## 6          Audi
## 9          Audi
## 23         Audi
## 25         Audi
## 38         Audi
```

```
#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(

#Manufacturer
df$manufacturer <- factor(df$manufacturer)
```

0.2 Transformación de variables numéricas a factores

```
#Year + Age
summary(df$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2001    2016    2017    2017    2019    2020 
```

```
df$age <-  2021 - df$year
df$year<-factor(df$year)
summary(df$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   2.000   4.000   3.843   5.000  20.000 
```

```
#EngineSize
summary(df$engineSize)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   1.500   2.000   1.909   2.000   5.500 
```

```
df$engineSize_factor <- factor(df$engineSize)
summary(df$engineSize_factor)
```

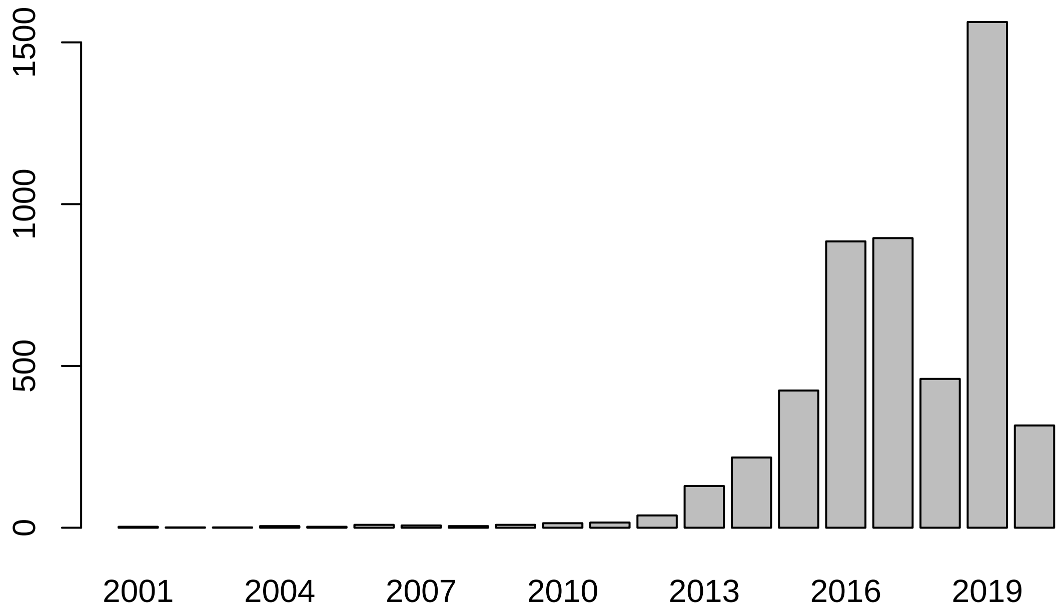
```
##    0    1  1.2  1.3  1.4  1.5  1.6  1.8  1.9    2  2.1  2.2  2.3  2.5  2.7  2.9 
##    9  365  147   63  310  554  345   38    1 2142  412   20    4    7    1   10 
##    3  3.2  3.5  3.7    4  4.2  4.4  4.7    5  5.5 
##  512    3    2    1   36    4    7    2    1    4 
```

0.3 Explorar variables con summary y plots

```
# Factores
summary( df$year )
```

```
## 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 
##    3    1    1    5    3    9    7    5    9   14   16   38  129  217  424  885 
## 2017 2018 2019 2020 
##  895  460 1563  316 
```

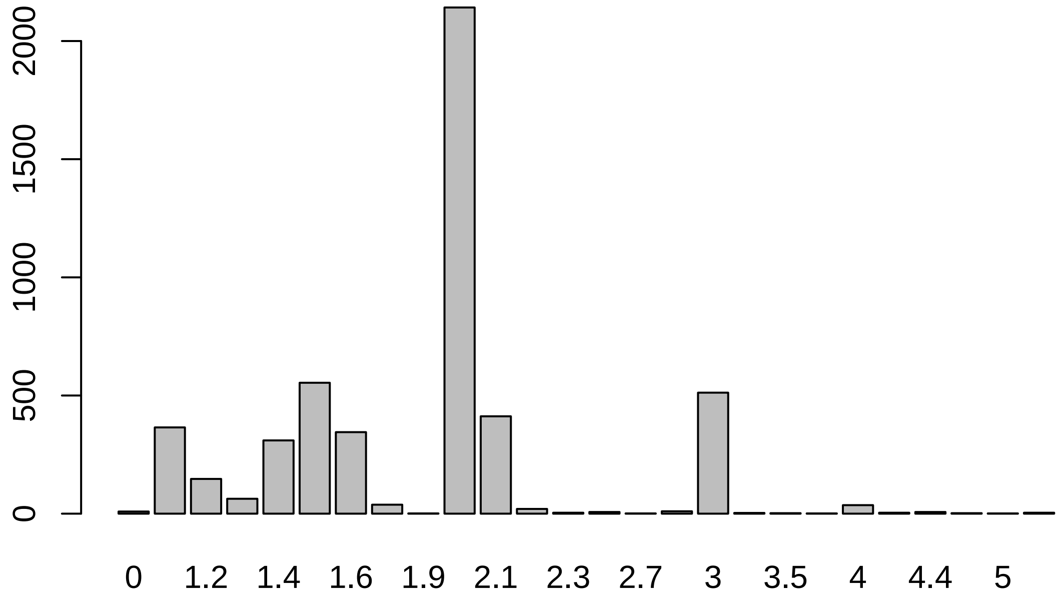
```
plot( df$year )
```



```
summary( df$engineSize_factor )
```

```
##    0    1  1.2  1.3  1.4  1.5  1.6  1.8  1.9    2  2.1  2.2  2.3  2.5  2.7  2.9
##    9  365  147   63  310  554  345   38    1 2142  412   20    4    7    1   10
##    3  3.2  3.5  3.7    4  4.2  4.4  4.7    5  5.5
##  512    3    2    1   36    4    7    2    1    4
```

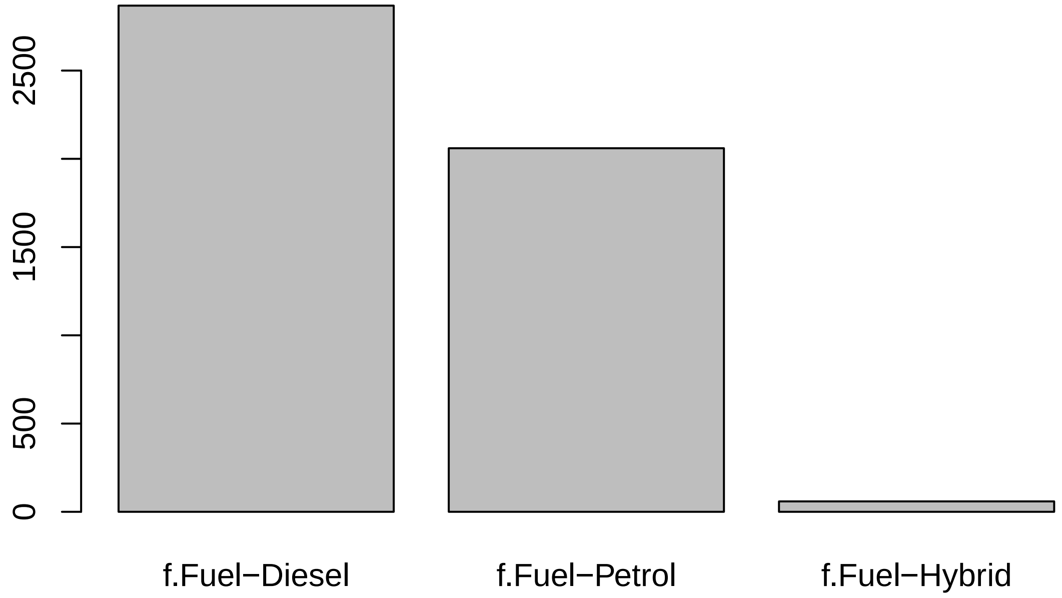
```
plot( df$engineSize_factor )
```



```
summary( df$fuelType )
```

```
## f.Fuel-Diesel f.Fuel-Petrol f.Fuel-Hybrid          NA's
##          2868          2060            59            13
```

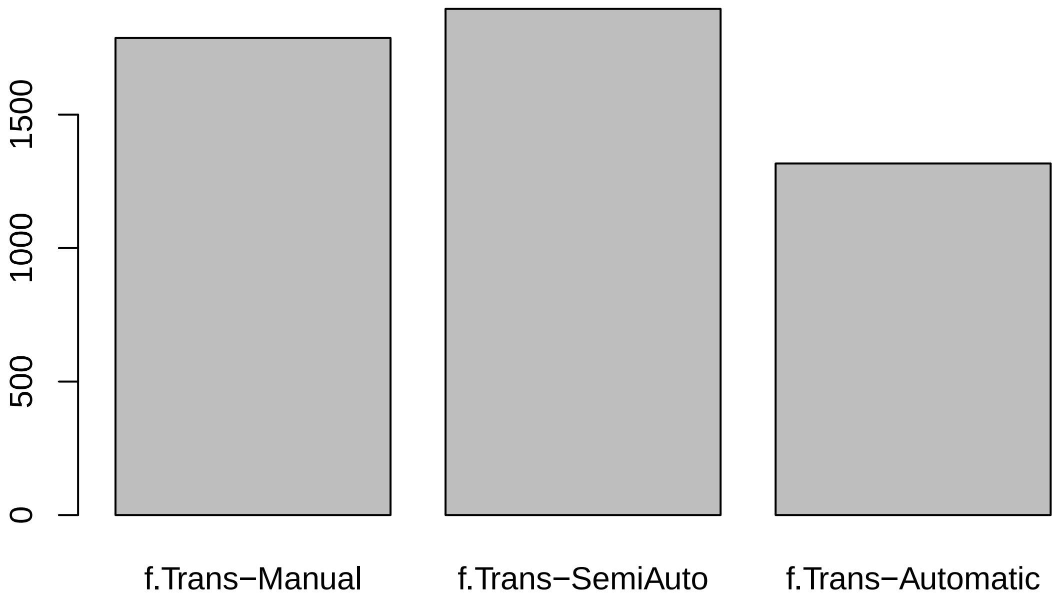
```
plot( df$fuelType )
```



```
summary( df$transmission )
```

```
##    f.Trans-Manual  f.Trans-SemiAuto f.Trans-Automatic
##              1787              1896              1317
```

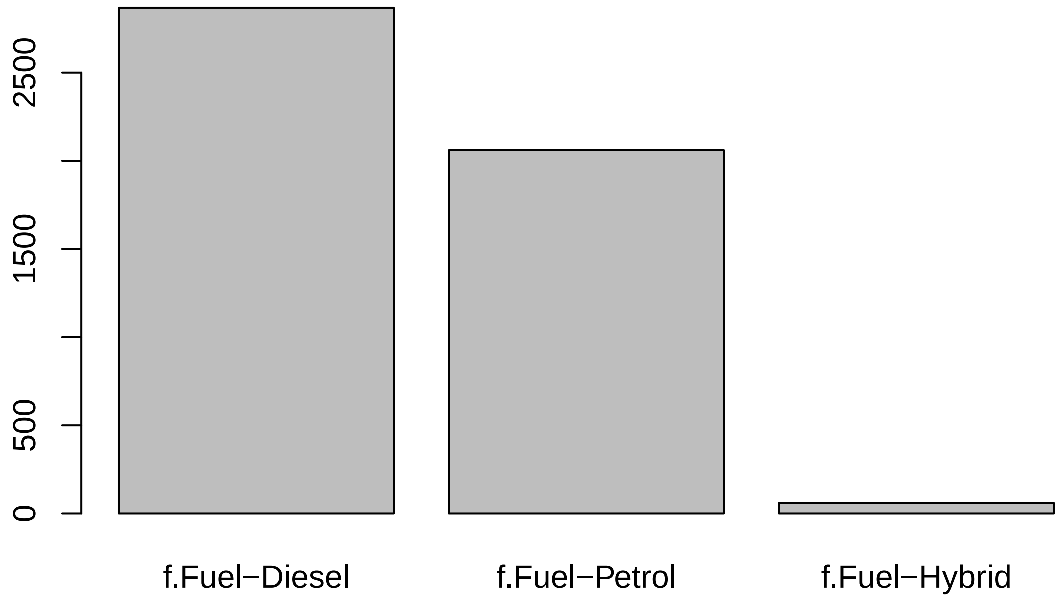
```
plot( df$transmission )
```



```
summary(df$fuelType )
```

```
## f.Fuel-Diesel f.Fuel-Petrol f.Fuel-Hybrid          NA's
##          2868          2060            59            13
```

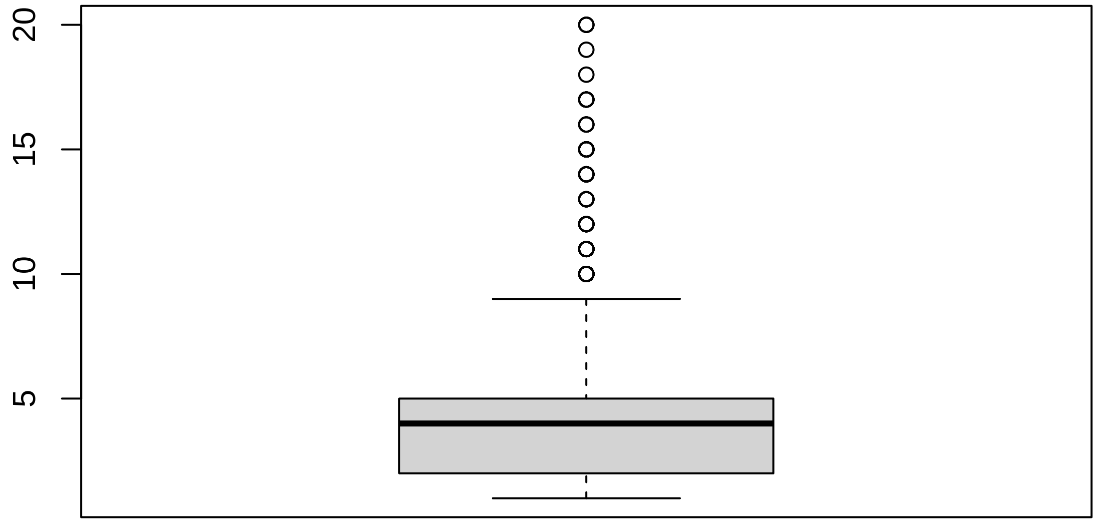
```
plot( df$fuelType )
```



```
# Numéricas
summary( df$age )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   2.000   4.000   3.843   5.000  20.000
```

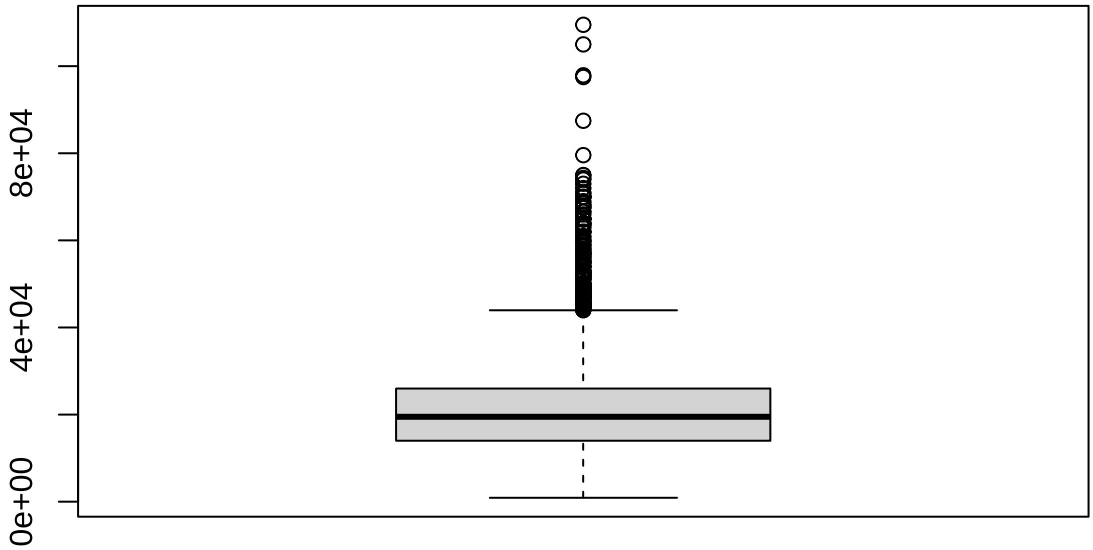
```
boxplot( df$age )
```



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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     899   13995   19498   21207   25980  109495
```

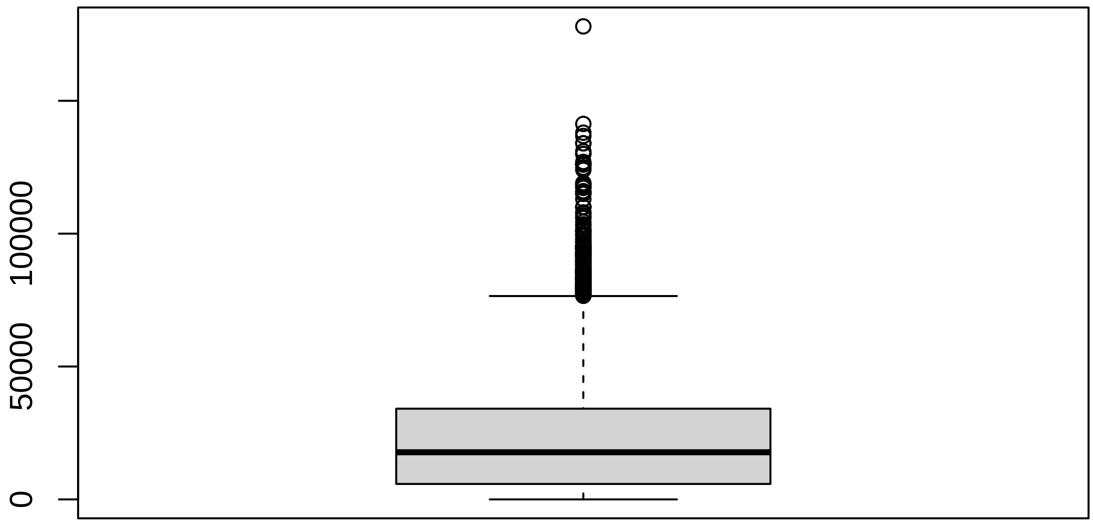
```
boxplot( df$price )
```



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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1    5815   17731   23590   34130  178000
```

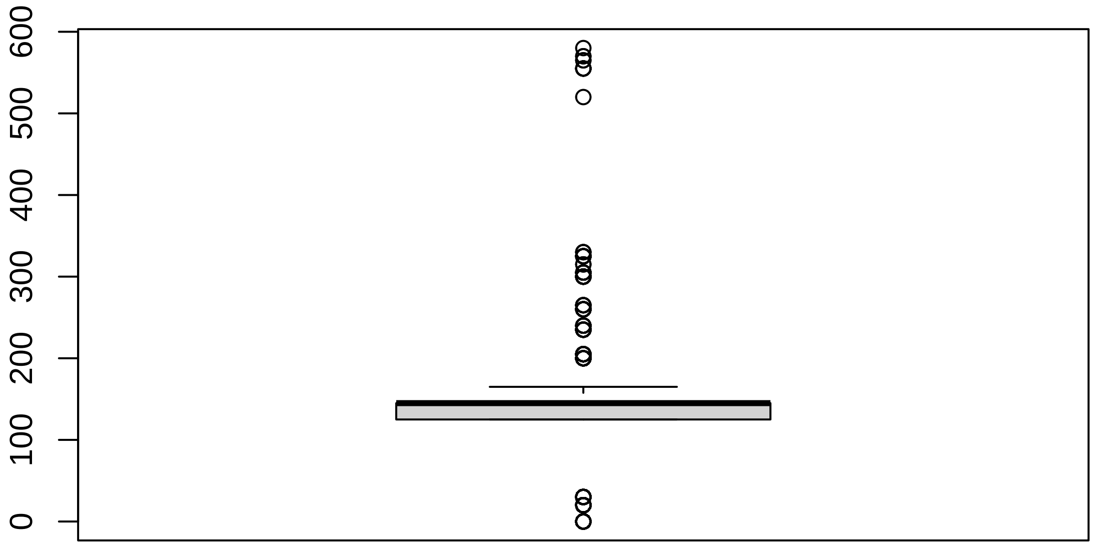
```
boxplot( df$mileage )
```



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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   125.0   145.0   122.8   145.0   580.0
```

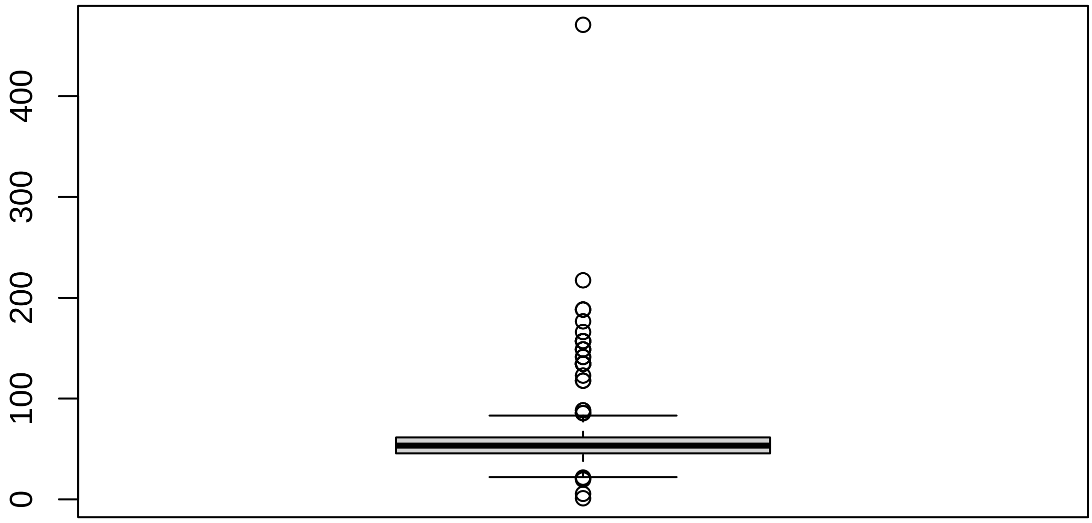
```
boxplot( df$tax )
```



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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.10   45.60   53.30   53.93   61.40  470.80
```

```
boxplot( df$mpg )
```



Funciones útiles

```
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) }
```

1 Por cada variable

1.1 Conteo de missings

```
imis<-rep(0,nrow(df))  # rows - trips
jmis<-rep(0,2*ncol(df))  # columns - variables

mis1<-countNA(df)
imis<-mis1$mis_ind
inds <- which(imis > 0)
df[inds,]
```

```
##                   model year price      transmission mileage fuelType tax   mpg
## 21018     BMW- 3 Series 2017 15300 f.Trans-Automatic   39428     <NA>   0 148.7
## 21110     BMW- 3 Series 2017 16000 f.Trans-Automatic   47495     <NA> 135 134.5
## 34344 Mercedes- C Class 2020 40999 f.Trans-Automatic     400     <NA> 135 217.3
## 39938          VW- Golf 2019 16889    f.Trans-Manual   12954     <NA> 150  45.6
## 40147          VW- Golf 2016 13795 f.Trans-Automatic   24463     <NA>  30  53.3
## 44517          VW- Polo 2019 12889    f.Trans-Manual   13016     <NA> 145  48.7
## 44524          VW- Polo 2019 13649    f.Trans-Manual    5000     <NA> 145  48.7
## 44653          VW- Polo 2019 14995 f.Trans-Automatic   10763     <NA> 145  45.6
## 46378        VW- Tiguan 2019 24999 f.Trans-Automatic    8491     <NA> 145  36.2
## 47541            VW- Up 2015  6799    f.Trans-Manual   28291     <NA>  20  62.8
## 47543            VW- Up 2020 10899    f.Trans-Manual    5000     <NA> 145  54.3
## 48422       VW- Touareg 2014 20995 f.Trans-Automatic   30523     <NA> 300  39.2
## 48423       VW- Touareg 2015 19995 f.Trans-Automatic   59115     <NA> 235  42.8
##       engineSize manufacturer age engineSize_factor
## 21018        2.0          BMW   4                 2
## 21110        2.0          BMW   4                 2
## 34344        2.0     Mercedes   1                 2
## 39938        1.0           VW   2                 1
## 40147        1.4           VW   5               1.4
## 44517        1.0           VW   2                 1
## 44524        1.0           VW   2                 1
## 44653        1.0           VW   2                 1
## 46378        1.5           VW   2               1.5
## 47541        1.0           VW   6                 1
## 47543        1.0           VW   1                 1
## 48422        3.0           VW   7                 3
## 48423        3.0           VW   6                 3
```

```
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             13
## tax                   0
## mpg                   0
## engineSize            0
## manufacturer          0
## age                   0
## engineSize_factor     0
```

Como se puede apreciar en la salida, solo aparecen un total de 13 missing values en todo el dataframe. Si miramos por variables, estos 13 missings aparecen en la columna de engineSize. Por otro lado, y si miramos por individuos, podemos ver como para los 13 individuos que tienen missings, este está en la columna de engineSize.

1.2 Conteo de outliers

```r
iouts<-rep(0,nrow(df))  # rows - trips
jouts<-rep(0,2*ncol(df))  # columns - variables

# Funcion que recibe como parametro una columna y devuelve los ids de los individuos outlier
outliers_column <- function(x){
  out_bounds <- calcQ(x)
  ex <- which((x<out_bounds$souti)|(x>out_bounds$souts))
  mild <- which(((x>out_bounds$souti)&(x<out_bounds$mouti))|((x<out_bounds$souts)&(x>out_bounds$mouts))
  list(extreme=ex, mild=mild)
}

# Estos son los outliers tanto mild como extreme de las variables numéricas
outs_price <- outliers_column(df$price)
outs_mileage <- outliers_column(df$mileage)
outs_tax <- outliers_column(df$tax)
outs_mpg <- outliers_column(df$mpg)
outs_age <- outliers_column(df$age)

# Conteo de extreme outliers
sum_extreme_outliers <- length(outs_price$extreme) + length(outs_mileage$extreme) + length(outs_tax$extr

# Conteo de mild outliers
sum_mild_outliers <- length(outs_price$mild) + length(outs_mileage$mild) + length(outs_tax$mild) + lengt

# Es curioso que sum_extreme > sum_mild
```

Ponemos los extreme outliers como missings

```r
df[outs_mileage$extreme,"mileage"]<-NA
df[outs_tax$extreme,"tax"]<-NA
df[outs_mpg$extreme,"mpg"]<-NA
df[outs_age$extreme,"age"]<-NA
```

1.3 Conteo de errores

```r
# Price
  err_price <- which(df$price<0)

# Age
  err_age <- which(df$age<0)

# Mileage
  err_mileage <- which(df$mileage<0)

# mpg
  err_mpg <- which(df$mpg<0)

# engineSize
  err_engineSize_fac <- which(df$engineSize_factor == "0")
  df[err_engineSize_fac,"engineSize_factor"] <- NA

# tax
  err_tax <- which(df$tax<0)
```

#Por cada individuo ## Conteo de missings (función que reciba el individuo como parámetro)

```
#DONE
```

1.4 Conteo de errores (función que reciba el individuo como parámetro)

1.5 Conteo de outliers (función que reciba el individuo como parámetro)

2 Imputación

```
library(missMDA)

names(df)
```

```
##  [1] "model"              "year"               "price"
##  [4] "transmission"       "mileage"            "fuelType"
##  [7] "tax"                "mpg"                "engineSize"
## [10] "manufacturer"       "age"                "engineSize_factor"
```

```
vars_num <- names(df)[c(3,5,7,8,11)]
vars_cat <- names(df)[c(1,2,4,6,10,12)]
vars_res <- names(df)[c(3)]
```

2.1 Variables numéricas

```
summary(df[,vars_num])
```

```
##      price           mileage             tax             mpg
##  Min.   :   899   Min.   :     1   Min.   :125    Min.   : 1.10
##  1st Qu.: 13995   1st Qu.:  5800   1st Qu.:145    1st Qu.:45.60
##  Median : 19498   Median : 17632   Median :145    Median :53.30
##  Mean   : 21207   Mean   : 23238   Mean   :147    Mean   :53.07
##  3rd Qu.: 25980   3rd Qu.: 34000   3rd Qu.:145    3rd Qu.:61.40
##  Max.   :109495   Max.   :119000   Max.   :205    Max.   :88.30
##                   NA's   :16       NA's   :1297   NA's   :42
##       age
##  Min.   : 1.000
##  1st Qu.: 2.000
##  Median : 4.000
##  Mean   : 3.786
##  3rd Qu.: 5.000
##  Max.   :14.000
##  NA's   :22
```

```
res.impca <- imputePCA(df[,vars_num], ncp = 2)
summary(res.impca$completeObs)
```

```
##      price           mileage             tax              mpg
##  Min.   :   899   Min.   :     1   Min.   :125.0   Min.   : 1.10
##  1st Qu.: 13995   1st Qu.:  5815   1st Qu.:145.0   1st Qu.:45.60
##  Median : 19498   Median : 17731   Median :145.0   Median :53.30
##  Mean   : 21207   Mean   : 23335   Mean   :147.2   Mean   :53.08
##  3rd Qu.: 25980   3rd Qu.: 34130   3rd Qu.:148.8   3rd Qu.:61.40
##  Max.   :109495   Max.   :119000   Max.   :205.0   Max.   :88.30
##       age
##  Min.   : 1.000
##  1st Qu.: 2.000
##  Median : 4.000
##  Mean   : 3.802
##  3rd Qu.: 5.000
##  Max.   :14.000
```

```
res.impca$completeObs[ outs_age$extreme, "age" ]
```

```
##     9880    10553    17953    19848    20506    20544    20688    20907
## 6.602015 8.673582 8.851468 5.201589 4.711389 7.264752 7.145131 7.729946
##    21393    32313    32539    32880    33048    33220    33245    33263
## 8.288108 5.121137 8.194822 7.444977 6.846731 8.919170 9.525526 6.001603
##    33449    34216    40774    47869    47892    48049
## 6.962624 9.070462 8.499986 9.926537 3.672846 4.947717
```

```
res.impca$completeObs[ outs_mileage$extreme, "mileage" ]
```

```
##    18967    18980    19848    19860    19884    20904    20905    30929
## 58822.00 60846.32 38556.10 36678.74 36505.55 48191.82 65719.81 59763.08
##    31611    40774    40845    40857    40868    40999    41002    48049
## 43062.02 73984.40 44376.78 38165.07 49579.92 81116.31 87156.48 35789.67
```

```
res.impca$completeObs[ outs_tax$extreme, "tax" ]
```

```
##        6        9       41       54       70       72      116      117
## 146.8751 149.8166 150.4097 147.5539 139.9082 146.0690 146.2748 162.1079
##      119      160      193      267      313      330      332      375
## 146.6063 143.2260 142.9819 145.3666 140.1407 145.0229 146.0074 141.1382
##      590      680      711      814      822      836      888     1046
## 146.0495 145.6798 139.3950 151.4931 141.2450 143.6145 146.7270 142.6511
##     1090     1120     1238     1240     1401     1410     1421     1447
## 164.6176 150.2173 148.8011 138.3558 147.6012 149.9946 154.1072 155.8185
##     1464     1475     1521     1635     1656     1672     1710     1723
## 144.1177 141.0957 145.9892 147.5801 145.7338 155.4156 148.5864 149.4048
##     1733     1740     1771     1790     1805     1870     1900     1901
## 149.7956 140.1895 141.5528 143.2464 143.9668 174.4169 142.6582 145.0365
##     1920     1955     1960     2082     2101     2135     2153     2205
## 149.2772 157.8106 145.5166 146.8619 150.6027 151.2503 157.4344 150.6660
##     2243     2247     2253     2269     2311     2336     2407     2438
## 138.7693 145.5790 148.9097 145.1528 140.9975 139.0951 143.3311 143.1626
##     2451     2470     2503     2568     2573     2631     2638     2641
## 142.6637 160.3896 156.7434 148.3496 145.3568 151.6135 144.3088 148.5887
##     2655     2675     2791     2834     2864     2865     2878     2886
## 144.4646 149.0618 145.5269 146.5429 147.1630 147.2093 166.9710 140.5098
##     2941     2982     2995     3043     3054     3059     3071     3100
## 144.8806 146.7688 145.5362 163.2755 140.8157 135.3222 147.0683 156.3106
##     3133     3218     3264     3297     3391     3393     3409     3537
## 140.4929 145.2146 145.8299 150.0931 168.7625 142.7244 145.4271 139.3017
##     3545     3548     3590     3606     3614     3615     3621     3622
## 150.2742 147.5722 145.1206 149.0002 147.2876 153.8697 145.9278 142.4212
##     3828     3895     3916     4003     4053     4143     4267     4296
## 142.0284 143.3359 148.1260 140.4269 143.8269 147.3346 143.8975 145.0911
##     4383     4413     4511     4622     4682     4932     4965     4968
## 144.4129 143.0913 147.0227 148.5590 142.4225 149.5841 147.7869 174.3064
##     4972     5110     5199     5240     5267     5330     5407     5581
## 159.5033 148.8898 147.9717 144.4403 145.8161 149.6234 141.3861 149.4004
##     5587     5621     5668     5820     5836     6179     6392     6455
## 149.6866 162.2782 143.2403 159.2125 146.9381 156.4877 145.0884 146.8060
##     6560     6601     6619     6633     6660     6685     6734     6741
## 141.3822 140.0200 145.6362 143.6898 163.1279 147.0521 146.5061 143.0331
##     6862     6864     6866     6868     6915     7084     7199     7258
## 140.1572 146.8436 142.3144 149.9090 162.4355 143.2284 144.3752 158.1175
##     7259     7309     7417     7436     7470     7491     7496     7538
## 144.7923 146.3411 176.2125 166.4764 145.6963 145.2110 153.1758 144.4206
##     7635     7676     7773     7787     7856     7900     7901     7908
## 147.2800 140.3919 176.8961 147.6457 146.3294 151.5580 156.5097 143.0559
##     7932     7934     7938     7947     7987     7994     8009     8024
## 149.5421 139.8784 144.1504 140.6540 154.7868 146.8816 138.5589 164.3911
##     8028     8031     8049     8052     8061     8100     8132     8210
## 148.7346 145.1646 139.1487 160.3384 147.2333 149.9993 143.7412 150.1837
##     8251     8365     8367     8395     8410     8415     8429     8536
## 158.9517 146.7029 136.4557 139.1093 134.3869 147.9909 140.4894 142.3865
##     8540     8552     8580     8690     8719     8794     8859     8865
## 154.2176 142.3558 141.8313 147.7579 145.2868 144.3882 150.3033 144.1511
##     8876     8889     8899     8944     8954     8959     8974     8979
## 154.4128 155.6466 151.2286 142.5549 141.0807 142.5227 152.8762 147.1043
##     8982     8984     8995     9010     9022     9026     9030     9077
## 151.2481 150.1544 151.0698 150.1818 150.8459 138.8770 147.4454 185.9908
##     9124     9145     9152     9158     9202     9270     9287     9291
## 149.8449 164.5315 142.7939 143.9733 147.4925 155.5260 145.4076 141.9938
##     9298     9486     9534     9622     9645     9655     9670     9822
## 142.5133 146.3875 145.1467 161.7661 142.9146 146.3813 146.9581 141.1352
##     9861     9872     9875     9880     9894     9915     9917     9931
## 146.1646 165.3214 140.9531 173.9509 168.2888 148.7023 140.7246 145.9930
##     9956     9969    10010    10072    10170    10171    10207    10214
## 158.5629 144.5353 151.5942 145.8681 148.0140 144.1346 179.3623 177.6020
##    10235    10253    10259    10265    10290    10314    10360    10376
## 172.4067 144.2658 174.8562 141.0100 139.1616 156.0013 138.3155 156.7543
##    10418    10441    10443    10483    10511    10514    10553    10583
## 140.9019 150.1181 154.2469 149.3722 164.0334 152.9973 175.9424 147.1301
##    10616    10617    10621    10658    10752    10762    10774    10797
## 155.6359 163.9378 154.2512 145.1528 142.6087 161.8801 148.6620 146.9680
##    10851    10898    10951    10959    11087    11354    11366    11369
## 149.1337 158.1543 136.5647 140.3042 157.8881 143.5498 152.9475 142.5146
##    11400    11428    11561    11651    11668    11687    11720    11731
## 144.7869 143.9337 160.6720 156.1493 139.3904 139.2615 155.8115 154.1532
##    11760    11765    11863    11864    11889    11955    12033    12035
## 149.4075 142.5511 146.6276 163.6252 155.6317 153.0055 147.3595 142.5994
##    12227    12265    12326    12512    12603    12773    12872    13029
## 138.7358 146.9632 142.9798 145.4528 150.6954 157.7683 139.0152 145.5504
##    13167    13301    13360    13362    13389    13583    13757    14079
## 145.5847 137.5992 151.5626 147.7213 137.4654 142.3022 157.8289 145.7667
##    14227    14236    14247    14282    14283    14449    14483    14794
## 138.1812 145.1110 135.4353 153.5322 146.8882 150.3560 154.3473 153.3695
##    14822    14918    14937    15100    15241    15257    15265    15271
## 141.0512 141.4589 164.8359 144.4219 149.2110 139.5328 141.1482 144.8321
##    15548    15563    15579    16136    16172    16281    16352    16357
## 148.1304 141.3114 143.4479 144.1009 144.9755 138.5569 145.2528 147.1697
##    16440    16641    16650    16655    16812    17059    17073    17283
## 140.8958 151.7712 142.1011 166.6693 142.6511 141.1161 141.0630 143.4850
##    17475    17562    17733    17934    17953    17979    18035    18059
## 159.9035 145.6501 164.7062 149.0268 180.9405 154.0267 146.7442 174.8020
##    18137    18141    18152    18195    18225    18278    18301    18317
## 162.5929 143.1366 141.6803 153.6502 145.0268 156.6831 157.9334 166.9108
##    18330    18338    18348    18351    18390    18410    18460    18511
## 145.0093 157.8922 150.9833 135.4537 144.4095 159.8197 145.0389 142.1915
##    18588    18609    18659    18660    18713    18748    18759    18760
## 141.1361 145.4027 144.5899 150.7279 145.5131 156.7091 146.4429 148.7947
##    18820    18840    18841    18847    18901    18955    19128    19133
## 149.4983 151.6800 146.3579 158.7255 161.2278 137.7000 147.2864 142.0787
##    19138    19152    19174    19218    19243    19254    19263    19274
## 149.0152 139.0722 142.8732 142.7282 153.0741 153.6066 142.8318 143.0829
##    19280    19302    19306    19334    19376    19385    19424    19460
## 143.8505 149.3823 140.1451 161.1557 134.7106 139.2738 137.9364 149.5074
##    19474    19532    19540    19551    19611    19627    19632    19638
## 144.7163 183.8383 148.8370 157.2341 144.4707 139.4905 146.6288 160.9649
##    19671    19696    19699    19707    19718    19746    19772    19802
## 135.3735 151.4388 131.6041 137.6487 154.2434 151.7487 137.7405 146.6802
##    19837    19838    19844    19848    19884    19925    19931    19937
## 148.9170 156.6118 154.4724 161.7622 145.0921 153.7613 156.4361 153.8258
##    19986    20006    20034    20066    20072    20120    20122    20160
## 149.6933 134.6355 150.5476 148.9953 137.0652 145.2219 142.6613 149.1882
##    20233    20238    20262    20275    20277    20278    20332    20391
## 136.5566 148.3913 139.0996 134.5909 135.7239 151.7928 149.0475 146.0184
##    20420    20472    20488    20506    20544    20587    20593    20644
## 151.4657 151.0167 151.1646 167.3009 172.7065 167.9669 149.1418 152.1415
##    20682    20696    20706    20744    20765    20781    20813    20823
## 167.7033 176.8418 140.3231 147.3673 143.7678 142.8301 151.9401 139.8733
##    20844    20856    20886    20902    20904    20905    20907    20943
## 140.2671 148.4713 152.6091 148.6814 150.2589 158.0092 175.6158 149.7240
##    21007    21018    21054    21091    21114    21151    21174    21185
## 170.1944 147.6640 153.4348 150.7260 150.4988 146.7816 138.8541 152.5331
##    21215    21216    21230    21319    21362    21393    21429    21431
## 143.0066 144.0371 138.6802 141.0134 146.4665 185.0860 148.3578 139.8257
##    21451    21452    21454    21469    21477    21498    21592    21616
## 145.8125 162.9424 164.7339 146.0115 135.3979 143.4181 137.4781 152.8119
##    21632    21693    21706    21861    21874    21887    21896    21926
## 158.4529 148.5850 161.2563 155.1687 140.7201 149.6491 149.0755 152.3646
##    21955    21977    22032    22049    22183    22217    22285    22343
## 140.8062 140.7104 148.5790 145.2489 137.6181 139.9751 143.9076 141.2739
##    22368    22414    22507    22579    22586    22648    22651    22691
## 145.8095 142.2755 146.9978 146.8476 144.3929 158.8751 161.5061 146.5488
##    22721    22747    22765    22794    22834    22835    22858    22863
## 142.8110 142.1820 143.4380 137.6503 146.2743 155.1581 152.0214 161.3632
##    22883    22887    22936    22944    22987    23003    23013    23043
## 138.3012 139.2216 144.3286 145.7715 153.2332 145.0482 145.2026 151.8723
##    23052    23065    23114    23127    23151    23209    23254    23261
## 141.8986 148.9200 149.3980 168.7296 165.7368 144.0516 145.9291 144.0995
##    23289    23321    23434    23443    23498    23554    23556    23565
## 141.0124 140.3595 151.5269 144.8244 145.9672 143.7631 138.4337 145.0167
##    23640    23642    23747    23875    23883    23916    23922    23923
## 137.7259 142.9648 143.7920 138.7721 153.1568 159.0387 151.6614 141.6914
##    24019    24134    24152    24267    24277    24293    24313    24343
## 144.7337 144.1833 143.0161 144.0508 144.9458 137.7072 146.4464 145.9387
##    24435    24488    24504    24525    24529    24547    24708    24753
## 143.9718 140.9341 144.0476 143.0017 142.3618 142.5966 147.7069 141.1488
##    24844    24847    24863    24885    24892    24955    24989    24992
## 141.5327 145.7225 150.0647 140.4264 140.6810 150.3530 143.7137 144.5030
##    25003    25152    25177    25227    25243    25273    25328    25527
## 159.6200 144.5772 144.1999 140.3290 142.9433 143.3140 144.1831 145.8524
##    25579    25581    25777    25819    26033    26084    26102    26143
## 150.4567 145.6546 142.1859 162.1724 153.6313 148.9712 149.2791 149.3824
##    26188    26191    26192    26272    26289    26335    26454    26507
## 145.8525 158.2216 140.3863 144.6777 150.5265 141.5064 149.6319 142.5110
##    26584    26720    26750    26830    26837    26907    27027    27038
## 137.0657 161.5266 141.5838 144.0420 145.3485 144.8223 144.0890 144.8602
##    27060    27061    27249    27332    27424    27492    27624    27692
## 139.0628 136.1009 144.5026 161.3455 149.4856 147.6048 160.2986 139.8605
##    27805    27859    27867    27898    28119    28208    28252    28287
## 138.4214 165.2132 143.3144 144.9601 149.4731 141.8806 141.7234 145.3629
##    28438    28603    28606    28620    28906    29024    29090    29312
## 145.3065 145.8424 144.2158 141.8986 141.6996 159.7901 145.1969 146.0260
##    29436    29545    29586    29715    29720    29806    29816    29943
## 142.0367 142.9036 142.8093 144.6024 138.1966 145.6565 140.8937 144.1728
##    29993    30193    30372    30374    30407    30634    30692    30695
## 145.8644 158.6130 160.1013 138.0249 143.7653 142.3417 164.8582 148.4301
##    30761    30764    30861    30892    30915    30966    31017    31024
## 143.5104 139.1832 143.2177 150.1973 140.6930 139.3449 143.5867 137.0716
##    31078    31125    31155    31157    31158    31187    31197    31236
## 170.3149 143.4629 143.5605 147.9243 143.0504 155.2772 145.5067 142.1932
##    31257    31294    31310    31324    31329    31342    31397    31418
## 147.0903 150.2054 152.4593 139.8022 152.6196 141.6044 149.6331 146.0343
##    31431    31466    31481    31499    31533    31536    31537    31586
## 143.5589 160.1024 137.8456 146.7918 140.8625 140.7631 136.3281 148.3441
##    31611    31624    31644    31662    31711    31717    31726    31733
## 143.8009 155.1385 153.4847 139.3136 141.4915 151.2841 147.9000 145.0801
##    31739    31753    31801    31829    31848    31870    31913    31973
## 148.2597 144.1164 141.1213 134.5516 147.6051 157.0001 144.3252 155.3291
##    31993    31997    32017    32105    32119    32186    32194    32204
## 147.2336 149.2080 143.8196 142.1021 148.6493 148.2349 139.9467 141.3282
##    32269    32305    32306    32313    32344    32385    32411    32412
## 144.3274 149.0227 153.4316 167.9414 170.0009 152.8446 139.3024 145.9180
##    32434    32489    32492    32538    32539    32551    32564    32571
## 138.7237 162.1213 147.7620 154.6511 171.2165 164.1709 161.0369 150.0367
##    32575    32627    32628    32631    32637    32680    32684    32706
## 142.9595 139.7706 151.4588 152.2318 143.4684 170.9206 149.2625 136.6467
##    32734    32751    32789    32797    32807    32808    32847    32880
## 142.8650 141.8349 144.8084 186.4304 148.2570 138.3936 143.4953 174.0514
##    32884    32900    32940    32954    32962    32975    32997    33005
## 139.7048 143.0687 146.2739 149.7473 137.9140 181.8741 139.6987 141.5290
##    33022    33036    33048    33073    33074    33075    33115    33171
## 155.8438 139.2576 171.5270 156.0838 152.9281 140.6615 159.8305 141.4761
##    33175    33183    33220    33263    33276    33335    33374    33395
## 157.7130 138.0722 180.7673 168.2946 148.7810 145.9395 149.8961 152.2600
##    33418    33449    33465    33470    33523    33544    33568    33580
## 136.8903 174.8756 140.7203 144.7557 148.5786 145.1909 141.5530 143.7710
##    33587    33614    33686    33796    33811    33910    33948    33963
## 142.3788 160.2335 137.8349 139.3193 142.0222 138.8930 147.7837 153.1302
##    34028    34072    34076    34173    34174    34201    34216    34311
## 141.3835 144.3109 137.5564 161.2333 155.6448 148.1104 175.3033 147.4600
##    34328    34360    34365    34397    34399    34520    34546    34698
## 141.3133 147.8111 137.6518 137.7301 140.8004 137.7625 156.1268 139.8055
##    35308    35335    35357    35384    35460    35475    35483    35492
## 148.5488 143.5994 168.4733 138.1660 149.9858 140.0414 145.7279 151.6926
##    35499    35537    35594    35624    35633    35658    35662    35697
## 136.9563 148.6361 145.3617 146.8186 135.3799 140.8711 144.8567 149.7023
##    35752    35774    35785    35786    35864    35877    35909    35975
## 136.1952 152.1965 146.3904 146.5661 148.4340 147.3322 138.4787 151.2694
##    35979    36025    36059    36060    36106    36145    36188    36191
## 151.9082 151.7379 149.4535 148.8682 152.6082 146.3284 152.9411 145.8934
##    36200    36208    36221    36225    36231    36293    36294    36301
## 134.3881 151.7598 144.3490 149.0642 143.9317 152.7437 152.3638 149.2806
##    36332    36352    36377    36419    36423    36440    36446    36479
## 149.8448 154.4191 148.0650 138.0429 149.9916 136.4958 146.8176 139.3849
##    36495    36511    36512    36518    36530    36586    36640    36690
## 141.8050 144.5828 149.2626 145.6607 142.5665 149.5980 148.8968 146.8113
##    36761    36865    36867    36897    36911    36925    36935    36950
## 145.2549 144.8536 146.2990 149.6750 146.9600 144.3983 137.5336 147.2280
##    36976    37015    37017    37023    37028    37041    37042    37064
## 149.2253 147.7066 143.4191 137.5262 141.8733 146.0095 145.4242 152.6010
##    37127    37146    37187    37192    37203    37204    37244    37257
## 148.7455 142.3025 142.3210 143.9520 146.5661 147.9108 146.3881 149.1506
##    37258    37259    37276    37395    37412    37434    37468    37492
## 151.6489 151.3053 144.4218 141.2927 144.2163 150.6819 154.0728 140.2533
##    37500    37503    37558    37562    37617    37624    37704    37758
## 148.5808 146.6042 140.1651 140.6642 138.3101 151.9656 150.3066 143.7001
##    37832    37864    37872    37889    37928    37966    38001    38018
## 149.3461 141.3510 140.7696 146.2214 145.6206 147.2779 144.4916 146.6828
##    38049    38064    38089    38101    38119    38253    38283    38305
## 144.4113 151.9541 139.9531 139.8033 149.2712 147.9665 141.8838 150.3340
##    38378    38406    38424    38455    38458    38470    38539    38552
## 141.5229 150.3910 148.3017 151.8954 150.4019 141.5827 149.7212 150.2197
##    38573    38584    38603    38642    38684    38689    38691    38706
## 148.7919 143.5370 148.3762 139.9666 143.4790 135.9167 138.2043 146.6319
##    38870    38915    38924    38947    38979    39005    39016    39033
## 145.6821 145.3579 150.2763 147.4994 145.4934 149.9408 147.6049 153.1687
##    39041    39048    39086    39109    39119    39164    39171    39177
## 150.2506 150.7866 140.5113 146.5711 152.8709 142.5332 138.9623 157.0628
##    39200    39214    39234    39272    39319    39320    39321    39325
## 152.1892 141.7617 148.4126 150.1686 145.3113 138.6672 138.5791 147.0976
##    39365    39366    39447    39454    39455    39483    39492    39505
## 153.2610 148.6977 137.1865 141.8188 138.2798 144.8470 146.1394 149.7452
##    39509    39521    39528    39539    39578    39598    39670    39694
## 141.1452 141.9777 156.8062 152.3609 150.8465 148.0375 146.6550 149.0611
##    39700    39704    39723    39749    39772    39773    39823    39828
## 146.1620 149.6168 144.0671 157.6801 145.9359 146.9668 148.5348 142.7768
##    39836    39884    39887    39934    40024    40033    40055    40057
## 153.9008 146.5758 151.5270 144.0776 153.0944 152.4936 147.3484 153.0556
##    40062    40082    40083    40095    40101    40127    40147    40193
## 144.9591 143.3022 149.4520 155.5823 143.1974 144.4987 148.3910 139.5410
##    40196    40229    40254    40282    40314    40342    40351    40373
## 139.6858 144.4565 140.6530 149.3498 143.5938 151.6311 144.0050 138.3194
##    40382    40427    40443    40445    40483    40550    40561    40608
## 149.2404 141.1128 149.8540 140.7997 145.9700 146.1471 152.3492 144.0397
##    40656    40676    40735    40748    40751    40769    40845    40857
## 146.9627 143.8545 159.4915 142.9203 145.7534 146.7870 144.5389 142.7664
##    40863    40868    40869    40903    40904    40915    40930    40966
## 154.4310 150.8446 156.5799 150.7751 145.4284 147.6965 151.1472 153.4169
##    40999    41002    41004    41009    41020    41073    41075    41393
## 178.4611 177.6483 146.9733 149.9661 150.5232 143.6760 146.0830 143.9428
##    41397    41402    41447    41451    41502    41517    41530    41533
## 143.0111 146.1973 147.0603 143.7682 142.7848 147.1286 143.5235 143.9995
##    41553    41586    41601    41621    41629    41636    41665    41681
## 133.4721 143.8011 144.7750 144.5840 143.2209 149.1044 141.0599 141.4197
##    41683    41689    41696    41708    41724    41734    41744    41768
## 144.1161 142.4596 146.4495 145.3600 145.7481 148.4298 140.4056 145.3724
##    41798    41803    41832    41858    41865    41886    41888    41903
## 144.0302 147.2187 143.5581 137.1161 147.5639 145.1695 145.8660 141.5524
##    41959    41999    42007    42026    42136    42171    42173    42219
## 143.2206 143.5186 146.5980 152.1322 150.8867 146.1237 141.3183 144.1628
##    42220    42221    42238    42249    42278    42304    42315    42318
## 143.0457 146.1935 142.7304 149.0157 148.9960 148.4858 143.4943 136.8899
##    42332    42363    42375    42383    42394    42441    42444    42450
## 145.8214 144.8168 142.6323 145.8643 143.4587 152.8201 143.8043 151.4090
##    42478    42508    42562    42623    42624    42637    42643    42686
## 143.4171 144.2594 143.6039 144.3631 145.7912 146.9618 144.7507 143.5709
##    42717    42720    42723    42747    42767    42794    42839    42857
## 140.4223 146.6708 136.8505 148.1545 145.3125 141.6104 146.9106 147.0698
##    42861    42882    42901    42919    42924    42993    43029    43031
## 146.8517 143.8022 142.9363 143.0800 141.5366 143.8275 142.3281 143.4536
##    43035    43058    43071    43097    43102    43128    43189    43211
## 144.2789 140.7307 143.3819 142.2736 140.4528 142.7265 144.2722 144.1047
##    43250    43304    43311    43314    43360    43392    43402    43406
## 142.6626 147.7310 130.3205 130.5732 146.0731 149.5752 146.0102 145.0029
##    43411    43448    43455    43458    43486    43502    43512    43531
## 146.3084 143.1879 141.6919 143.0002 144.0903 141.6999 140.9036 142.1010
##    43545    43580    43607    43609    43623    43634    43736    43737
## 143.0541 143.6380 140.9286 148.2061 144.8551 142.4482 144.0515 141.7888
##    43755    43772    43861    43866    43870    43882    43927    43952
## 146.5317 144.0082 142.4462 142.2464 141.3951 147.4643 148.0039 141.9579
##    43975    43977    43982    44039    44052    44055    44068    44070
## 142.1328 145.8429 146.7003 149.1274 148.0597 155.3901 147.8351 151.3060
##    44074    44081    44125    44135    44138    44210    44217    44228
## 144.8078 150.3081 144.6165 141.4630 146.6647 142.8505 142.6996 143.3495
##    44291    44302    44311    44331    44345    44355    44379    44390
## 144.9493 149.0470 145.8140 146.4749 142.4414 149.9417 143.3547 141.6759
##    44409    44437    44455    44484    44506    44533    44536    44575
## 142.8360 141.8683 143.2495 144.6141 140.8885 148.4126 144.8682 144.1900
##    44591    44595    44600    44610    44628    44632    44830    45938
## 146.2595 143.2035 136.7051 148.0075 143.5705 143.8693 157.4815 154.8449
##    46715    46754    46772    46774    46786    46797    46812    46839
## 144.5436 140.6746 146.3907 138.6981 142.7423 149.0800 141.8876 142.8135
##    46893    46895    46952    46957    46987    46992    47003    47074
## 140.1236 147.4454 152.8384 151.4176 142.8053 149.5522 142.5002 140.6121
##    47077    47081    47165    47178    47211    47229    47274    47277
## 148.6692 145.6160 143.5397 142.1647 140.1540 146.1860 140.9740 142.9104
##    47283    47390    47405    47421    47433    47437    47481    47505
## 152.1357 139.7618 144.9408 160.7338 147.5283 144.4026 154.7748 146.7894
##    47541    47546    47555    47596    47603    47604    47609    47612
## 145.0594 141.2430 140.3250 152.6617 151.9346 153.0945 152.5085 153.0283
##    47615    47632    47637    47665    47666    47669    47670    47684
## 147.3011 149.9165 145.6653 151.4962 139.8078 146.5069 150.0749 145.0289
##    47697    47713    47720    47747    47753    47760    47791    47792
## 149.8588 140.8267 149.8609 152.4368 149.0813 143.6400 153.1562 150.4536
##    47824    47831    47876    47892    48041    48049    48052    48069
## 144.3355 146.3464 181.6364 152.1738 167.4937 158.8977 167.1406 160.6597
##    48073    48087    48116    48161    48164    48217    48361    48366
## 168.0760 154.7009 158.0606 155.8736 161.4270 157.8235 156.7409 167.0752
##    48391    48404    48407    48409    48422    48423    48672    48711
## 157.4299 156.9699 152.6466 157.9641 160.9549 161.9799 138.4384 145.8964
##    48716    48729    48747    48778    48823    48828    48885    48910
## 146.0887 141.8667 144.6109 150.1768 146.3380 144.4168 148.7233 145.3995
##    48919    48971    49016    49119    49134    49146    49257    49272
## 140.4414 143.6516 145.3698 145.8871 148.1323 150.6208 147.5814 154.7132
##    49277    49283    49290    49308    49309    49327    49343    49355
## 150.7096 143.0909 142.9302 153.0324 152.4138 156.5294 153.1589 156.7496
##    49365    49368    49568    49603    49617    49636    49658    49675
## 150.4841 152.6958 145.2282 146.3590 147.0334 143.8089 151.9698 142.6022
##    49689
## 157.0084
```

```
res.impca$completeObs[ outs_mpg$extreme, "mpg" ]
```

```
##     5148     6308     8876    11341    11561    14452    14579    14730
## 42.81214 44.47775 59.83923 53.13376 38.57218 58.87374 37.31721 49.91053
##    16294    16491    16612    16900    16987    17777    18477    18491
## 54.51028 58.57037 52.81379 55.12006 54.71413 51.01415 48.46714 48.93926
##    18584    18660    18820    19986    20034    20160    20593    21018
## 57.12398 56.19816 55.04812 54.86740 56.46759 54.91984 56.13217 56.57459
##    21110    21896    24955    25579    26143    26289    27591    30892
## 61.36529 53.51528 55.02701 54.25338 54.45418 54.96793 45.70120 55.09093
##    32186    32415    33022    34344    36701    37259    37337    37835
## 54.60482 58.12067 58.75804 45.60711 56.60592 57.25503 57.76944 57.34651
##    38305    40342
## 56.46981 55.49111
```

```
df[, vars_num] <- res.impca$completeObs

summary(df$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   4.000   3.802   5.000  14.000
```

```
df[outs_age$extreme,"age"]
```

```
##  [1] 6.602015 8.673582 8.851468 5.201589 4.711389 7.264752 7.145131 7.729946
##  [9] 8.288108 5.121137 8.194822 7.444977 6.846731 8.919170 9.525526 6.001603
## [17] 6.962624 9.070462 8.499986 9.926537 3.672846 4.947717
```

```r
df[outs_mpg$extreme,"mpg"]
```

```
##  [1] 42.81214 44.47775 59.83923 53.13376 38.57218 58.87374 37.31721 49.91053
##  [9] 54.51028 58.57037 52.81379 55.12006 54.71413 51.01415 48.46714 48.93926
## [17] 57.12398 56.19816 55.04812 54.86740 56.46759 54.91984 56.13217 56.57459
## [25] 61.36529 53.51528 55.02701 54.25338 54.45418 54.96793 45.70120 55.09093
## [33] 54.60482 58.12067 58.75804 45.60711 56.60592 57.25503 57.76944 57.34651
## [41] 56.46981 55.49111
```

2.2 Factores

```r
summary(df[,vars_cat])
```

```
##                model            year                 transmission
##  VW- Golf         : 471   2019    :1563   f.Trans-Manual    :1787
##  Mercedes- C Class: 376   2017    : 895   f.Trans-SemiAuto :1896
##  VW- Polo         : 337   2016    : 885   f.Trans-Automatic:1317
##  BMW- 3 Series    : 274   2018    : 460
##  Mercedes- A Class: 260   2015    : 424
##  Mercedes- E Class: 201   2020    : 316
##  (Other)          :3081   (Other): 457
##          fuelType      manufacturer  engineSize_factor
##  f.Fuel-Diesel:2868   Audi     :1066   2       :2142
##  f.Fuel-Petrol:2060   BMW      :1107   1.5     : 554
##  f.Fuel-Hybrid:  59   Mercedes:1333   3       : 512
##  NA's         :  13   VW      :1494   2.1     : 412
##                                        1       : 365
##                                        (Other):1006
##                                        NA's    :   9
```

```r
res.immca <- imputeMCA(df[,vars_cat],ncp=10)
summary(res.immca$completeObs)
```

```
##                model            year                 transmission
##  VW- Golf         : 471   2019    :1563   f.Trans-Manual    :1787
##  Mercedes- C Class: 376   2017    : 895   f.Trans-SemiAuto :1896
##  VW- Polo         : 337   2016    : 885   f.Trans-Automatic:1317
##  BMW- 3 Series    : 274   2018    : 460
##  Mercedes- A Class: 260   2015    : 424
##  Mercedes- E Class: 201   2020    : 316
##  (Other)          :3081   (Other): 457
##          fuelType      manufacturer  engineSize_factor
##  f.Fuel-Diesel:2873   Audi     :1066   2       :2149
##  f.Fuel-Petrol:2068   BMW      :1107   1.5     : 554
##  f.Fuel-Hybrid:  59   Mercedes:1333   3       : 513
##                       VW      :1494   2.1     : 412
##                                        1       : 366
##                                        1.6     : 345
##                                        (Other): 661
```

```r
df[,vars_cat]<-res.immca$completeObs
```

3 Discretización de variables numéricas en factores (por cuantiles)

```r
vars_num
```

```
## [1] "price"   "mileage" "tax"     "mpg"     "age"
```

3.1 Identificar los outliers multivariantes

4 Relaciones entre variables

4.1 Correlaciones

```
# No se que quiere decir exactamente... Pensaba que esto venía por el profiling...
```

4.2 Texto que no sé que quiere decir

5 Profiling

En esta parte es donde empieza a entrar todo lo de quali y quanti

5.1 Target numérico (Price)

5.2 Target factor (AUDI)