

# USED CAR PRICES: CASE STUDY

DATA ANALYSIS AND INFORMATION EXPLOITATION

2021-2022 Q2



Xavier Alaman Obregon  
Miquel Parra Núñez

# **Index**

Deliverable 1: Data Processing, Description, Validation and Profiling	3
Deliverable 2: PCA, CA and Clustering	46
Deliverable 3: General and Binary/Logistic Regression Models	123

# USED CAR PRICES CASE STUDY

Deliverable I: Data Processing, Description, Validation and Profiling

Miquel Parra i Xavier Alaman

March 26, 2022

## Contents

<b>1 R libraries imports, useful functions and data loading</b>	<b>2</b>
1.1 Load required packages . . . . .	2
1.2 Sample load . . . . .	3
1.3 Useful functions . . . . .	3
<b>2 Data Description</b>	<b>3</b>
2.1 Original variables description . . . . .	4
<b>3 Univariate Descriptive Analysis</b>	<b>4</b>
3.1 Numeric variables . . . . .	5
3.1.1 years_sell . . . . .	5
3.1.2 price . . . . .	5
3.1.3 mileage . . . . .	6
3.1.4 mpg . . . . .	7
3.1.5 tax . . . . .	8
3.2 Factors . . . . .	9
3.2.1 model . . . . .	9
3.2.2 year . . . . .	10
3.2.3 transmission . . . . .	11
3.2.4 fuelType . . . . .	12
3.2.5 engineSize . . . . .	13
3.2.6 manufacturer . . . . .	14
<b>4 Data Quality Report</b>	<b>15</b>
4.1 Initialization of counts for missings, outliers and errors. . . . .	15
4.2 Errors . . . . .	16
4.3 Univariate Outliers . . . . .	16
4.3.1 price . . . . .	16
4.3.2 mileage . . . . .	17
4.3.3 tax . . . . .	17
4.3.4 mpg . . . . .	18
4.3.5 years_sell . . . . .	19
4.4 Number of errors, missings and outliers for individual and variable . . . . .	20

4.4.1	Number of missing values of each variable . . . . .	20
4.4.2	Number of errors per each variable . . . . .	20
4.4.3	Number of outliers per each variable . . . . .	20
4.4.4	Number of missing values individual . . . . .	21
4.4.5	Number of errors individual . . . . .	21
4.4.6	Number of outliers individual . . . . .	21
4.4.7	New variable adding the total number missing values, outliers and errors . . . . .	22
4.5	Imputation . . . . .	22
4.5.1	Imputation of numeric variables . . . . .	22
4.5.1.1	mileage . . . . .	22
4.5.1.2	tax . . . . .	23
4.5.1.3	mpg . . . . .	24
4.5.2	Imputation of factor variables . . . . .	25
4.5.2.1	engineSize . . . . .	26
4.6	Correlation of numeric variables with MOE . . . . .	27
4.7	Discretization . . . . .	27
4.7.1	price variable . . . . .	27
4.7.2	mileage variable . . . . .	28
4.7.3	tax variable . . . . .	28
4.7.4	mpg variable . . . . .	28
4.7.5	years_sell variable . . . . .	29
4.8	Definition of binary outcome: Audi . . . . .	29
4.9	Multivariate outliers . . . . .	31
4.10	Profiling . . . . .	34

# 1 R libraries imports, useful functions and data loading

In this first section we will load all required packages and libraries, declare additional functions, and load our data.

## 1.1 Load required packages

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

requiredPackages <- c("effects","FactoMineR","car",
  "factoextra","RColorBrewer","ggplot2","dplyr","ggmap",
  "ggthemes","knitr","treemap")

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

#verify they are loaded
search()
```

## 1.2 Sample load

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

# Clean workspace
rm(list=ls())

# Users file path
miquel_fp <- "C:/Users/Miquel/Documents/GitHub/ADEI/"
xavi_fp <- "~/Documents/FIB/ADEI/ADEI/"
filepath <- xavi_fp
filepath <- miquel_fp

# Set working directory
setwd(filepath)

# Load data from file
load(paste0(filepath, "MyOldCars-Raw.RData"))

# Index reset
row.names(df) <- NULL
```

## 1.3 Useful functions

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

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

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

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

## 2 Data Description

During this project we will be working with a subset of the pre-treated original dataset “Uk used car dataset”. A sample of 5000 cars has been randomly selected from Mercedes, BMW, Volkswagen and Audi manufacturers and stored into a RData file *MyOldCars-Raw.RData*.

## 2.1 Original variables description

- **model:** Car model.
- **year:** Car registration year.
- **price:** Car price in £.
- **transmission:** Type of transmission [“Manual”, “Automatic”, “Semi-Auto”].
- **mileage:** Distance used, accumulated miles.
- **fuelType:** Type of engine fuel [“Petrol”, “Diesel”, “Hybrid”, “Other” ].
- **tax:** Applied road tax.
- **mpg:** Miles per gallon.
- **engineSize:** Engine size in liters. The cars with engine size 0 are in fact electric cars, nevertheless Mercedes C class, and other given cars are not electric cars, so data imputation is required.
- **manufacturer:** Car manufacturer [“Audi”, “BMW”, “Mercedes”, “VW”].

```
summary(df)
```

```
##      model          year        price      transmission
##  Length:5000    Min.   :1999   Min.   : 650  Length:5000
##  Class  :character  1st Qu.:2016   1st Qu.:13995  Class  :character
##  Mode   :character  Median :2017   Median :19498  Mode   :character
##                  Mean   :2017   Mean   :21470
##                  3rd Qu.:2019   3rd Qu.:26039
##                  Max.  :2020   Max.  :109990
##      mileage        fuelType        tax        mpg
##  Min.   : 4   Length:5000    Min.   : 0.0  Min.   : 8.80
##  1st Qu.:5999  Class  :character  1st Qu.:125.0 1st Qu.:44.80
##  Median :16619  Mode   :character  Median :145.0  Median :53.30
##  Mean   :23312
##  3rd Qu.:33834
##  Max.  :153000
##      engineSize    manufacturer
##  Min.   :0.000  Length:5000
##  1st Qu.:1.500  Class  :character
##  Median :2.000  Mode   :character
##  Mean   :1.917
##  3rd Qu.:2.000
##  Max.  :6.600
```

```
head(df, 3)
```

```
##   model year price transmission mileage fuelType tax  mpg engineSize
## 1   A1 2016 11000      Manual  29946  Petrol  30 55.4      1.4
## 2   A3 2015 10200      Manual  46112  Petrol  20 60.1      1.4
## 3   A4 2017 18500  Automatic 17418  Diesel 145 62.8      2.0
##   manufacturer
## 1       Audi
## 2       Audi
## 3       Audi
```

## 3 Univariate Descriptive Analysis

In this step of the process original numeric variables corresponding to qualitative concepts have to be converted to factors. New factors grouping original levels will be considered very positively.

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

For each variable we will perform the necessary transformations and also visualize its distribution.

### 3.1 Numeric variables

#### 3.1.1 years\_sell

From `year` we can create a new variable called `years_sell`. It will contain the same information as `year` but it will give more valuable information for a human understanding perspective.

`years_sell` represents the years the car has been sold.

```
df$years_sell <- 2022 - df$year
summary(df$years_sell)

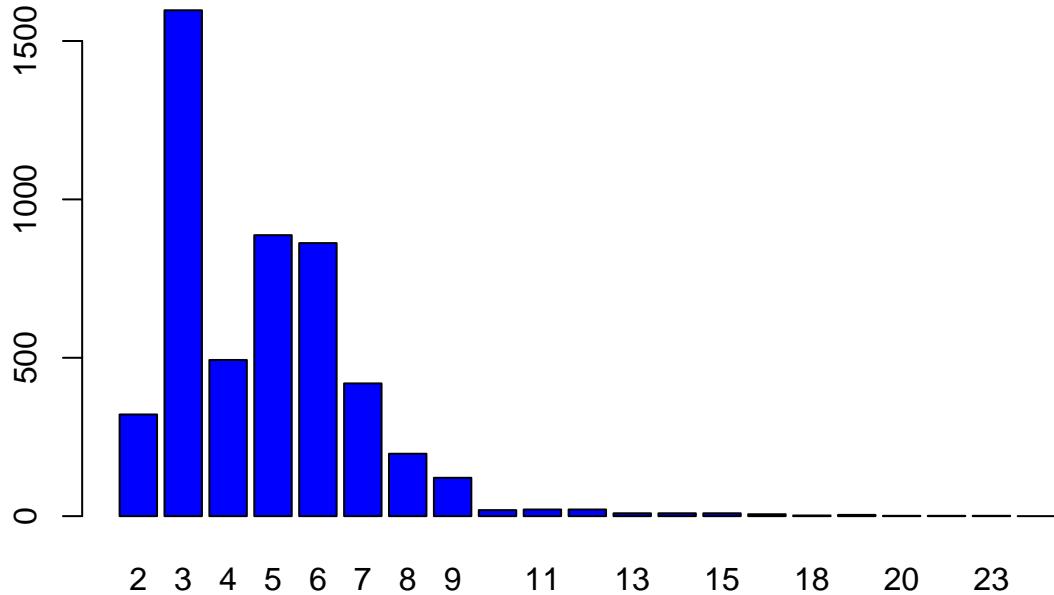
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##  2.000   3.000   5.000   4.787   6.000  23.000

table(df$years_sell,useNA="always")

##      2      3      4      5      6      7      8      9      10     11     12     13     14     15     16     18
## 321 1597 493 887 862 419 197 121 19 21 21 9 9 9 6 2
## 19 20 21 23 <NA>
## 4 1 1 1 0

barplot(table(df$years_sell,useNA="always"), main = "years_sell barplot", col = "blue")
```

**years\_sell barplot**

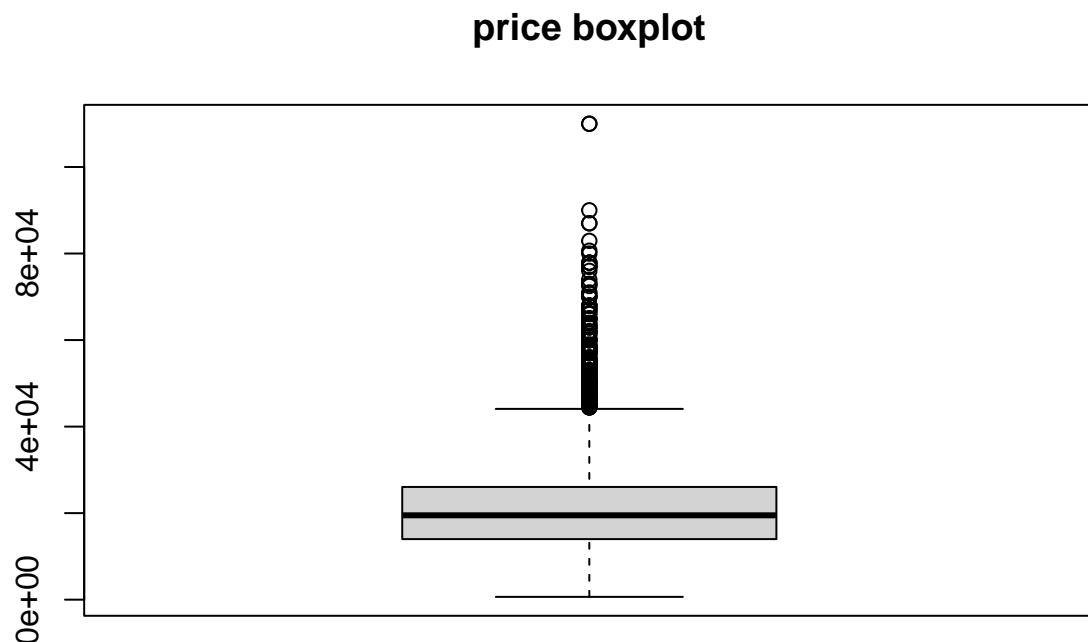


#### 3.1.2 price

```
summary(df$price)

##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##  650 13995 19498 21470 26039 109990
```

```
boxplot(df$price, main="price boxplot")
```



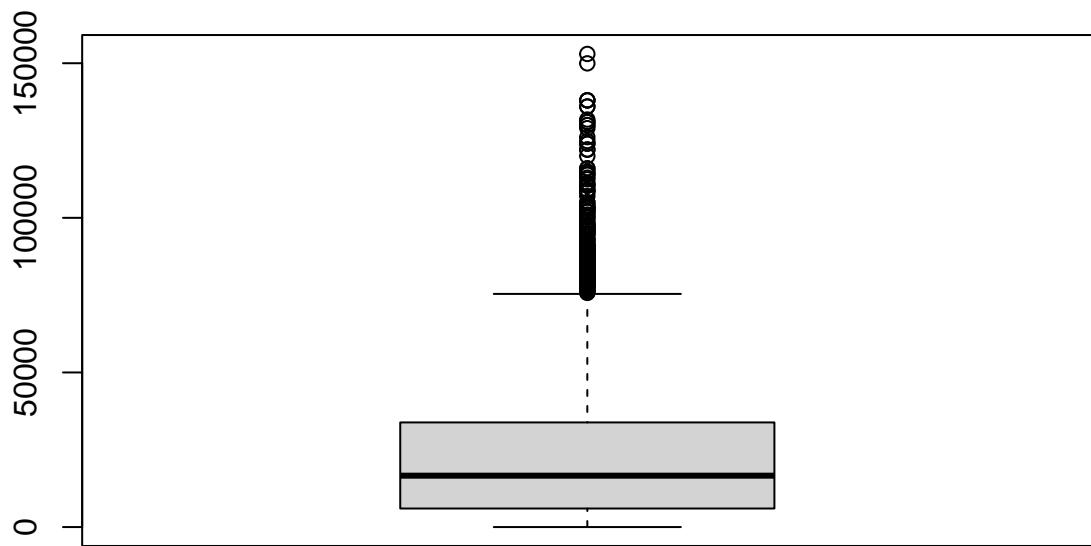
### 3.1.3 mileage

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

```
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##        4    5999  16619  23312  33834 153000
```

```
boxplot(df$mileage, main="mileage boxplot")
```

**mileage boxplot**

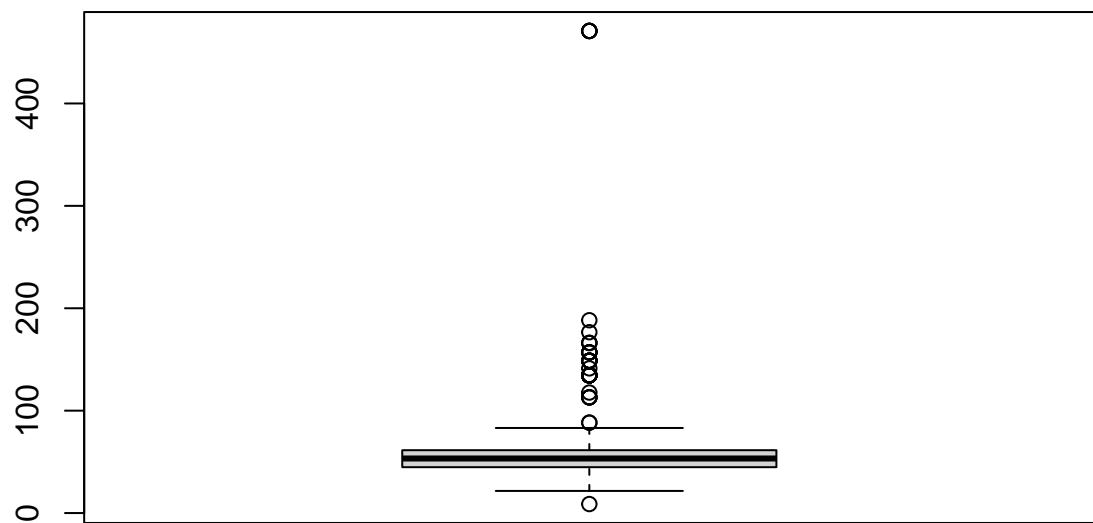


### 3.1.4 mpg

```
summary(df$mpg)  
  
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.  
##     8.80    44.80   53.30    53.89   61.40   470.80
```

```
boxplot(df$mpg, main="mpg boxplot")
```

mpg boxplot

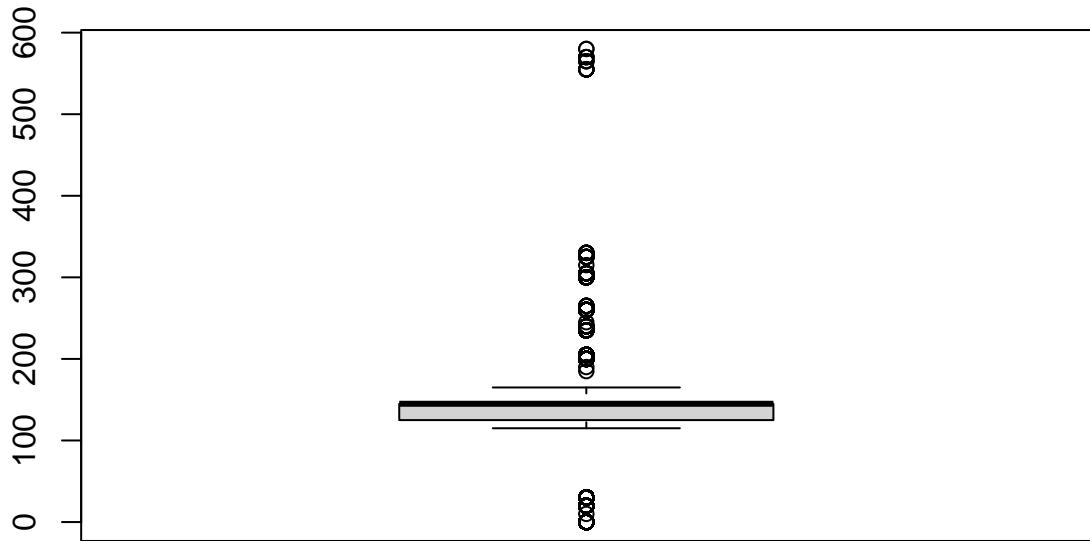


### 3.1.5 tax

```
summary(df$tax)  
  
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.  
##      0.0   125.0   145.0   125.3   145.0   580.0
```

```
boxplot(df$tax, main="tax boxplot")
```

**tax boxplot**



## 3.2 Factors

### 3.2.1 model

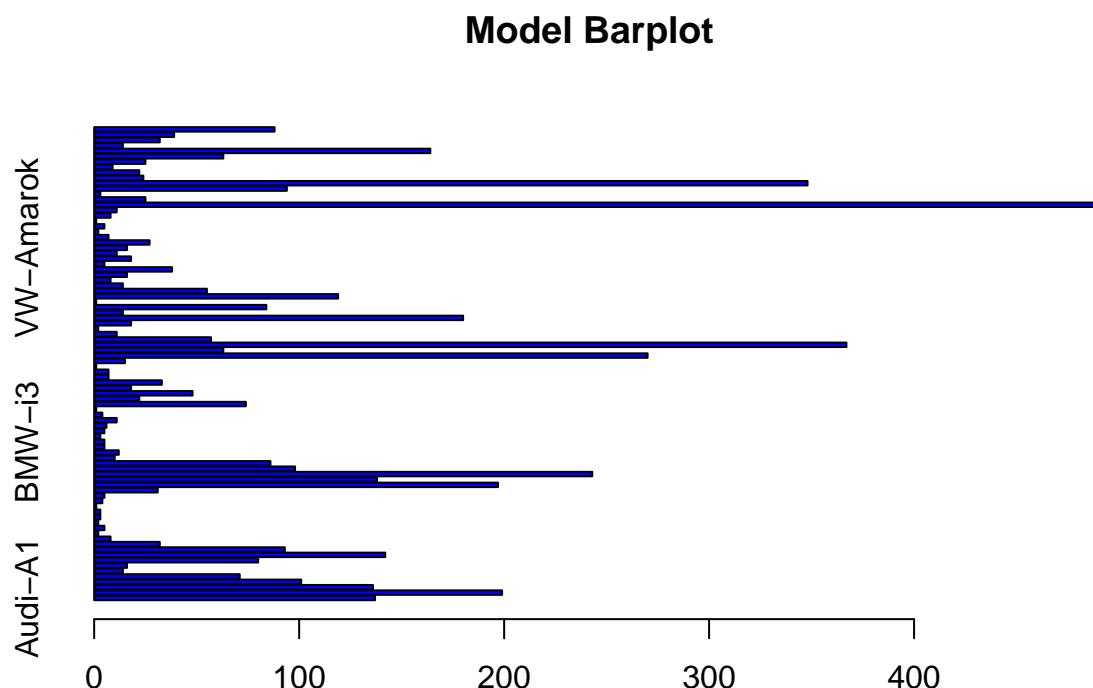
```
df$model<-factor(paste0(trimws(df$manufacturer),"-",trimws(df$model)))

summary(df$model)
```

##	Audi-A1	Audi-A3	Audi-A4	Audi-A5
##	137	199	136	101
##	Audi-A6	Audi-A7	Audi-A8	Audi-Q2
##	71	14	16	80
##	Audi-Q3	Audi-Q5	Audi-Q7	Audi-Q8
##	142	93	32	8
##	Audi-R8	Audi-RS3	Audi-RS4	Audi-RS5
##	2	5	2	3
##	Audi-RS6	Audi-S3	Audi-S4	Audi-SQ5
##	3	1	4	5
##	Audi-TT	BMW-1 Series	BMW-2 Series	BMW-3 Series
##	31	197	138	243
##	BMW-4 Series	BMW-5 Series	BMW-6 Series	BMW-7 Series
##	98	86	10	12
##	BMW-8 Series	BMW-i3	BMW-i8	BMW-M2
##	5	5	3	5
##	BMW-M3	BMW-M4	BMW-M5	BMW-M6
##	6	11	4	1
##	BMW-X1	BMW-X2	BMW-X3	BMW-X4
##	74	22	48	18
##	BMW-X5	BMW-X6	BMW-X7	BMW-Z3
##	33	7	7	1
##	BMW-Z4	Mercedes-A Class	Mercedes-B Class	Mercedes-C Class
##	15	270	63	367
##	Mercedes-CL Class	Mercedes-CLA Class	Mercedes-CLC Class	Mercedes-CLS Class

##	57	11	2	18
##	Mercedes-E Class	Mercedes-GL Class	Mercedes-GLA Class	Mercedes-GLB Class
##	180	14	84	1
##	Mercedes-GLC Class	Mercedes-GLE Class	Mercedes-GLS Class	Mercedes-M Class
##	119	55	14	8
##	Mercedes-S Class	Mercedes-SL CLASS	Mercedes-SLK	Mercedes-V Class
##	16	38	5	18
##	Mercedes-X-CLASS	VW-Amarok	VW-Arteon	VW-Beetle
##	11	16	27	7
##	VW-Caddy Life	VW-Caddy Maxi Life	VW-California	VW-Caravelle
##	2	5	1	8
##	VW-CC	VW-Golf	VW-Golf SV	VW-Jetta
##	11	488	25	3
##	VW-Passat	VW-Polo	VW-Scirocco	VW-Sharan
##	94	348	24	22
##	VW-Shuttle	VW-T-Cross	VW-T-Roc	VW-Tiguan
##	9	25	63	164
##	VW-Tiguan Allspace	VW-Touareg	VW-Touran	VW-Up
##	14	32	39	88

```
barplot(summary(df$model), main = "Model Barplot", col = "blue", horiz=TRUE)
```



### 3.2.2 year

As you could imagine the distribution of years and years\_sell is the same, but moved from right to left as a result of the subtract operation. We have considered to join cars from year 1999 to 2009 because they are residual values, i.e. the amount of individuals per each one (year) is not representative enough.

```
df[which(df$year<=2009), "year"] <- "2009 or before"

df$year <- factor(df$year)
df$year <- factor(df$year, labels = paste0("f.Year-",levels(df$year)))

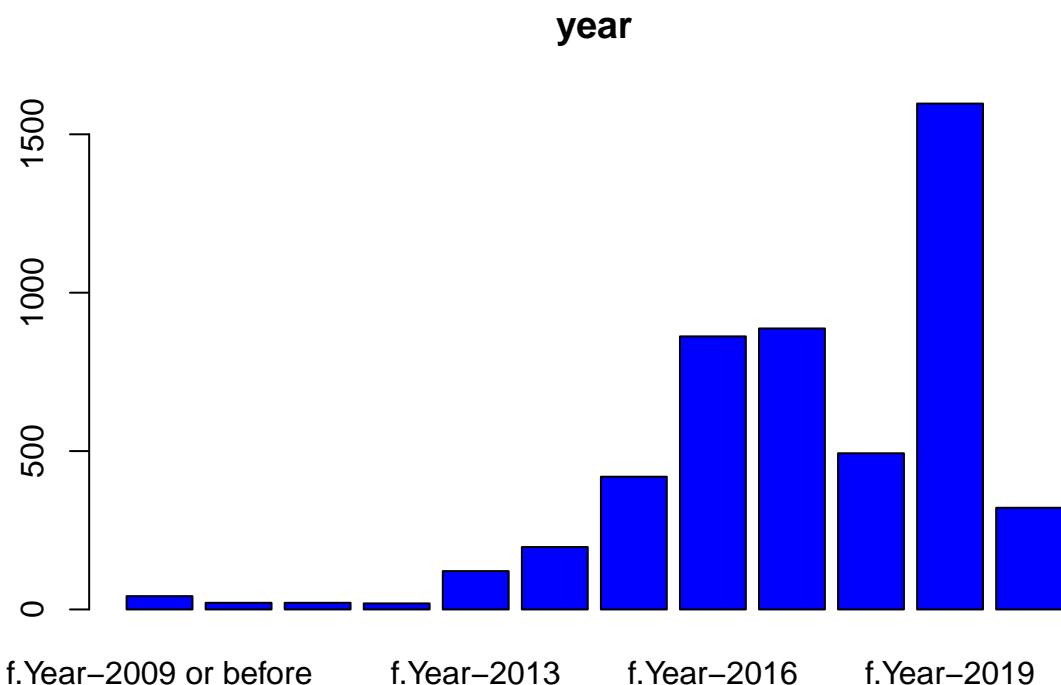
summary(df$year)
```

```

## f.Year-2009 or before          f.Year-2010          f.Year-2011
##                      42                      21                      21
## f.Year-2012          f.Year-2013          f.Year-2014
##                      19                      121                     197
## f.Year-2015          f.Year-2016          f.Year-2017
##                     419                     862                     887
## f.Year-2018          f.Year-2019          f.Year-2020
##                     493                    1597                     321

```

```
barplot(summary(df$year), main = "year", col = "blue")
```



### 3.2.3 transmission

```

df$transmission<-factor(df$transmission)
df$transmission <- factor(df$transmission, levels = c("Manual", "Semi-Auto", "Automatic"), labels = paste0
summary(df$transmission)

```

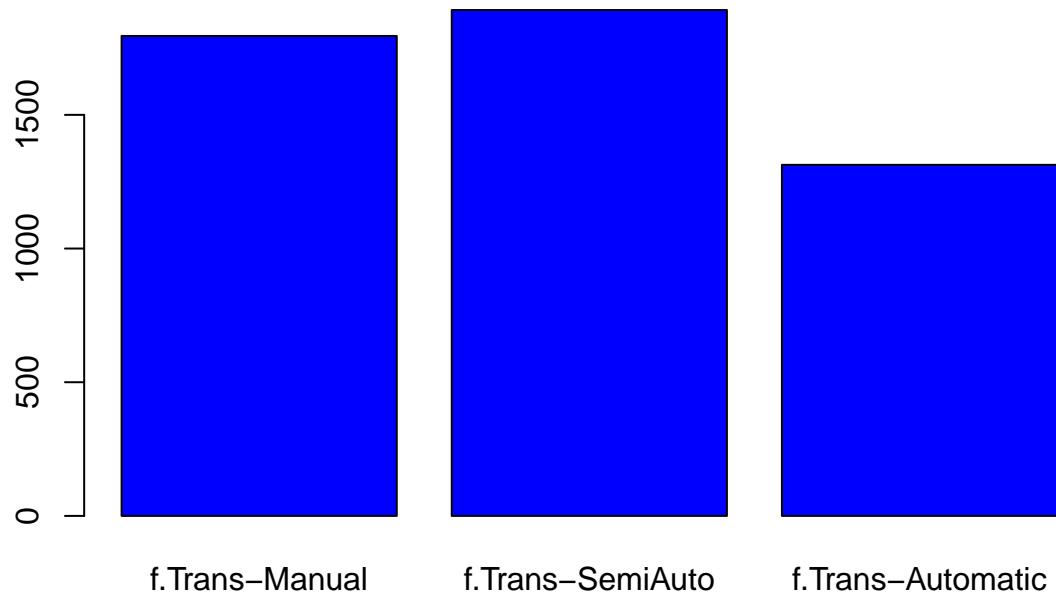
```

##      f.Trans-Manual  f.Trans-SemiAuto f.Trans-Automatic
##                      1795                  1892                  1313

```

```
barplot(summary(df$transmission), main = "Transmission Barplot", col = "blue")
```

## Transmission Barplot



### 3.2.4 fuelType

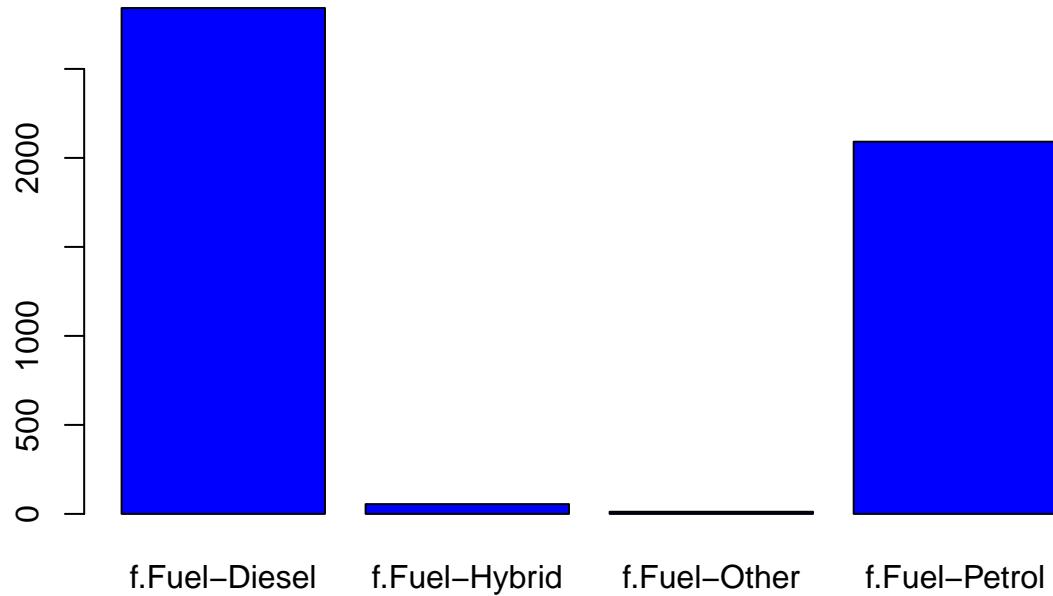
```
df$fuelType<-factor(df$fuelType)
df$fuelType <- factor(df$fuelType, labels = paste0("f.Fuel-",levels(df$fuelType)))

summary(df$fuelType)

## f.Fuel-Diesel f.Fuel-Hybrid f.Fuel-Other f.Fuel-Petrol
##           2842           55           12          2091

barplot(summary(df$fuelType), main = "FuelType Barplot", col = "blue")
```

## FuelType Barplot



### 3.2.5 engineSize

We have considered to join cars with engineSize 4.2, 4.4, 4.7, 5, 5.2, 5.5, 6.2 and 6.6 because they are residual values, i.e. the amount of individuals per each one (engineSize) is not representative enough.

```
df[which(df$engineSize>=4.2),"engineSize"] <- "4.2 or more"

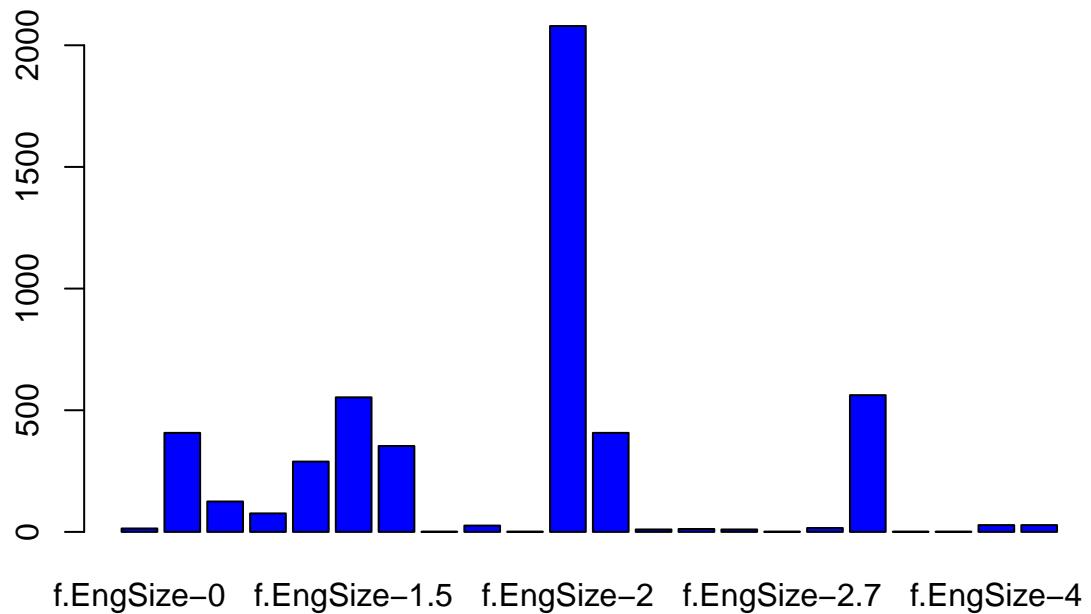
df$engineSize<-factor(df$engineSize)
df$engineSize <- factor(df$engineSize, labels = paste0("f.EngSize-",levels(df$engineSize)))

summary(df$engineSize)

##          f.EngSize-0          f.EngSize-1          f.EngSize-1.2
##                14                407                125
##          f.EngSize-1.3          f.EngSize-1.4          f.EngSize-1.5
##                  76                289                553
##          f.EngSize-1.6          f.EngSize-1.7          f.EngSize-1.8
##                 353                  1                  26
##          f.EngSize-1.9          f.EngSize-2          f.EngSize-2.1
##                  1                2079                407
##          f.EngSize-2.2          f.EngSize-2.3          f.EngSize-2.5
##                 10                  12                  10
##          f.EngSize-2.7          f.EngSize-2.9          f.EngSize-3
##                  1                  16                  562
##          f.EngSize-3.2          f.EngSize-3.5          f.EngSize-4
##                  1                  1                  28
## f.EngSize-4.2 or more
##                      28

barplot(summary(df$engineSize), main = "EngineSize Barplot", col = "blue")
```

### EngineSize Barplot



#### 3.2.6 manufacturer

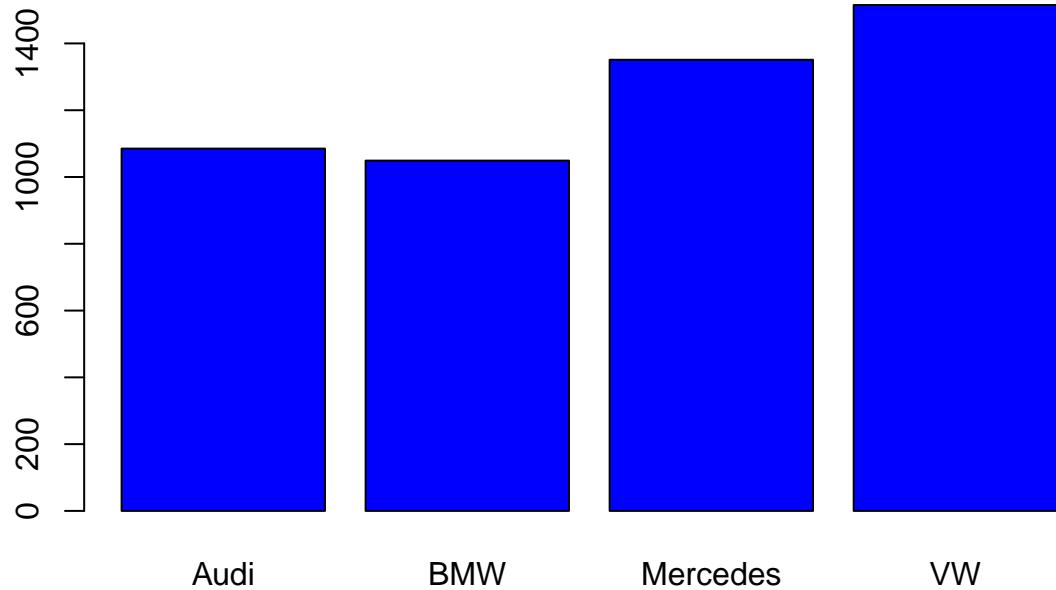
```
df$manufacturer<-factor(df$manufacturer)
df$manufacturer <- factor(df$manufacturer, labels = levels(df$manufacturer))

summary(df$manufacturer)
```

```
##      Audi      BMW Mercedes      VW
##      1085     1049     1351    1515
```

```
barplot(summary(df$manufacturer), main = "Manufacturer Barplot", col = "blue")
```

## Manufacturer Barplot



## 4 Data Quality Report

### 4.1 Initialization of counts for missings, outliers and errors.

```
#####
imis<-rep(0,nrow(df)) # rows - cars
jmis<-rep(0,ncol(df)) # columns - variables
#####
mis1<-countNA(df) #There are no missings at the beginning

# Number of missings for the current set of cars
sum(mis1$mis_ind)

## [1] 0

# Number of missings for the current set of variables
sum(mis1$mis_col)

## [1] 0

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

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

As you can see from the previous stats there are no missings in the variables for the random data subset.

## 4.2 Errors

After the first analysis of the samples and the provided documentation of the dataset we could say that the only visible errors are in the engineSize variable.

Engine size equal to zero is considered as an electrical vehicle so this error in the data needs to be considered and treated properly.

```
sel<-which(df$engineSize==0 & df$fuelType!="f.Fuel-Electric")
ierrs[sel]<-ierrs[sel]+1
df[sel,"engineSize"]<-NA
selmiss <- sel
jerrs[9] <- length(sel)
```

## 4.3 Univariate Outliers

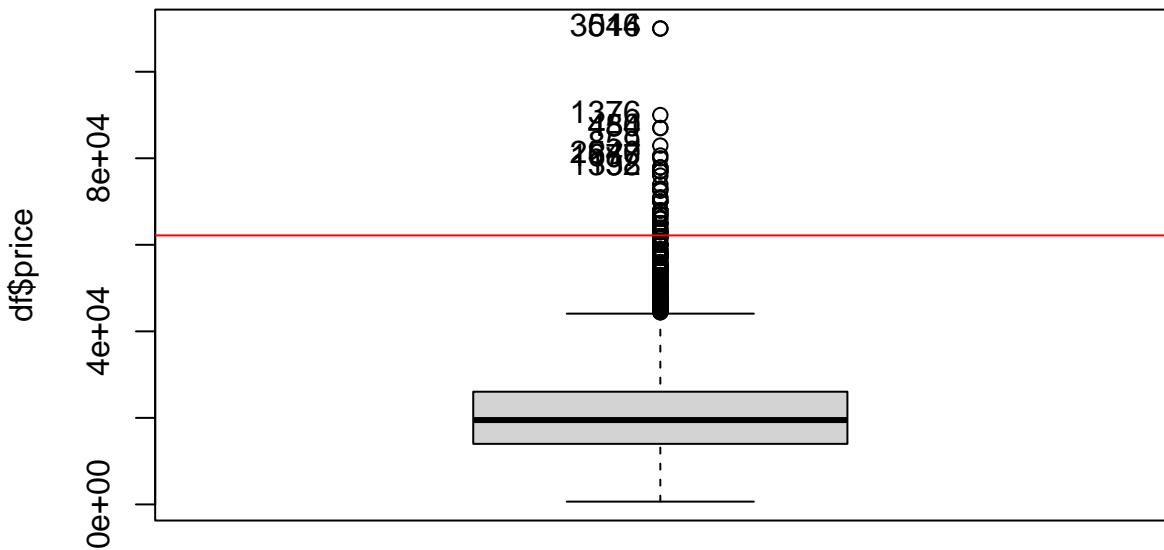
For each variable, we have executed calcQ in order to find the severe/extreme outliers (lower and upper). Then, we have recodified to/with NA the value of the variable of each individual with a value in the variable less than the under severe outlier or greater than the upper severe outlier to later apply imputation.

### 4.3.1 price

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

```
## [1] 514 3046 1376 450 484 859 2540 1677 192 1338
```

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



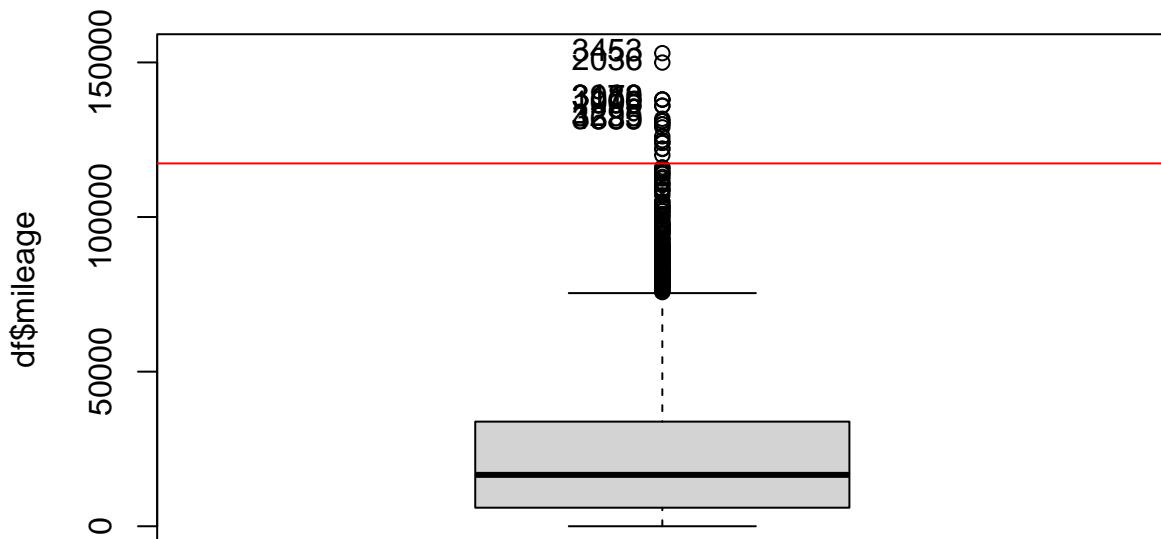
```
# Outliers:
llout_price<-which((df$price<var_out$souti)|(df$price>var_out$souts))#souts abline
iouts[llout_price]<-iouts[llout_price]+1
jouts[3]<-length(llout_price)
```

#### 4.3.2 mileage

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

```
## [1] 3453 2036 3113 3170 3988 995 1006 3295 3385 4689
```

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



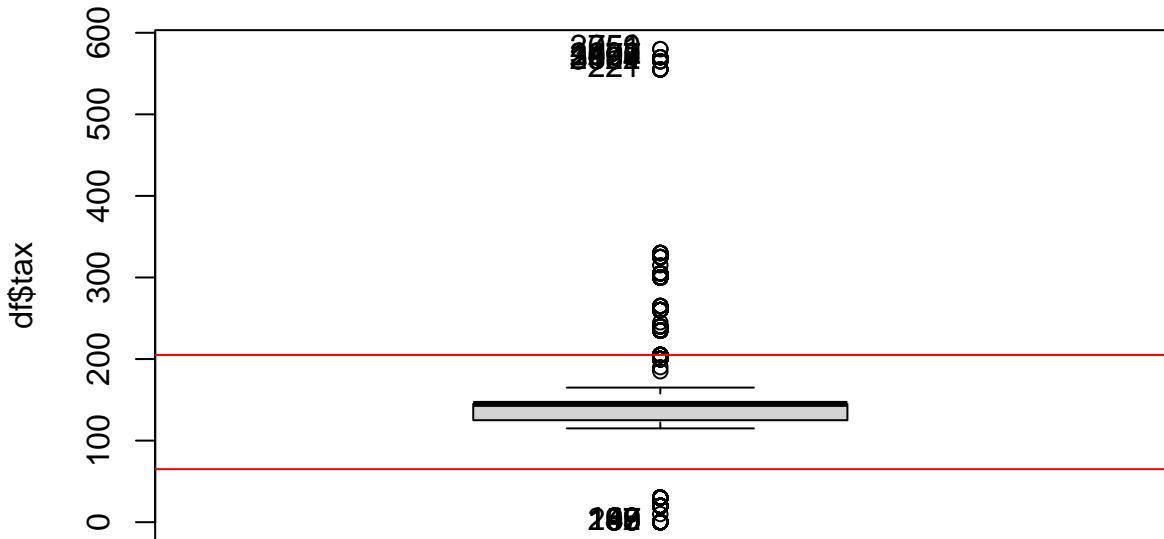
```
# Outliers:
llout_mileage<-which((df$mileage<var_out$souti)|(df$mileage>var_out$souts))#souts abline
iouts[llout_mileage]<-iouts[llout_mileage]+1
jouts[5]<-length(llout_mileage)
df[llout_mileage,"mileage"]<-NA #llout
```

#### 4.3.3 tax

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

```
## [1] 7 33 45 47 150 169 182 197 198 209 759 2051 1008 1077 2207
## [16] 3450 2094 3184 3322 221
```

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



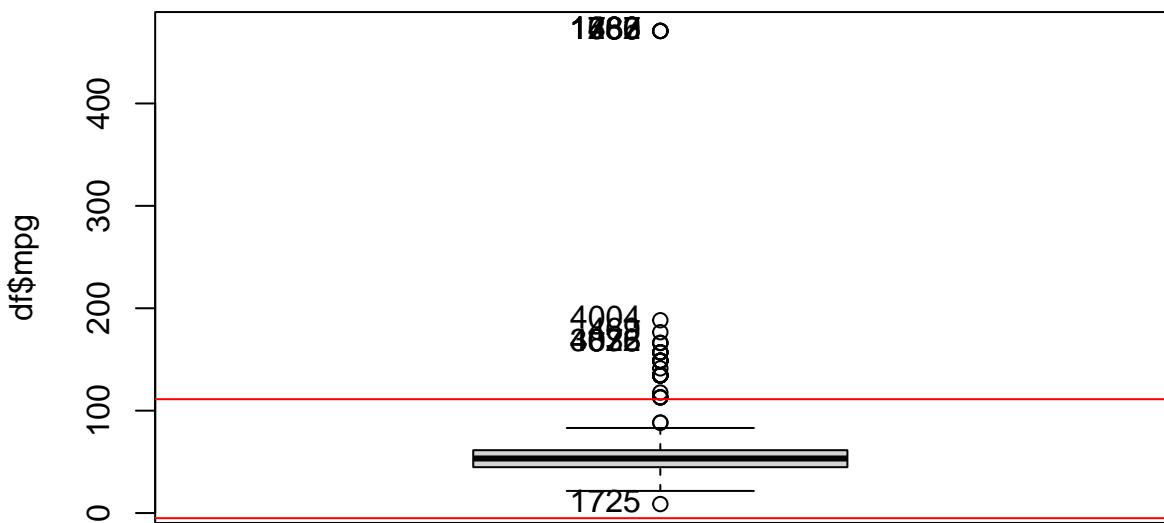
```
# Outliers:
llout_tax<-which((df$tax<var_out$souti)|(df$tax>var_out$souts))#souts abline
iouts[llout_tax]<-iouts[llout_tax]+1
jouts[7]<-length(llout_tax)
df[llout_tax,"tax"]<-NA #llout
```

#### 4.3.4 mpg

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

```
## [1] 1725 1280 1487 1636 1762 1763 4004 469 3832 4026 4073
```

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



```
# Outliers:
llout_mpg<-which((df$mpg<var_out$souti)|(df$mpg>var_out$souts))#souts abline
iouts[llout_mpg]<-iouts[llout_mpg]+1
jouts[8]<-length(llout_mpg)
df[llout_mpg, "mpg"]<-NA #llout
```

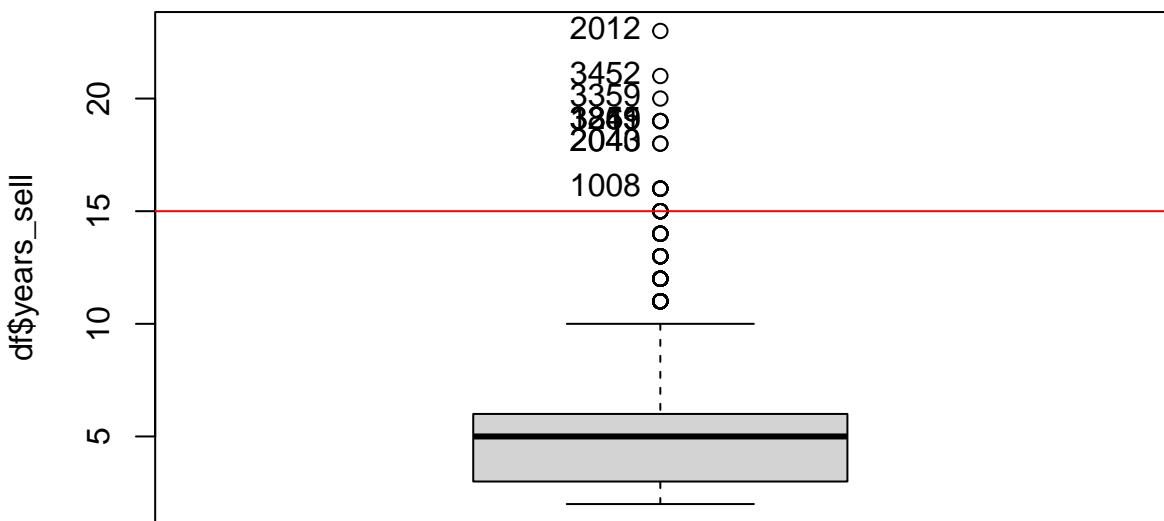
#### 4.3.5 years\_sell

We have decided not to assign NA to years\_sell because it is a special variable with a low number of univariate outliers.

```
Boxplot(df$years_sell)

## [1] 2012 3452 3359 1819 3249 3355 3361 2040 2043 1008

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



```
# Outliers:
llout_years_sell<-which((df$years_sell<var_out$souti)|(df$years_sell>var_out$souts))#souts abline
iouts[llout_years_sell]<-iouts[llout_years_sell]+1
jouts[11]<-length(llout_years_sell)
#df[llout_years_sell,"years_sell"]<-NA #llout
```

## 4.4 Number of errors, missings and outliers for individual and variable

### 4.4.1 Number of missing values of each variable

```
jmis
```

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

### 4.4.2 Number of errors per each variable

```
jerrs
```

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

### 4.4.3 Number of outliers per each variable

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

```
## [1] "tax  : 1292"
## [1] "mpg  : 48"
## [1] "price : 42"
## [1] "mileage : 20"
## [1] "years_sell : 15"
## [1] "model : 0"
## [1] "year : 0"
## [1] "transmission : 0"
## [1] "fuelType : 0"
## [1] "engineSize : 0"
## [1] "manufacturer : 0"
```

#### 4.4.4 Number of missing values individual

```
sum(imis)
```

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

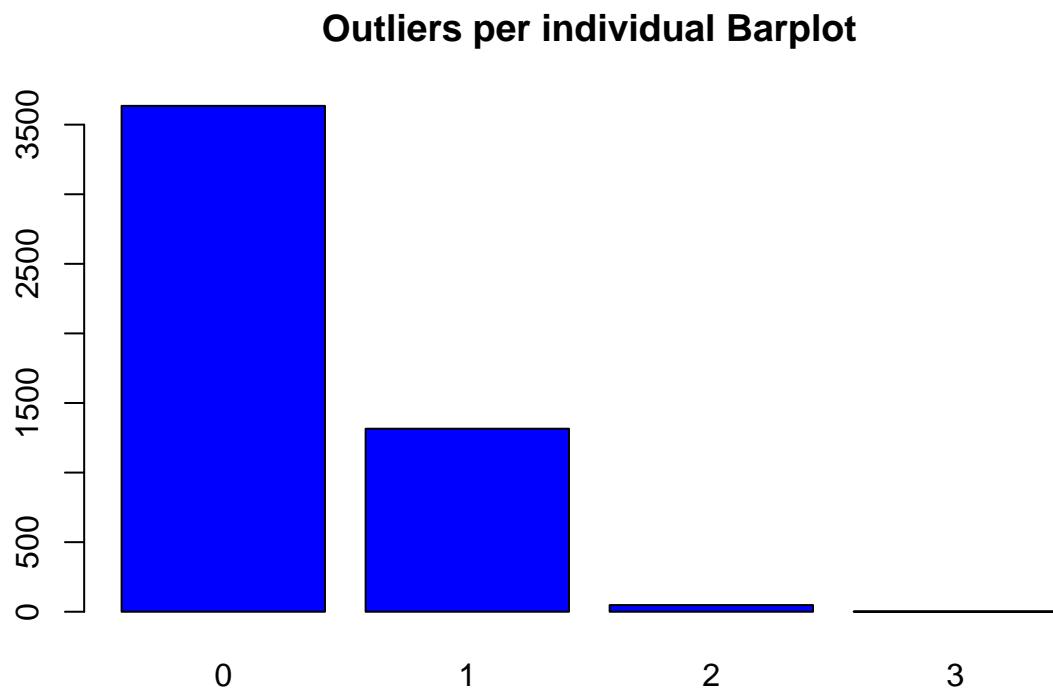
#### 4.4.5 Number of errors individual

```
sum(ierrs)
```

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

#### 4.4.6 Number of outliers individual

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



#### 4.4.7 New variable adding the total number missing values, outliers and errors

We have created a new variable to know the total missing values, outliers and errors per individual.

```
df$totalMOE<- imis+ioouts+ierrs
```

## 4.5 Imputation

### 4.5.1 Imputation of numeric variables

We do imputation in order to give a value to the missings (NA's). The (regularized) iterative PCA algorithm first consists imputing missing values with initial values such as the mean of the variable.

```
library(missMDA)

names(df)

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

vars_con<-names(df)[c(5,7,8)]
vars_dis<-names(df)[c(1:2, 4, 6, 9, 10)]
vars_res<-names(df)[c(3)]

summary(df[,vars_con])

##      mileage          tax          mpg
##  Min.   : 4   Min.   :115.0   Min.   : 8.80
##  1st Qu.: 5987  1st Qu.:145.0   1st Qu.:44.10
##  Median : 16509  Median :145.0   Median :52.30
##  Mean   : 22876  Mean   :146.8   Mean   :52.71
##  3rd Qu.: 33533  3rd Qu.:145.0   3rd Qu.:61.40
##  Max.   :116000  Max.   :205.0   Max.   :88.30
##  NA's   :20      NA's   :1292    NA's   :48

res.impca<-imputePCA(df[,vars_con],ncp=2)
summary(res.impca)

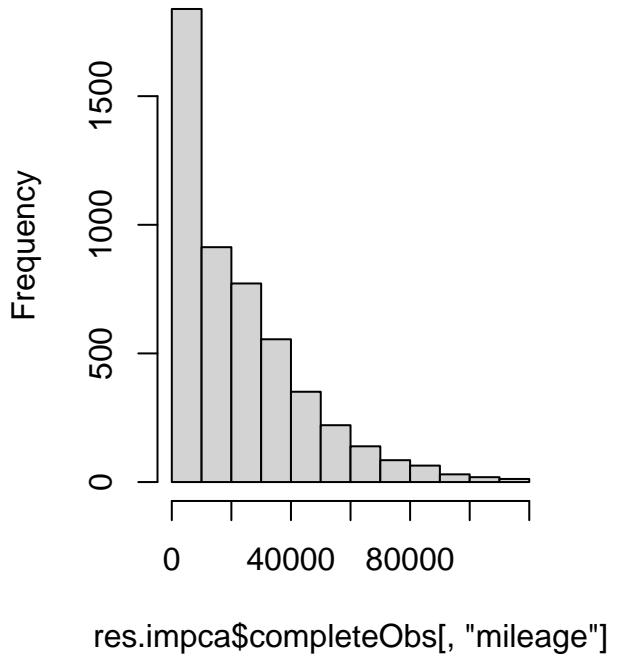
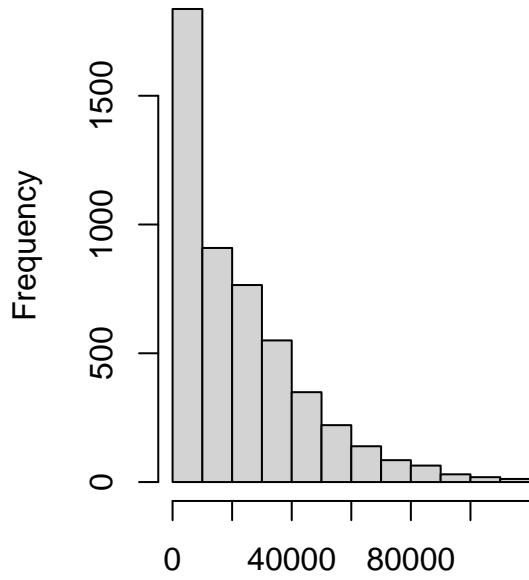
##      Length Class  Mode
## completeObs 15000  -none- numeric
## fittedX     15000  -none- numeric
```

We can notice a difference between the two plots, that means that the imputation has been/was correctly applied.

#### 4.5.1.1 mileage Plot comparison for mileage variable after imputation.

```
par(mfrow = c(1,2))
hist(df$mileage)
hist(res.impca$completeObs[, "mileage"])
```

## Histogram of df\$mileage      gram of res.impca\$completeObs[, "



```
quantile(df$mileage, seq(0,1,0.1), na.rm=T)
```

```
##      0%      10%      20%      30%      40%      50%      60%      70%
## 4.0 2000.0 4701.8 7370.7 11623.4 16508.5 22989.0 29712.1
## 80% 90% 100%
## 38260.8 52948.4 116000.0
```

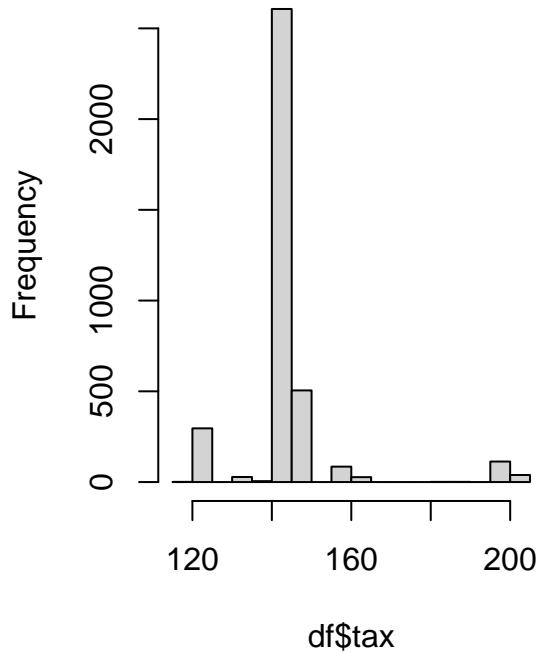
```
round(quantile(res.impca$completeObs[, "mileage"], seq(0,1,0.1), na.rm=T), dig=1)
```

```
##      0%      10%      20%      30%      40%      50%      60%      70%
## 4.0 2000.0 4714.4 7404.0 11648.6 16528.5 23000.0 29753.3
## 80% 90% 100%
## 38221.0 52770.7 116000.0
```

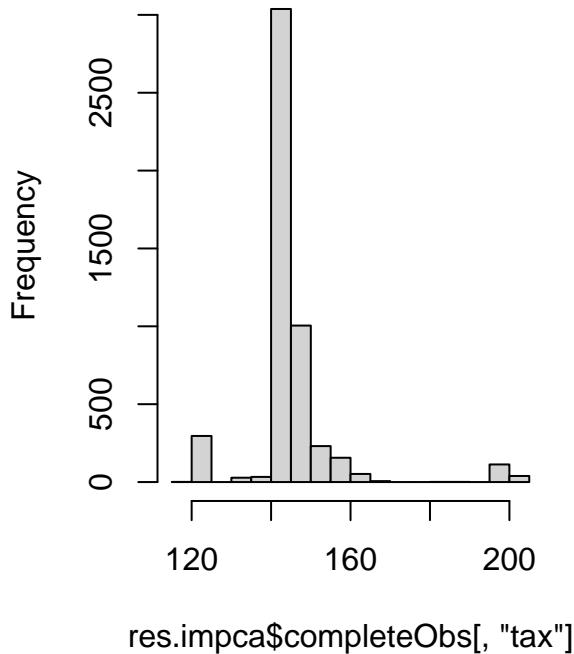
4.5.1.2 **tax** Plot comparison for tax variable after imputation.

```
par(mfrow=c(1,2))
hist(df$tax)
hist(res.impca$completeObs[, "tax"])
```

Histogram of df\$tax



Histogram of res.impca\$completeObs[, "tax"]



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

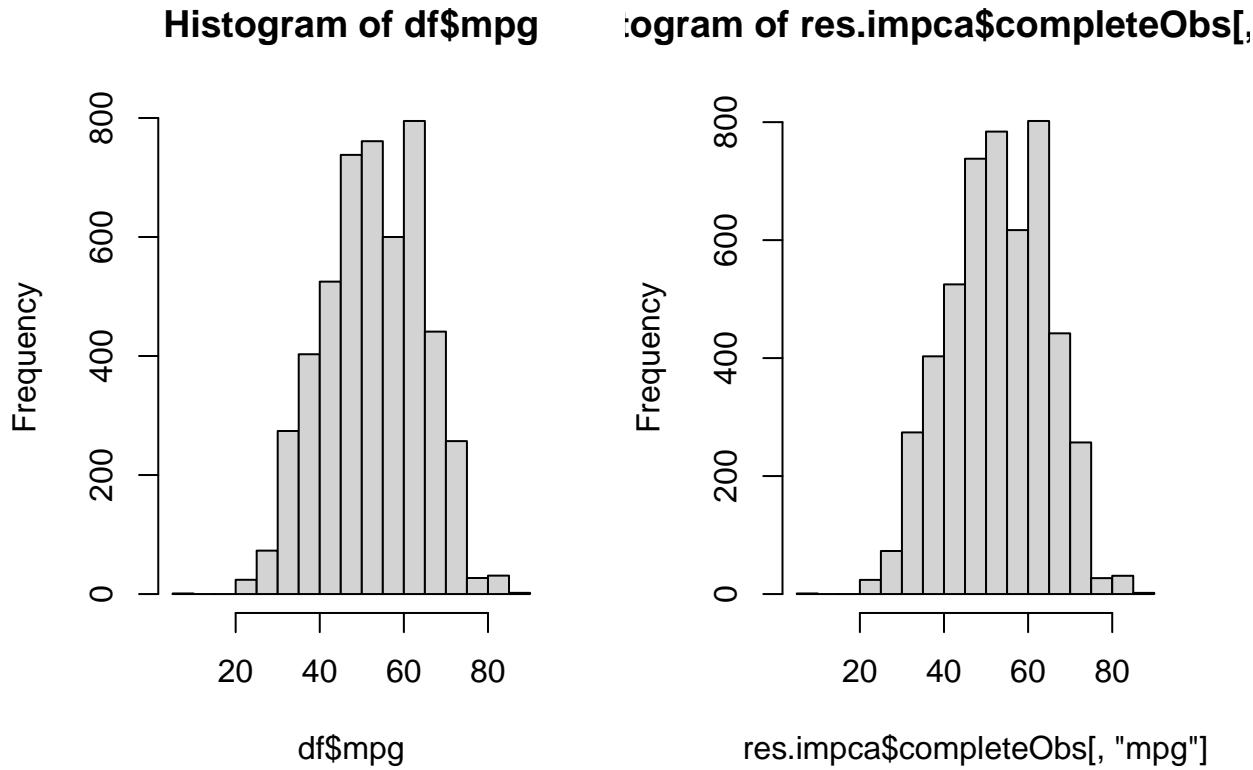
```
## 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
## 115 145 145 145 145 145 145 145 150 150 205
```

```
round(quantile(res.impca$completeObs[, "tax"], seq(0,1,0.1), na.rm=T), dig=1)
```

```
## 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
## 115.0 142.7 145.0 145.0 145.0 145.0 145.0 145.7 150.0 151.7 205.0
```

4.5.1.3 mpg Plot comparison for mpg variable after imputation.

```
par(mfrow = c(1,2))
hist(df$mpg)
hist(res.impca$completeObs[, "mpg"])
```



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

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%   100%
## 8.80 37.20 42.20 47.03 49.60 52.30 56.50 60.10 62.80 67.30 88.30
```

```
round(quantile(res.impca$completeObs[, "mpg"], seq(0, 1, 0.1), na.rm=T), dig=1)
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%   100%
## 8.8 37.2 42.2 47.1 49.6 53.3 56.5 60.1 62.8 67.3 88.3
```

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

#### 4.5.2 Imputation of factor variables

We do imputation in order to give a value to the missings (NA's). The (regularized) iterative MCA algorithm first consists in coding the categorical variables using the indicator matrix of dummy variables. Then, in the initialization step, missing values are imputed with initial values such as the proportion of the category for each category using the non-missing entries.

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

```
##          model          year      transmission
##  VW-Golf      : 488  f.Year-2019:1597  f.Trans-Manual  :1795
##  Mercedes-C Class: 367  f.Year-2017: 887  f.Trans-SemiAuto :1892
##  VW-Polo      : 348  f.Year-2016: 862  f.Trans-Automatic:1313
##  Mercedes-A Class: 270  f.Year-2018: 493
##  BMW-3 Series   : 243  f.Year-2015: 419
##  Audi-A3       : 199  f.Year-2020: 321
##  (Other)        :3085  (Other)      : 421
##          fuelType      engineSize      manufacturer
##  f.Fuel-Diesel:2842  f.EngSize-2   :2079   Audi      :1085
##  f.Fuel-Hybrid: 55   f.EngSize-3   : 562   BMW      :1049
```

```

##  f.Fuel-Other : 12  f.EngSize-1.5: 553  Mercedes:1351
##  f.Fuel-Petrol:2091  f.EngSize-1 : 407  VW      :1515
##                      f.EngSize-2.1: 407
##                      f.EngSize-1.6: 353
##                      (Other)      : 639

```

```

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

```

```

##          model          year          transmission
##  VW-Golf      : 488  f.Year-2019:1597  f.Trans-Manual   :1795
##  Mercedes-C Class: 367  f.Year-2017: 887  f.Trans-SemiAuto :1892
##  VW-Polo      : 348  f.Year-2016: 862  f.Trans-Automatic:1313
##  Mercedes-A Class: 270  f.Year-2018: 493
##  BMW-3 Series   : 243  f.Year-2015: 419
##  Audi-A3       : 199  f.Year-2020: 321
##  (Other)       :3085  (Other)      : 421
##          fuelType        engineSize   manufacturer
##  f.Fuel-Diesel:2842  f.EngSize-2 :2079  Audi      :1085
##  f.Fuel-Hybrid: 55   f.EngSize-3 : 562  BMW      :1049
##  f.Fuel-Other : 12   f.EngSize-1.5: 553  Mercedes:1351
##  f.Fuel-Petrol:2091  f.EngSize-1 : 407  VW       :1515
##                      f.EngSize-2.1: 407
##                      f.EngSize-1.6: 353
##                      (Other)      : 639

```

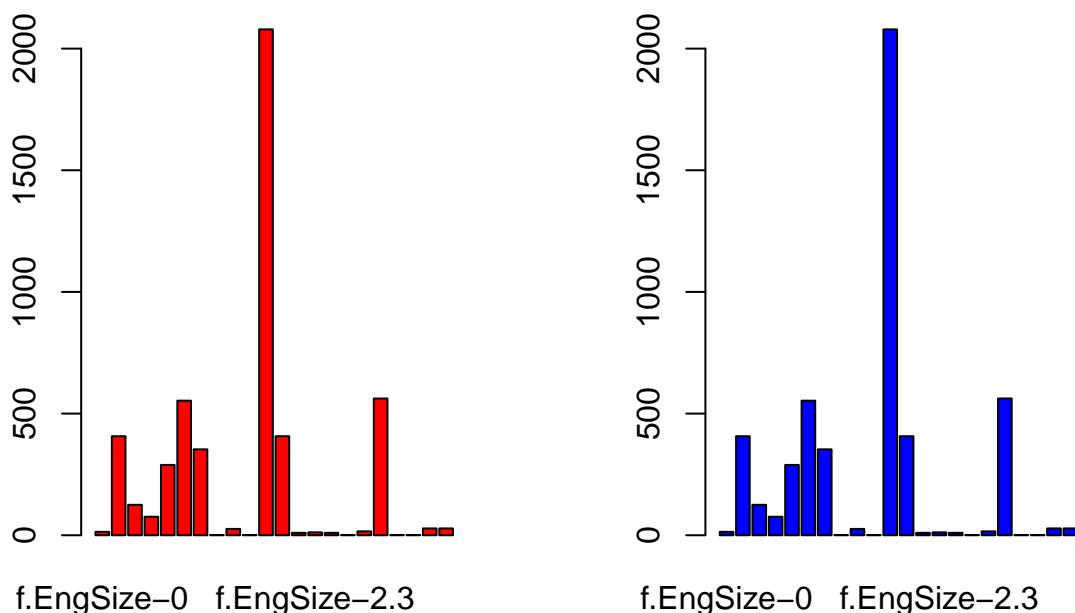
We can notice a difference between the two plots, that means that the imputation has been/was correctly applied.

#### 4.5.2.1 engineSize Plot comparison for engineSize variable.

```

par(mfrow=c(1,2))
barplot(table(df$engineSize),col="red")
barplot(table(res.immca$completeObs[, "engineSize"]),col="blue")

```



```
df[, vars_dis ]<-res.immca$complete0bs
sum(countNA(df)$mis_ind)==0
```

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

## 4.6 Correlation of numeric variables with MOE

```
cor(df[,c(3,5,7,8,11)], df$totalMOE)
```

```
## [,1]
## price      -0.29796314
## mileage     0.45404451
## tax         0.02667251
## mpg         0.32802523
## years_sell  0.52600573
```

As we can see years\_sell is one of the variables with most correlation with total\_MOE. This means that as the time of a car being sold grows more tendency to have errors, outliers or/and missing increase.

## 4.7 Discretization

We do discretization in order to make it easier to understand the numeric variables.

### 4.7.1 price variable

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

```
##      0%      25%      50%      75%     100%
## 650.00 13995.00 19498.00 26039.25 109990.00
```

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

```
##      0%      10%      20%      30%      40%      50%      60%      70%
## 650.0 10318.0 12890.0 15145.5 17399.6 19498.0 21989.4 24904.5
##      80%      90%     100%
## 28000.0 33950.2 109990.0
```

```
df$aux_price<-factor(cut(df$price/1000,breaks=c(quantile(df$price,seq(0,1,0.25),na.rm=TRUE))/1000,inclu
```

```
## [0.65,14] (14,19.5] (19.5,26] (26,110]
##      1259      1246      1245      1250
```

```
tapply(df$price,df$aux_price,median)
```

```
## [0.65,14] (14,19.5] (19.5,26] (26,110]
## 10995.0 16950.0 22646.0 31986.5
```

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

```
##
## f.price-[0.65,14] f.price-(14,19.5] f.price-(19.5,26] f.price-(26,110]
##          1259             1246             1245             1250
##          <NA>
##          0
```

#### 4.7.2 mileage variable

```
df$aux_mileage<-factor(cut(df$mileage, breaks=c(quantile(df$mileage, seq(0,1,0.25), na.rm=TRUE)), include.lowest = T))
summary(df$aux_mileage)

## [4,6e+03] (6e+03,1.65e+04] (1.65e+04,3.35e+04] (3.35e+04,1.16e+05]
## 1252 1248 1250 1250

tapply(df$mileage,df$aux_mileage,median)

## [4,6e+03] (6e+03,1.65e+04] (1.65e+04,3.35e+04] (3.35e+04,1.16e+05]
## 2753.5 10396.0 24443.5 48095.0

levels(df$aux_mileage)<-paste("f.mileage-",levels(df$aux_mileage),sep="")
table(df$aux_mileage,useNA="always")

## 
## f.mileage-[4,6e+03] f.mileage-(6e+03,1.65e+04]
## 1252 1248
## f.mileage-(1.65e+04,3.35e+04] f.mileage-(3.35e+04,1.16e+05]
## 1250 1250
## <NA>
## 0
```

#### 4.7.3 tax variable

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

## 0% 25% 50% 75% 100%
## 115.0000 145.0000 145.0000 148.0945 205.0000

df$aux_tax<-factor(cut(df$tax, breaks=c(0, 125, 145, 580), include.lowest = T ))
summary(df$aux_tax)

## [0,125] (125,145] (145,580]
## 297 3099 1604

tapply(df$tax,df$aux_tax,median)

## [0,125] (125,145] (145,580]
## 125 145 150

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

## 
## f.tax-[0,125] f.tax-(125,145] f.tax-(145,580] <NA>
## 297 3099 1604 0
```

#### 4.7.4 mpg variable

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

## 0% 25% 50% 75% 100%
## 8.8 44.8 53.3 61.4 88.3
```

```

df$aux_mpg<-factor(cut(df$mpg, breaks=c(quantile(df$mpg,seq(0,1,0.25),na.rm=TRUE)),include.lowest = T ))
summary(df$aux_mpg)

##  [8.8,44.8] (44.8,53.3] (53.3,61.4] (61.4,88.3]
##      1300      1378      1188      1134

tapply(df$mpg,df$aux_mpg,median)

##  [8.8,44.8] (44.8,53.3] (53.3,61.4] (61.4,88.3]
##      39.2      49.6      57.7      67.3

levels(df$aux_mpg)<-paste("f.mpg-",levels(df$aux_mpg),sep="")
table(df$aux_mpg,useNA="always")

## 
##  f.mpg-[8.8,44.8] f.mpg-(44.8,53.3] f.mpg-(53.3,61.4] f.mpg-(61.4,88.3]
##      1300          1378          1188          1134
##      <NA>
##          0

```

#### 4.7.5 years\_sell variable

```

df$aux_years_sell<-factor(cut(df$years_sell, breaks=c(quantile(df$years_sell,seq(0,1,0.25),na.rm=TRUE)),include.lowest = T ))
summary(df$aux_years_sell)

##  [2,3]  (3,5]  (5,6]  (6,23]
##  1918    1380    862    840

tapply(df$years_sell,df$aux_years_sell,median)

##  [2,3]  (3,5]  (5,6]  (6,23]
##      3      5      6      8

levels(df$aux_years_sell)<-paste("f.years_sell-",levels(df$aux_years_sell),sep="")
table(df$aux_years_sell,useNA="always")

## 
##  f.years_sell-[2,3]  f.years_sell-(3,5]  f.years_sell-(5,6]  f.years_sell-(6,23]
##      1918          1380          862          840
##      <NA>
##          0

```

## 4.8 Definition of binary outcome: Audi

Create binary target, define lists of numeric and qualitative variables and save your raw base database

We have created a binary target to know if the car is Audi or not. We will use this variable later to do profiling with it.

```

# Binary Target: Audi?
df$Audi<-ifelse(df$manufacturer == "Audi",1,0)
df$Audi<-factor(df$Audi,labels=paste("Audi",c("No","Yes")))
summary(df$Audi)

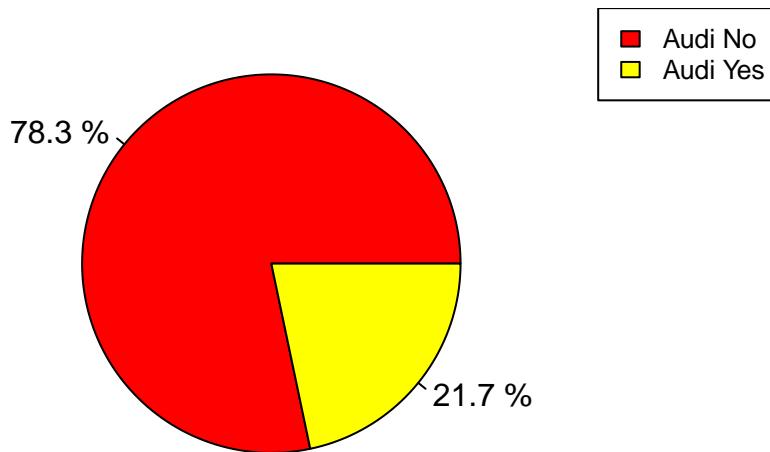
##  Audi No Audi Yes
##      3915      1085

```

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

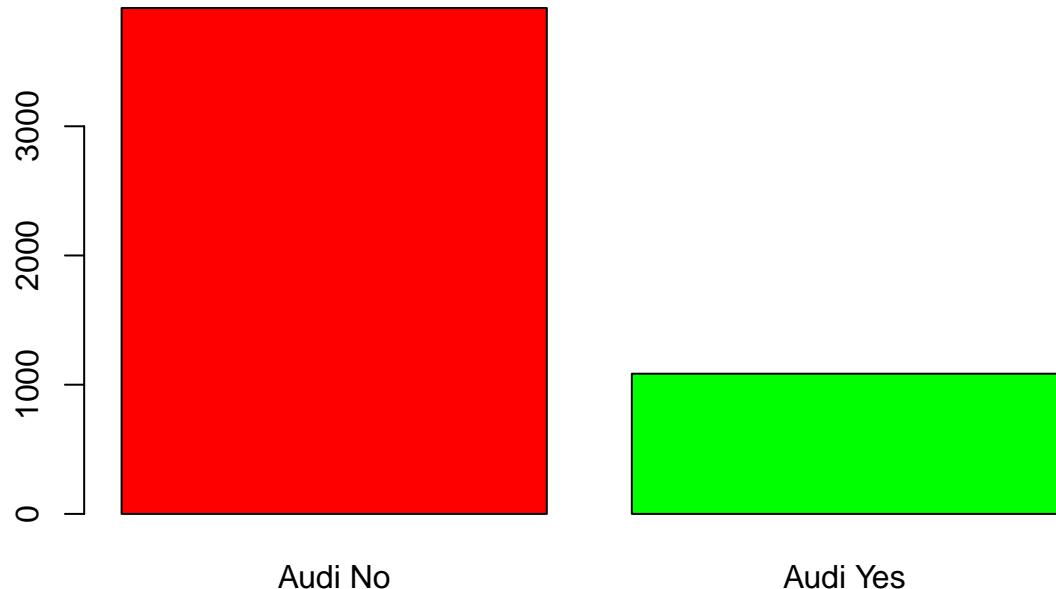
##
##  Audi No Audi Yes
##    78.3      21.7

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



```
# Bar Chart
barplot(table(df$Audi),main="Barplot Binary Outcome - Factor",col=c("red","green"))
```

## Barplot Binary Outcome – Factor



### 4.9 Multivariate outliers

We have executed Moutlier function in order to find the multivariate outliers. Then, we have created a new variable (df\$mout) in order to distinguish/differentiate individuals that are multivariate outliers (they have a robust distance greater than the cutoff distance) and individuals that not.

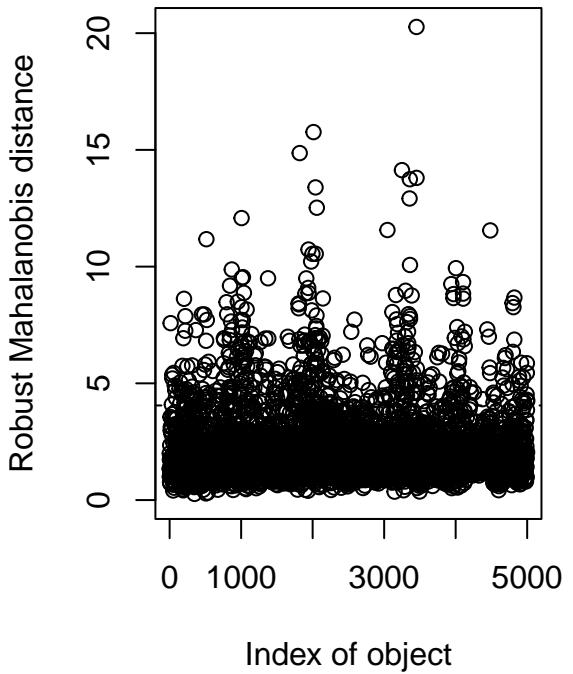
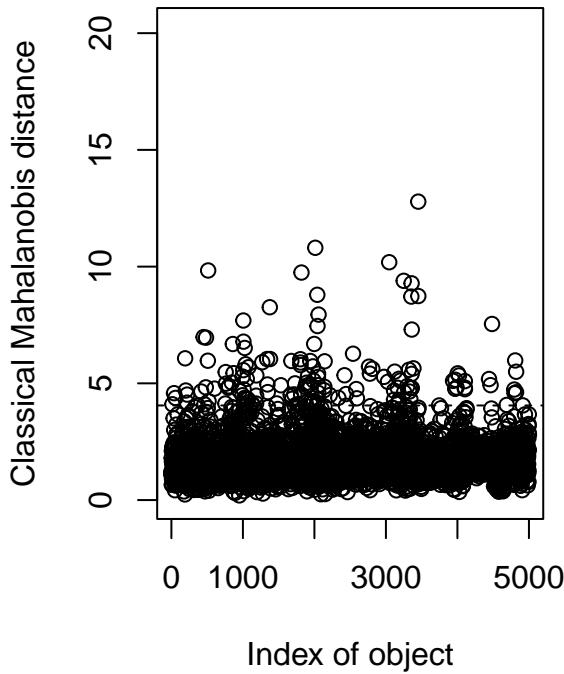
```
library(chemometrics)

## Loading required package: rpart

summary(df[,c(3, 5, 8, 11)])

##      price          mileage          mpg          years_sell
##  Min.   : 650   Min.   : 4   Min.   : 8.80   Min.   : 2.000
##  1st Qu.: 13995  1st Qu.: 5999  1st Qu.:44.80  1st Qu.: 3.000
##  Median : 19498  Median : 16529  Median :53.30  Median : 5.000
##  Mean   : 21470  Mean   : 22888  Mean   :52.74  Mean   : 4.787
##  3rd Qu.: 26039  3rd Qu.: 33516  3rd Qu.:61.40  3rd Qu.: 6.000
##  Max.   :109990  Max.   :116000  Max.   :88.30  Max.   :23.000

mout<-Moutlier(df[,c(3, 5, 8, 11)], quantile = 0.9975, plot = TRUE)
```



```

length(which(mout$rd>mout$cutoff))

## [1] 463

11<-which(mout$rd>mout$cutoff)
Boxplot(mout$rd)

## [1] 3452 2012 1819 3249 3453 3359 2040 3355 2058 1006

df[11[1:3],c(3,5,8,11)]

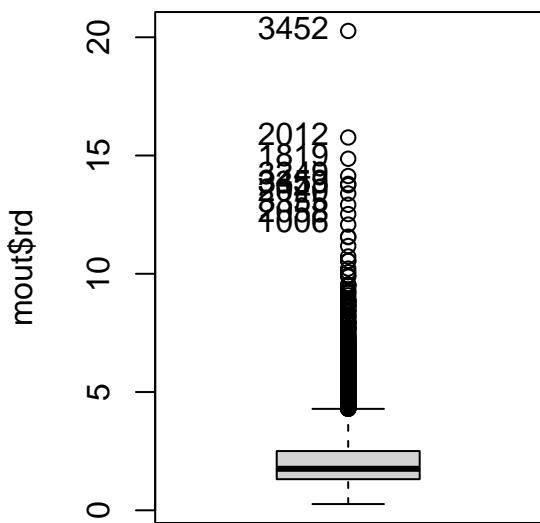
##      price mileage  mpg years_sell
## 14 16200     89334 62.8          6
## 31 56985      1510 33.2          3
## 37 63985     8450 32.8          4

df$mout <- 0
df$mout[ 11 ]<-1
df$mout <- factor( df$mout, labels=c( "NoMOut", "YesMOut" ))
table(df$mout)

## 
##  NoMOut  YesMOut 
##      4537      463

par(mfrow=c(1,1))

```

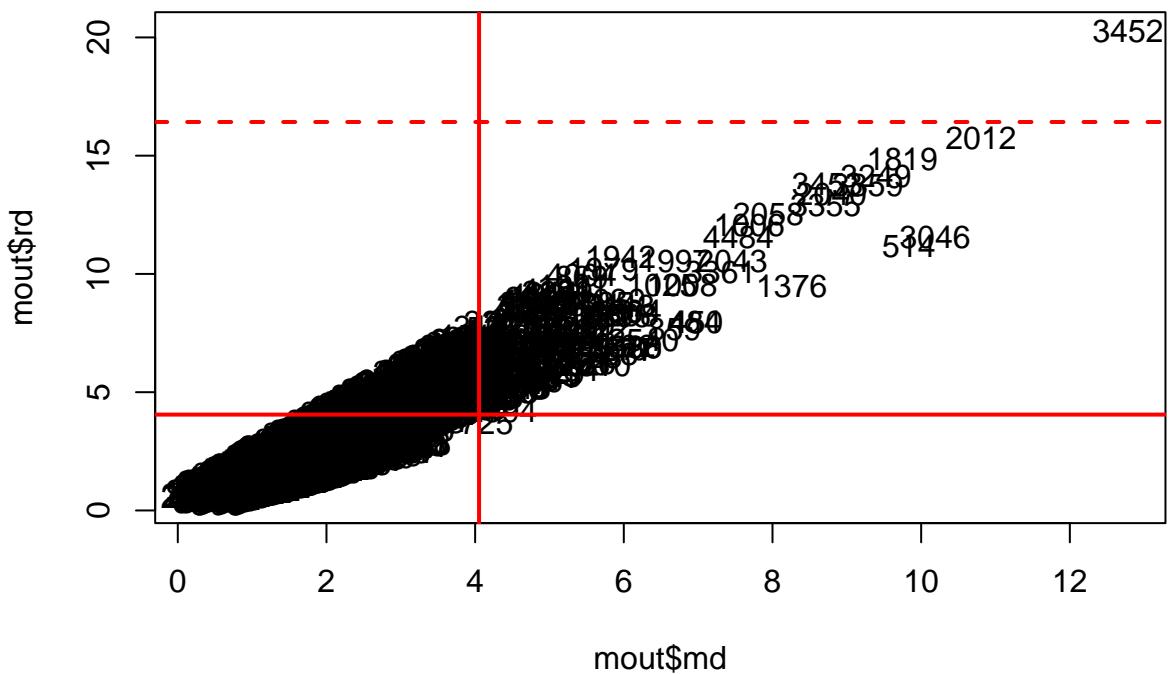


```

plot(mout$md,mout$rd, type="n")
text(mout$md,mout$rd,labels=rownames(df[,vars_con]))
abline(h=mout$cutoff, col="red", lwd=2)
abline(v=mout$cutoff, col="red", lwd=2)

abline(h=mout$cutoff^2,col="red",lwd=2,lty=2)
abline(v=mout$cutoff^2,col="red",lwd=2,lty=2)

```



## 4.10 Profiling

```

library(FactoMineR)
summary(df$price)

##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##      650    13995   19498  21470  26039 109990

res.condes<-condes(df,3)
res.condes$quanti # Global association to numeric variables

##          correlation      p.value
## totalMOE      -0.2979631 4.715577e-103
## mileage       -0.5131584  0.000000e+00
## years_sell   -0.5538620  0.000000e+00
## mpg          -0.5903613  0.000000e+00

res.condes$quali # Global association to factors

##          R2      p.value
## model      0.515041082 0.000000e+00
## year       0.353483215 0.000000e+00
## engineSize 0.413280059 0.000000e+00
## aux_price   0.731460406 0.000000e+00
## aux_mileage 0.293743806 0.000000e+00
## aux_mpg     0.300971015 0.000000e+00
## aux_years_sell 0.321950885 0.000000e+00
## transmission 0.230475968 5.322731e-285
## manufacturer 0.092467374 9.441791e-105
## aux_tax      0.089992603 4.717795e-103
## Audi         0.004668061 1.327290e-06
## fuelType     0.003656164 3.811516e-04

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

##                                     Estimate      p.value
## aux_years_sell=f.years_sell-[2,3] 9251.70250 0.000000e+00
## aux_mpg=f.mpg-[8.8,44.8]          9723.17533 0.000000e+00
## aux_price=f.price-(26,110)        14249.54562 0.000000e+00
## aux_mileage=f.mileage-[4,6e+03]   8087.75077 3.393062e-225
## year=f.Year-2019                 12374.45389 3.012776e-202
## engineSize=f.EngSize-3           12016.91222 5.157298e-195
## transmission=f.Trans-SemiAuto  4505.05997 1.119939e-131
## aux_tax=f.tax-(125,145)          4770.39249 1.736430e-103
## year=f.Year-2020                 16602.91086 2.019709e-77
## engineSize=f.EngSize-4           36182.92607 1.924377e-73
## year=f.Year-2016                 469.10740 8.174011e-60
## engineSize=f.EngSize-4.2 or more 27624.31893 3.490423e-43
## model=Audi-Q8                  43576.84832 3.318259e-41
## manufacturer=Mercedes           3039.49797 2.027962e-40
## model=Mercedes-GLE Class        10820.23923 7.120933e-37
## model=BMW-X7                   42413.20546 1.314652e-34
## model=Audi-Q7                  13708.16082 6.673097e-29
## model=Mercedes-GLC Class        3183.60042 8.847262e-28
## engineSize=f.EngSize-2.9         28576.84571 8.454111e-27
## transmission=f.Trans-Automatic 2531.21056 7.568797e-26
## aux_mileage=f.mileage-(6e+03,1.65e+04] 2756.38254 2.237589e-25
## year=f.Year-2017                 2745.46170 1.557611e-21
## model=Audi-Q5                  2914.98273 6.957440e-21
## model=Mercedes-S Class          17293.72332 3.925870e-20

```

```

## model=Mercedes-GLS Class
## model=BMW-8 Series
## model=BMW-X5
## model=BMW-M5
## model=Audi-R8
## model=BMW-M4
## model=Mercedes-SL CLASS
## model=VW-Touareg
## aux_years_sell=f.years_sell-(3,5]
## model=BMW-X6
## model=BMW-X3
## model=BMW-7 Series
## engineSize=f.EngSize-2
## Audi=Audi Yes
## manufacturer=Audi
## model=BMW-M2
## aux_price=f.price-(19.5,26]
## model=Audi-RS6
## model=VW-Caravelle
## model=Audi-A8
## model=Audi-RS5
## manufacturer=BMW
## model=BMW-X4
## model=Audi-RS4
## fuelType=f.Fuel-Diesel
## year=f.Year-2018
## model=BMW-i8
## model=Audi-S4
## model=BMW-X2
## model=VW-California
## model=BMW-M3
## model=Audi-RS3
## engineSize=f.EngSize-2.5
## model=VW-Tiguan Allspace
## model=VW-Jetta
## model=Mercedes-CLC Class
## model=VW-Touran
## model=Mercedes-X-CLASS
## aux_mpg=f.mpg-(44.8,53.3]
## model=VW-Arteon
## model=VW-Golf SV
## model=BMW-3 Series
## model=VW-Amarok
## model=Mercedes-SLK
## model=Mercedes-V Class
## model=Mercedes-CLS Class
## model=VW-Beetle
## engineSize=f.EngSize-1.5
## model=VW-CC
## model=VW-Scirocco
## model=Mercedes-A Class
## fuelType=f.Fuel-Petrol
## engineSize=f.EngSize-1.8
## engineSize=f.EngSize-2.1
## year=f.Year-2012
## model=Mercedes-C Class
## Audi=Audi No
## model=VW-Passat
## model=Audi-A3
## year=f.Year-2011
## model=Mercedes-E Class
## year=f.Year-2010
## model=Audi-A1
## model=BMW-1 Series
18533.56260 1.822382e-19
35992.74832 2.204029e-19
8463.71195 1.821812e-17
37683.09832 6.285862e-17
56033.34832 9.515733e-17
18969.53014 4.519729e-16
4815.03253 2.006258e-12
4073.84832 1.374518e-09
645.29335 1.510423e-08
15456.63403 2.045855e-08
364.01498 4.555479e-07
7855.09832 9.081694e-07
609.28213 1.057731e-06
898.85989 1.327290e-06
1120.63823 1.327290e-06
15534.34832 2.007301e-06
1231.36722 2.727027e-06
20839.68165 5.941713e-06
9357.72332 1.085753e-05
4406.53582 1.100877e-05
19173.34832 2.030061e-05
975.38554 2.200900e-05
3188.29276 2.841145e-05
23962.34832 4.062218e-05
81.36076 6.524733e-05
7588.80081 2.276380e-04
15037.68165 3.187854e-04
11958.09832 3.326456e-04
696.07559 3.885212e-04
29034.34832 7.553580e-04
4761.68165 5.636620e-03
3637.14832 2.176207e-02
6487.23321 4.836521e-02
-1746.22311 4.737339e-02
-19960.98502 4.630031e-02
-23029.15168 4.266624e-02
-11006.95937 4.185280e-02
-771.46986 3.982262e-02
-258.09211 3.326596e-02
-2896.28131 2.745189e-02
-12467.49168 2.132947e-02
-9145.48707 1.449274e-02
-512.27668 9.982846e-03
-20110.85168 9.206574e-03
-452.42946 5.836196e-03
-370.31835 5.286254e-03
-18955.36597 5.108585e-03
-1747.80132 7.195178e-04
-18679.10623 6.089175e-04
-15278.40168 4.165586e-04
-10172.40353 2.833106e-05
-1217.90283 2.275137e-05
-9426.92832 1.590542e-05
-2501.31936 1.564794e-05
-4887.18916 1.434448e-05
-4932.22934 2.721980e-06
-898.85989 1.327290e-06
-12997.69424 6.406054e-07
-11593.82254 4.796664e-08
-7439.76560 1.630019e-08
-2977.47390 1.281271e-08
-8897.47989 3.665675e-10
-14095.87796 4.173040e-13
-13141.11869 7.132624e-14

```

```

## aux_tax=f.tax-[0,125]          -3090.21034 2.050124e-18
## year=f.Year-2009 or before    -9958.24179 1.363765e-21
## model=VW-Golf                 -12379.53283 5.234941e-26
## engineSize=f.EngSize-1.4      -7037.38028 4.806127e-28
## year=f.Year-2013              -5024.10129 1.983361e-29
## model=VW-Up                   -20805.93577 1.276864e-31
## year=f.Year-2014              -3095.45499 8.013091e-33
## engineSize=f.EngSize-1.2      -11809.30679 7.807887e-34
## engineSize=f.EngSize-1.6      -7001.12118 4.564378e-34
## year=f.Year-2015              -478.50193 1.365347e-36
## aux_mileage=f.mileage-(1.65e+04,3.35e+04] -3644.94145 1.161832e-43
## aux_mpg=f.mpg-(53.3,61.4]      -3602.22515 5.344170e-46
## aux_years_sell=f.years_sell-(5,6] -3361.32631 8.174011e-60
## engineSize=f.EngSize-1         -8959.45696 1.684048e-65
## aux_price=f.price-(14,19.5]     -4657.16617 4.332775e-70
## aux_tax=f.tax-(145,580]        -1680.18215 5.578448e-71
## model=VW-Polo                 -17500.47352 1.353249e-73
## manufacturer=VW                -5135.52174 8.812113e-101
## aux_mpg=f.mpg-(61.4,88.3]      -5862.85807 4.311250e-109
## aux_years_sell=f.years_sell-(6,23] -6535.66954 4.910168e-149
## aux_mileage=f.mileage-(3.35e+04,1.16e+05] -7199.19185 9.366109e-175
## transmission=f.Trans-Manual   -7036.27053 1.829258e-279
## aux_price=f.price-[0.65,14]     -10823.74668 0.000000e+00

```

As we can see in the output `condes$quanti`, price variable is related with totalMOE, mpg, mileage and years\_sell variables because their p-values are less than 0.05. More concretely, against more price less totalMOE, less mileage, less years\_sell and less mpg.

As we can see in the output `condes$quali`, model variable has a high relation, i.e. certain models have high and/or low prices.

Regarding the output of `condes$category`, we can feature cars with model Audi-R8 which on average cost 56033.34832€ than baseline price, cars with model Audi-Q8 which on average cost 43576.84832€ more than baseline price and cars with model BMW-X7 which on average cost 42413.20546€ more than baseline price. On the other hand, we can feature cars with model Mercedes-CLC Class which on average cost 23029.15168€ less than baseline price, cars with model Mercedes-SLK which on average cost 20110.85168€ less than baseline price and finally cars with model VW-Up which on average cost 20805.93577€ less than baseline price.

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

```

## Audi No Audi Yes
##      3915      1085

```

```

res.catdes<-catdes(df,18)
res.catdes$quanti.var # Global association to numeric variables

```

```

##                  Eta2      P-value
## mpg      0.009703866 2.934540e-12
## price    0.004668061 1.327290e-06
## tax      0.004417699 2.552799e-06
## mileage 0.001260481 1.205183e-02

```

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

```

## $'Audi No'
##                  v.test Mean in category Overall mean sd in category Overall sd
## mpg      6.964885      53.33278      52.73678      11.34995      11.49277
## mileage -2.510208     22486.99051     22887.97553     21188.80333     21454.13386
## tax      -4.699370      146.58458      146.98985      11.00856      11.58247
## price    -4.830697     21080.13921     21470.24440     10650.88867     10845.87254
##                  p.value
## mpg      3.286717e-12

```

```

## mileage 1.206600e-02
## tax      2.609653e-06
## price    1.360558e-06
##
## $'Audi Yes'
##          v.test Mean in category Overall mean sd in category  Overall sd
## price     4.830697    22877.85899  21470.24440    11411.69143 10845.87254
## tax       4.699370     148.45220   146.98985     13.34934   11.58247
## mileage   2.510208    24334.84772  22887.97553    22325.59521 21454.13386
## mpg      -6.964885     50.58624    52.73678     11.74518   11.49277
##          p.value
## price    1.360558e-06
## tax      2.609653e-06
## mileage 1.206600e-02
## mpg      3.286717e-12

```

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

```

##          p.value df
## model      0.000000e+00 87
## manufacturer 0.000000e+00  3
## engineSize  1.938026e-76 21
## aux_mpg     1.676014e-16  3
## aux_price   2.927530e-06  3
## fuelType    8.418028e-06  3
## mout        3.693529e-03  1
## transmission 4.235600e-03  2
## aux_tax     1.645093e-02  2
## aux_mileage 2.230200e-02  3

```

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

```

## $'Audi No'
##          Cla/Mod     Mod/Cla Global
## manufacturer=VW      100.00000 38.69731801 30.30
## manufacturer=Mercedes 100.00000 34.50830140 27.02
## manufacturer=BMW      100.00000 26.79438059 20.98
## model=VW-Golf        100.00000 12.46487867 9.76
## engineSize=f.EngSize-2.1 100.00000 10.39591315 8.14
## model=Mercedes-C Class 100.00000 9.37420179 7.34
## model=VW-Polo        100.00000 8.88888889 6.96
## model=Mercedes-A Class 100.00000 6.89655172 5.40
## model=BMW-3 Series    100.00000 6.20689655 4.86
## model=BMW-1 Series    100.00000 5.03192848 3.94
## model=Mercedes-E Class 100.00000 4.59770115 3.60
## model=VW-Tiguan       100.00000 4.18901660 3.28
## model=BMW-2 Series    100.00000 3.52490421 2.76
## model=Mercedes-GLC Class 100.00000 3.03959132 2.38
## model=BMW-4 Series    100.00000 2.50319285 1.96
## model=VW-Passat       100.00000 2.40102171 1.88
## model=VW-Up            100.00000 2.24776501 1.76
## model=BMW-5 Series    100.00000 2.19667944 1.72
## model=Mercedes-GLA Class 100.00000 2.14559387 1.68
## engineSize=f.EngSize-1.3 100.00000 1.94125160 1.52
## model=BMW-X1          100.00000 1.89016603 1.48
## engineSize=f.EngSize-1.2 96.00000 3.06513410 2.50
## engineSize=f.EngSize-1.5 86.79928 12.26053640 11.06
## model=VW-T-Roc         100.00000 1.60919540 1.26
## model=Mercedes-B Class 100.00000 1.60919540 1.26
## aux_price=f.price-[0.65,14] 83.32010 26.79438059 25.18
## model=Mercedes-CL Class 100.00000 1.45593870 1.14
## model=Mercedes-GLE Class 100.00000 1.40485313 1.10

```

	p.value	v.test	
## aux_mpg=f.mpg-(61.4,88.3]	83.33333	24.13793103	22.68
## model=BMW-X3	100.00000	1.22605364	0.96
## aux_mpg=f.mpg-(53.3,61.4]	82.65993	25.08301405	23.76
## model=VW-Touran	100.00000	0.99616858	0.78
## model=Mercedes-SL CLASS	100.00000	0.97062580	0.76
## fuelType=f.Fuel-Hybrid	96.36364	1.35376756	1.10
## model=BMW-X5	100.00000	0.84291188	0.66
## model=VW-Touareg	100.00000	0.81736909	0.64
## engineSize=f.EngSize-1	84.52088	8.78671775	8.14
## model=VW-Arteon	100.00000	0.68965517	0.54
## model=VW-T-Cross	100.00000	0.63856960	0.50
## model=VW-Golf SV	100.00000	0.63856960	0.50
## model=VW-Scirocco	100.00000	0.61302682	0.48
## fuelType=f.Fuel-Diesel	79.80296	57.93103448	56.84
## mout=NoMOut	78.84064	91.36653895	90.74
## model=VW-Sharan	100.00000	0.56194125	0.44
## model=BMW-X2	100.00000	0.56194125	0.44
## aux_tax=f.tax-[0,125]	84.17508	6.38569604	5.94
## transmission=f.Trans-SemiAuto	80.23256	38.77394636	37.84
## model=Mercedes-V Class	100.00000	0.45977011	0.36
## model=Mercedes-CLS Class	100.00000	0.45977011	0.36
## model=BMW-X4	100.00000	0.45977011	0.36
## model=VW-Amarok	100.00000	0.40868455	0.32
## model=Mercedes-S Class	100.00000	0.40868455	0.32
## model=BMW-Z4	100.00000	0.38314176	0.30
## model=VW-Tiguan Allspace	100.00000	0.35759898	0.28
## model=Mercedes-GLS Class	100.00000	0.35759898	0.28
## model=Mercedes-GL Class	100.00000	0.35759898	0.28
## aux_mileage=f.mileage-(6e+03,1.65e+04]	80.28846	25.59386973	24.96
## model=Audi-RS4	0.00000	0.00000000	0.04
## model=Audi-R8	0.00000	0.00000000	0.04
## aux_years_sell=f.years_sell-(6,23]	75.11905	16.11749681	16.80
## year=f.Year-2015	73.50835	7.86717752	8.38
## model=Audi-RS6	0.00000	0.00000000	0.06
## model=Audi-RS5	0.00000	0.00000000	0.06
## mout=YesMOut	73.00216	8.63346105	9.26
## aux_mileage=f.mileage-(3.35e+04,1.16e+05]	75.36000	24.06130268	25.00
## model=Audi-S4	0.00000	0.00000000	0.08
## engineSize=f.EngSize-2.5	30.00000	0.07662835	0.20
## transmission=f.Trans-Manual	75.82173	34.76372925	35.90
## engineSize=f.EngSize-4	50.00000	0.35759898	0.56
## aux_price=f.price-(26,110]	74.96000	23.93358876	25.00
## model=Audi-SQ5	0.00000	0.00000000	0.10
## model=Audi-RS3	0.00000	0.00000000	0.10
## fuelType=f.Fuel-Petrol	75.65758	40.40868455	41.82
## model=Audi-Q8	0.00000	0.00000000	0.16
## engineSize=f.EngSize-2	74.36267	39.48914432	41.58
## model=Audi-A7	0.00000	0.00000000	0.28
## model=Audi-A8	0.00000	0.00000000	0.32
## aux_mpg=f.mpg-[8.8,44.8]	70.53846	23.42273308	26.00
## model=Audi-TT	0.00000	0.00000000	0.62
## model=Audi-Q7	0.00000	0.00000000	0.64
## engineSize=f.EngSize-1.4	47.40484	3.49936143	5.78
## model=Audi-A6	0.00000	0.00000000	1.42
## model=Audi-Q2	0.00000	0.00000000	1.60
## model=Audi-Q5	0.00000	0.00000000	1.86
## model=Audi-A5	0.00000	0.00000000	2.02
## model=Audi-A4	0.00000	0.00000000	2.72
## model=Audi-A1	0.00000	0.00000000	2.74
## model=Audi-Q3	0.00000	0.00000000	2.84
## model=Audi-A3	0.00000	0.00000000	3.98
## manufacturer=Audi	0.00000	0.00000000	21.70
## manufacturer=VW	6.200610e-198	30.015283	

## manufacturer=Mercedes	4.216696e-172	27.965963
## manufacturer=BMW	5.589380e-128	24.066644
## model=VW-Golf	1.159189e-55	15.716865
## engineSize=f.EngSize-2.1	4.367102e-46	14.251826
## model=Mercedes-C Class	1.985400e-41	13.482401
## model=VW-Polo	3.119709e-39	13.104093
## model=Mercedes-A Class	2.535101e-30	11.443480
## model=BMW-3 Series	2.812726e-27	10.818523
## model=BMW-1 Series	3.908271e-22	9.673488
## model=Mercedes-E Class	3.008936e-20	9.218639
## model=VW-Tiguan	1.765897e-18	8.771316
## model=BMW-2 Series	1.278514e-15	7.996649
## model=Mercedes-GLC Class	1.532980e-13	7.384265
## model=BMW-4 Series	2.968357e-11	6.648132
## model=VW-Passat	8.069916e-11	6.499291
## model=VW-Up	3.611143e-10	6.269971
## model=BMW-5 Series	5.947978e-10	6.191803
## model=Mercedes-GLA Class	9.794760e-10	6.112719
## engineSize=f.EngSize-1.3	7.186080e-09	5.786517
## model=BMW-X1	1.181999e-08	5.702302
## engineSize=f.EngSize-1.2	2.000762e-08	5.611935
## engineSize=f.EngSize-1.5	7.077720e-08	5.389187
## model=VW-T-Roc	1.817722e-07	5.217073
## model=Mercedes-B Class	1.817722e-07	5.217073
## aux_price=f.price-[0.65,14]	3.496523e-07	5.094506
## model=Mercedes-CL Class	8.047644e-07	4.934208
## model=Mercedes-GLE Class	1.320844e-06	4.836592
## aux_mpg=f.mpg-(61.4,88.3]	1.808258e-06	4.773751
## model=BMW-X3	7.468166e-06	4.479890
## aux_mpg=f.mpg-(53.3,61.4]	2.176387e-05	4.245981
## model=VW-Touran	6.898687e-05	3.979753
## model=Mercedes-SL CLASS	8.829322e-05	3.920697
## fuelType=f.Fuel-Hybrid	1.981835e-04	3.721321
## model=BMW-X5	3.029459e-04	3.612768
## model=VW-Touareg	3.875960e-04	3.548389
## engineSize=f.EngSize-1	1.055041e-03	3.275422
## model=VW-Arteon	1.327652e-03	3.209935
## model=VW-T-Cross	2.171678e-03	3.065691
## model=VW-Golf SV	2.171678e-03	3.065691
## model=VW-Scirocco	2.777248e-03	2.991374
## fuelType=f.Fuel-Diesel	3.165296e-03	2.951212
## mout=NoMOut	4.528525e-03	2.838788
## model=VW-Sharan	4.541297e-03	2.837889
## model=BMW-X2	4.541297e-03	2.837889
## aux_tax=f.tax-[0,125]	9.288054e-03	2.601265
## transmission=f.Trans-SemiAuto	9.456906e-03	2.595079
## model=Mercedes-V Class	1.213442e-02	2.508211
## model=Mercedes-CLS Class	1.213442e-02	2.508211
## model=BMW-X4	1.213442e-02	2.508211
## model=VW-Amarok	1.982863e-02	2.329575
## model=Mercedes-S Class	1.982863e-02	2.329575
## model=BMW-Z4	2.534506e-02	2.236102
## model=VW-Tiguan Allspace	3.239438e-02	2.139510
## model=Mercedes-GLS Class	3.239438e-02	2.139510
## model=Mercedes-GL Class	3.239438e-02	2.139510
## aux_mileage=f.mileage-(6e+03,1.65e+04]	4.802947e-02	1.977108
## model=Audi-RS4	4.705501e-02	-1.985805
## model=Audi-R8	4.705501e-02	-1.985805
## aux_years_sell=f.years_sell-(6,23]	1.539365e-02	-2.422983
## year=f.Year-2015	1.482733e-02	-2.436569
## model=Audi-RS6	1.019619e-02	-2.569104
## model=Audi-RS5	1.019619e-02	-2.569104
## mout=YesMOut	4.528525e-03	-2.838788
## aux_mileage=f.mileage-(3.35e+04,1.16e+05]	3.928202e-03	-2.883871

```

## model=Audi-S4          2.207781e-03 -3.060757
## engineSize=f.EngSize-2.5 1.574193e-03 -3.160647
## transmission=f.Trans-Manual 1.556181e-03 -3.163998
## engineSize=f.EngSize-4 1.073648e-03 -3.270481
## aux_price=f.price-(26,110] 1.069335e-03 -3.271619
## model=Audi-SQ5 4.777044e-04 -3.492956
## model=Audi-RS3 4.777044e-04 -3.492956
## fuelType=f.Fuel-Petrol 1.287964e-04 -3.828749
## model=Audi-Q8 4.818120e-06 -4.572555
## engineSize=f.EngSize-2 1.435083e-08 -5.669149
## model=Audi-A7 4.805857e-10 -6.225316
## model=Audi-A8 2.215780e-11 -6.691048
## aux_mpg=f.mpg-[8.8,44.8] 1.250134e-14 -7.710818
## model=Audi-TT 1.917911e-21 -9.509386
## model=Audi-Q7 4.068178e-22 -9.669384
## engineSize=f.EngSize-1.4 1.255775e-32 -11.895050
## model=Audi-A6 1.226198e-48 -14.656391
## model=Audi-Q2 7.866556e-55 -15.595047
## model=Audi-Q5 7.969683e-64 -16.866251
## model=Audi-A5 2.173682e-69 -17.607081
## model=Audi-A4 5.313784e-94 -20.567976
## model=Audi-A1 1.036756e-94 -20.647092
## model=Audi-Q3 2.894022e-98 -21.038835
## model=Audi-A3 1.877840e-139 -25.138712
## manufacturer=Audi 0.000000e+00 -Inf
##
## $'Audi Yes'
##
## manufacturer=Audi          Cla/Mod     Mod/Cla Global
## model=Audi-A3 100.000000 100.0000000 21.70
## model=Audi-Q3 100.000000 18.3410138 3.98
## model=Audi-A1 100.000000 13.0875576 2.84
## model=Audi-A4 100.000000 12.6267281 2.74
## model=Audi-A5 100.000000 12.5345622 2.72
## model=Audi-Q5 100.000000 9.3087558 2.02
## model=Audi-Q2 100.000000 8.5714286 1.86
## model=Audi-A6 100.000000 7.3732719 1.60
## engineSize=f.EngSize-1.4 52.595156 14.0092166 5.78
## model=Audi-Q7 100.000000 2.9493088 0.64
## model=Audi-TT 100.000000 2.8571429 0.62
## aux_mpg=f.mpg-[8.8,44.8] 29.461538 35.2995392 26.00
## model=Audi-A8 100.000000 1.4746544 0.32
## model=Audi-A7 100.000000 1.2903226 0.28
## engineSize=f.EngSize-2 25.637326 49.1244240 41.58
## model=Audi-Q8 100.000000 0.7373272 0.16
## fuelType=f.Fuel-Petrol 24.342420 46.9124424 41.82
## model=Audi-SQ5 100.000000 0.4608295 0.10
## model=Audi-RS3 100.000000 0.4608295 0.10
## aux_price=f.price-(26,110] 25.040000 28.8479263 25.00
## engineSize=f.EngSize-4 50.000000 1.2903226 0.56
## transmission=f.Trans-Manual 24.178273 40.0000000 35.90
## engineSize=f.EngSize-2.5 70.000000 0.6451613 0.20
## model=Audi-S4 100.000000 0.3686636 0.08
## aux_mileage=f.mileage-(3.35e+04,1.16e+05] 24.640000 28.3870968 25.00
## mout=YesMOut 26.997840 11.5207373 9.26
## model=Audi-RS6 100.000000 0.2764977 0.06
## model=Audi-RS5 100.000000 0.2764977 0.06
## year=f.Year-2015 26.491647 10.2304147 8.38
## aux_years_sell=f.years_sell-(6,23] 24.880952 19.2626728 16.80
## model=Audi-RS4 100.000000 0.1843318 0.04
## model=Audi-R8 100.000000 0.1843318 0.04
## aux_mileage=f.mileage-(6e+03,1.65e+04] 19.711538 22.6728111 24.96
## model=VW-Tiguan Allspace 0.000000 0.0000000 0.28
## model=Mercedes-GLS Class 0.000000 0.0000000 0.28

```

			p.value	v.test
## model=Mercedes-GL Class	0.000000	0.0000000	0.28	
## model=BMW-Z4	0.000000	0.0000000	0.30	
## model=VW-Amarok	0.000000	0.0000000	0.32	
## model=Mercedes-S Class	0.000000	0.0000000	0.32	
## model=Mercedes-V Class	0.000000	0.0000000	0.36	
## model=Mercedes-CLS Class	0.000000	0.0000000	0.36	
## model=BMW-X4	0.000000	0.0000000	0.36	
## transmission=f.Trans-SemiAuto	19.767442	34.4700461	37.84	
## aux_tax=f.tax-[0,125]	15.824916	4.3317972	5.94	
## model=VW-Sharan	0.000000	0.0000000	0.44	
## model=BMW-X2	0.000000	0.0000000	0.44	
## mout=NoMOut	21.159356	88.4792627	90.74	
## fuelType=f.Fuel-Diesel	20.197044	52.9032258	56.84	
## model=VW-Scirocco	0.000000	0.0000000	0.48	
## model=VW-T-Cross	0.000000	0.0000000	0.50	
## model=VW-Golf SV	0.000000	0.0000000	0.50	
## model=VW-Arteon	0.000000	0.0000000	0.54	
## engineSize=f.EngSize-1	15.479115	5.8064516	8.14	
## model=VW-Touareg	0.000000	0.0000000	0.64	
## model=BMW-X5	0.000000	0.0000000	0.66	
## fuelType=f.Fuel-Hybrid	3.636364	0.1843318	1.10	
## model=Mercedes-SL CLASS	0.000000	0.0000000	0.76	
## model=VW-Touran	0.000000	0.0000000	0.78	
## aux_mpg=f.mpg-(53.3,61.4]	17.340067	18.9861751	23.76	
## model=BMW-X3	0.000000	0.0000000	0.96	
## aux_mpg=f.mpg-(61.4,88.3]	16.666667	17.4193548	22.68	
## model=Mercedes-GLE Class	0.000000	0.0000000	1.10	
## model=Mercedes-CL Class	0.000000	0.0000000	1.14	
## aux_price=f.price-[0.65,14]	16.679905	19.3548387	25.18	
## model=VW-T-Roc	0.000000	0.0000000	1.26	
## model=Mercedes-B Class	0.000000	0.0000000	1.26	
## engineSize=f.EngSize-1.5	13.200723	6.7281106	11.06	
## engineSize=f.EngSize-1.2	4.000000	0.4608295	2.50	
## model=BMW-X1	0.000000	0.0000000	1.48	
## engineSize=f.EngSize-1.3	0.000000	0.0000000	1.52	
## model=Mercedes-GLA Class	0.000000	0.0000000	1.68	
## model=BMW-5 Series	0.000000	0.0000000	1.72	
## model=VW-Up	0.000000	0.0000000	1.76	
## model=VW-Passat	0.000000	0.0000000	1.88	
## model=BMW-4 Series	0.000000	0.0000000	1.96	
## model=Mercedes-GLC Class	0.000000	0.0000000	2.38	
## model=BMW-2 Series	0.000000	0.0000000	2.76	
## model=VW-Tiguan	0.000000	0.0000000	3.28	
## model=Mercedes-E Class	0.000000	0.0000000	3.60	
## model=BMW-1 Series	0.000000	0.0000000	3.94	
## model=BMW-3 Series	0.000000	0.0000000	4.86	
## model=Mercedes-A Class	0.000000	0.0000000	5.40	
## model=VW-Polo	0.000000	0.0000000	6.96	
## model=Mercedes-C Class	0.000000	0.0000000	7.34	
## engineSize=f.EngSize-2.1	0.000000	0.0000000	8.14	
## model=VW-Golf	0.000000	0.0000000	9.76	
## manufacturer=BMW	0.000000	0.0000000	20.98	
## manufacturer=Mercedes	0.000000	0.0000000	27.02	
## manufacturer=VW	0.000000	0.0000000	30.30	
## manufacturer=Audi	0.000000e+00	Inf		
## model=Audi-A3	1.877840e-139	25.138712		
## model=Audi-Q3	2.894022e-98	21.038835		
## model=Audi-A1	1.036756e-94	20.647092		
## model=Audi-A4	5.313784e-94	20.567976		
## model=Audi-A5	2.173682e-69	17.607081		
## model=Audi-Q5	7.969683e-64	16.866251		
## model=Audi-Q2	7.866556e-55	15.595047		
## model=Audi-A6	1.226198e-48	14.656391		

```

## engineSize=f.EngSize-1.4 1.255775e-32 11.895050
## model=Audi-Q7 4.068178e-22 9.669384
## model=Audi-TT 1.917911e-21 9.509386
## aux_mpg=f.mpg-[8.8,44.8] 1.250134e-14 7.710818
## model=Audi-A8 2.215780e-11 6.691048
## model=Audi-A7 4.805857e-10 6.225316
## engineSize=f.EngSize-2 1.435083e-08 5.669149
## model=Audi-Q8 4.818120e-06 4.572555
## fuelType=f.Fuel-Petrol 1.287964e-04 3.828749
## model=Audi-SQ5 4.777044e-04 3.492956
## model=Audi-RS3 4.777044e-04 3.492956
## aux_price=f.price-(26,110] 1.069335e-03 3.271619
## engineSize=f.EngSize-4 1.073648e-03 3.270481
## transmission=f.Trans-Manual 1.556181e-03 3.163998
## engineSize=f.EngSize-2.5 1.574193e-03 3.160647
## model=Audi-S4 2.207781e-03 3.060757
## aux_mileage=f.mileage-(3.35e+04,1.16e+05] 3.928202e-03 2.883871
## mout=YesMOut 4.528525e-03 2.838788
## model=Audi-RS6 1.019619e-02 2.569104
## model=Audi-RS5 1.019619e-02 2.569104
## year=f.Year-2015 1.482733e-02 2.436569
## aux_years_sell=f.years_sell-(6,23] 1.539365e-02 2.422983
## model=Audi-RS4 4.705501e-02 1.985805
## model=Audi-R8 4.705501e-02 1.985805
## aux_mileage=f.mileage-(6e+03,1.65e+04] 4.802947e-02 -1.977108
## model=VW-Tiguan Allspace 3.239438e-02 -2.139510
## model=Mercedes-GLS Class 3.239438e-02 -2.139510
## model=Mercedes-GL Class 3.239438e-02 -2.139510
## model=BMW-Z4 2.534506e-02 -2.236102
## model=VW-Amarok 1.982863e-02 -2.329575
## model=Mercedes-S Class 1.982863e-02 -2.329575
## model=Mercedes-V Class 1.213442e-02 -2.508211
## model=Mercedes-CLS Class 1.213442e-02 -2.508211
## model=BMW-X4 1.213442e-02 -2.508211
## transmission=f.Trans-SemiAuto 9.456906e-03 -2.595079
## aux_tax=f.tax-[0,125] 9.288054e-03 -2.601265
## model=VW-Sharan 4.541297e-03 -2.837889
## model=BMW-X2 4.541297e-03 -2.837889
## mout=NoMOut 4.528525e-03 -2.838788
## fuelType=f.Fuel-Diesel 3.165296e-03 -2.951212
## model=VW-Scirocco 2.777248e-03 -2.991374
## model=VW-T-Cross 2.171678e-03 -3.065691
## model=VW-Golf SV 2.171678e-03 -3.065691
## model=VW-Arteon 1.327652e-03 -3.209935
## engineSize=f.EngSize-1 1.055041e-03 -3.275422
## model=VW-Touareg 3.875960e-04 -3.548389
## model=BMW-X5 3.029459e-04 -3.612768
## fuelType=f.Fuel-Hybrid 1.981835e-04 -3.721321
## model=Mercedes-SL CLASS 8.829322e-05 -3.920697
## model=VW-Touran 6.898687e-05 -3.979753
## aux_mpg=f.mpg-(53.3,61.4] 2.176387e-05 -4.245981
## model=BMW-X3 7.468166e-06 -4.479890
## aux_mpg=f.mpg-(61.4,88.3] 1.808258e-06 -4.773751
## model=Mercedes-GLE Class 1.320844e-06 -4.836592
## model=Mercedes-CL Class 8.047644e-07 -4.934208
## aux_price=f.price-[0.65,14] 3.496523e-07 -5.094506
## model=VW-T-Roc 1.817722e-07 -5.217073
## model=Mercedes-B Class 1.817722e-07 -5.217073
## engineSize=f.EngSize-1.5 7.077720e-08 -5.389187
## engineSize=f.EngSize-1.2 2.000762e-08 -5.611935
## model=BMW-X1 1.181999e-08 -5.702302
## engineSize=f.EngSize-1.3 7.186080e-09 -5.786517
## model=Mercedes-GLA Class 9.794760e-10 -6.112719
## model=BMW-5 Series 5.947978e-10 -6.191803

```

```

## model=VW-Up          3.611143e-10  -6.269971
## model=VW-Passat      8.069916e-11  -6.499291
## model=BMW-4 Series   2.968357e-11  -6.648132
## model=Mercedes-GLC Class 1.532980e-13  -7.384265
## model=BMW-2 Series    1.278514e-15  -7.996649
## model=VW-Tiguan       1.765897e-18  -8.771316
## model=Mercedes-E Class 3.008936e-20  -9.218639
## model=BMW-1 Series    3.908271e-22  -9.673488
## model=BMW-3 Series    2.812726e-27  -10.818523
## model=Mercedes-A Class 2.535101e-30  -11.443480
## model=VW-Polo         3.119709e-39  -13.104093
## model=Mercedes-C Class 1.985400e-41  -13.482401
## engineSize=f.EngSize-2.1 4.367102e-46  -14.251826
## model=VW-Golf          1.159189e-55  -15.716865
## manufacturer=BMW       5.589380e-128 -24.066644
## manufacturer=Mercedes  4.216696e-172 -27.965963
## manufacturer=VW         6.200610e-198 -30.015283

```

As we can see in the output catdes\$quanti.var, we can feature that  $\mu_{mpgAudiNo}$ ,  $\mu_{mpgAudiYes}$  and  $\mu_{mpg}$  are not the same because the p-value is less than 0.05.

As we can see in the output catdes\$quanti, we can feature that Audi cars on average consume more, because on average they have less mpg.

As we can see in the output catdes\$test.chi2, we can feature that Audi variable and model variable are not independent because the p-value is less than 0.05. Also, Audi variable and engineSize variable are not independent for the same reason.

As we can see in the output catdes\$category, we can feature that VW, Mercedes and BMW cars are overrepresented in Audi No sample. Also, we can feature that engineSize 1.4, 2, 4 and 2.5 are overrepresented in Audi Yes sample and engineSize 1, 1.5, 1.2, 1.3, 2.1 are underrepresented in Audi Yes sample.

# USED CAR PRICES CASE STUDY

## Deliverable 2: PCA, CA and Clustering

Miquel Parra i Xavier Alaman

May 1, 2022

## Contents

<b>1 R libraries imports, useful functions and data loading</b>	<b>1</b>
1.1 Load Required Packages . . . . .	1
1.2 Sample load . . . . .	2
<b>2 PCA</b>	<b>2</b>
2.1 Eigenvalues and dominant axes analysis . . . . .	4
2.2 Individuals point of view (contribution) . . . . .	4
2.3 Interpreting the axes . . . . .	5
2.3.1 Dimension 1 . . . . .	5
2.3.2 Dimension 2 . . . . .	7
<b>3 Hierarchical Clustering</b>	<b>8</b>
<b>4 K-Means Classification</b>	<b>29</b>
<b>5 CA</b>	<b>53</b>
<b>6 MCA</b>	<b>58</b>
6.1 Eigenvalues and dominant axes analysis.....	60
6.2 Individuals point of view.....	63
6.3 Interpreting map of categories.....	63
6.4 Interpreting the axes association to factor map.....	64
6.5 Hierarchical Clustering (from MCA).....	64
6.6 Comparisons .....	77

## 1 R libraries imports, useful functions and data loading

In this first section we will load all required packages and libraries, and load our data.

### 1.1 Load Required Packages

```

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

requiredPackages <- c("effects", "FactoMineR", "car", "missMDA", "mvoutlier", "chemometrics", "factoextra", "RColorBrewer")

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

```

## 1.2 Sample load

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

# Clean workspace
rm(list=ls())

# Users file path
miquel_fp <- "C:/Users/Miquel/Documents/GitHub/ADEI/"
xavi_fp <- "~/Documents/FIB/ADEI/ADEI/"
filepath <- xavi_fp
filepath <- miquel_fp

# Set working directory
setwd(filepath)

# Load data from file
load(paste0(filepath, "MyOldCars-5000Clean.RData"))

# Index reset
row.names(df) <- NULL
```

## 2 PCA

We have done PCA in order to combine related variables, and focus on uncorrelated or independent ones, especially those along which the observations have high variance. Also, to obtain a smaller set of variables that explain most of the variance in the original data, in more compact and insightful form.

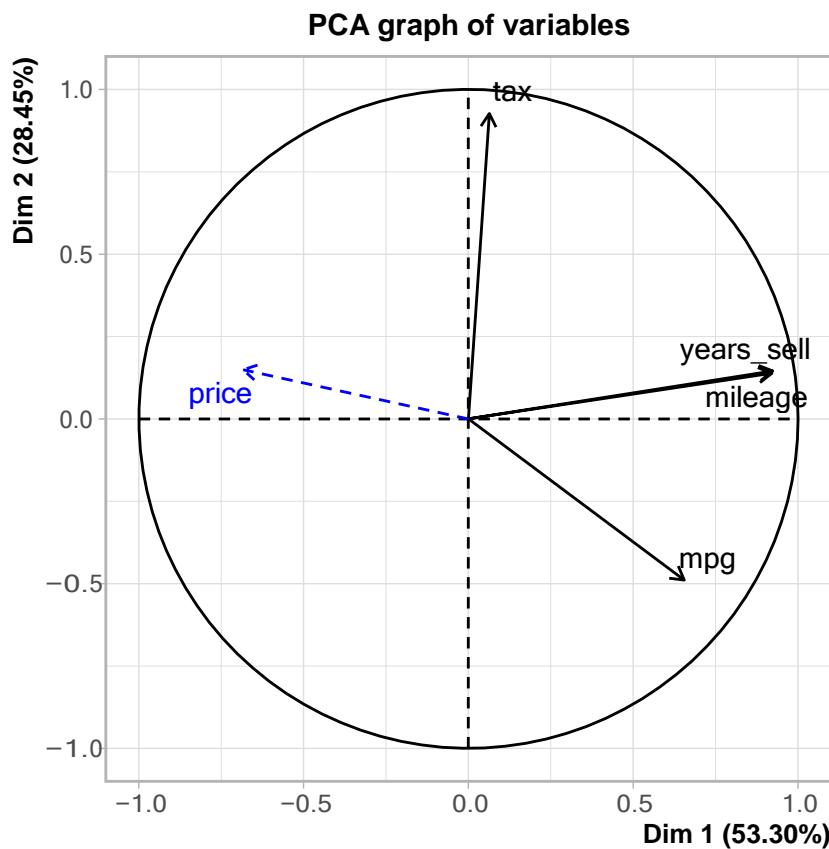
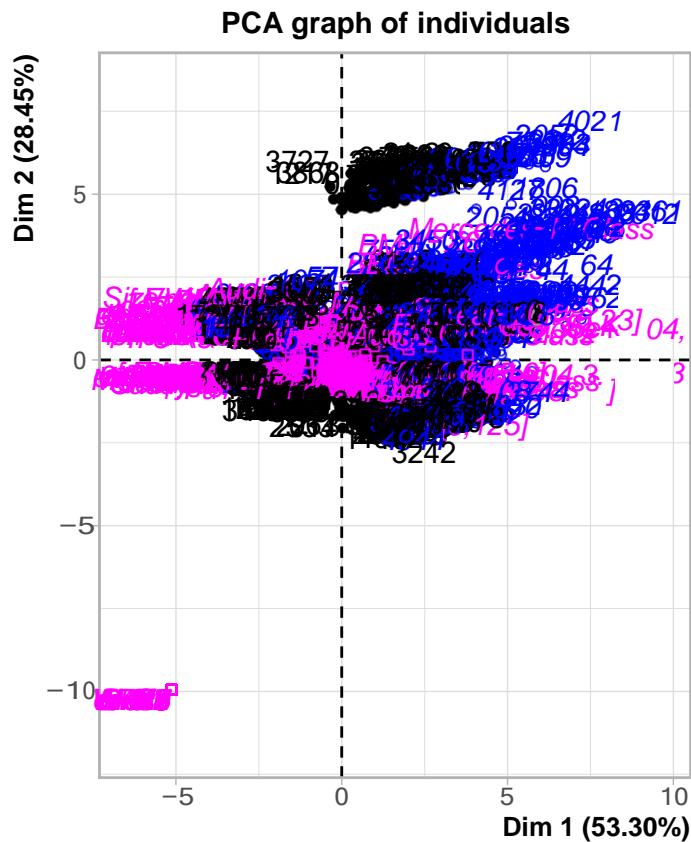
```
names(df)

## [1] "model"           "year"            "price"           "transmission"
## [5] "mileage"         "fuelType"        "tax"             "mpg"
## [9] "engineSize"      "manufacturer"   "years_sell"     "totalMOE"
## [13] "aux_price"       "aux_mileage"    "aux_tax"        "aux_mpg"
## [17] "aux_years_sell" "Audi"           "mout"

vars_con<-names(df)[c(5, 7, 8, 11)]
vars_res<-names(df)[c(3, 18)]
vars_dis<-names(df)[c(1, 2, 4, 6, 9, 10, 13, 14, 15, 16, 17)]

# Multivariate outliers should be included as supplementary observations
```

```
11 <- which( df$mout == "YesMOut")
res.pca<-PCA(df[, c(vars_res, vars_dis, vars_con)], quali.sup=c(2:13), quanti.sup=c(1), ind.sup=11)
```



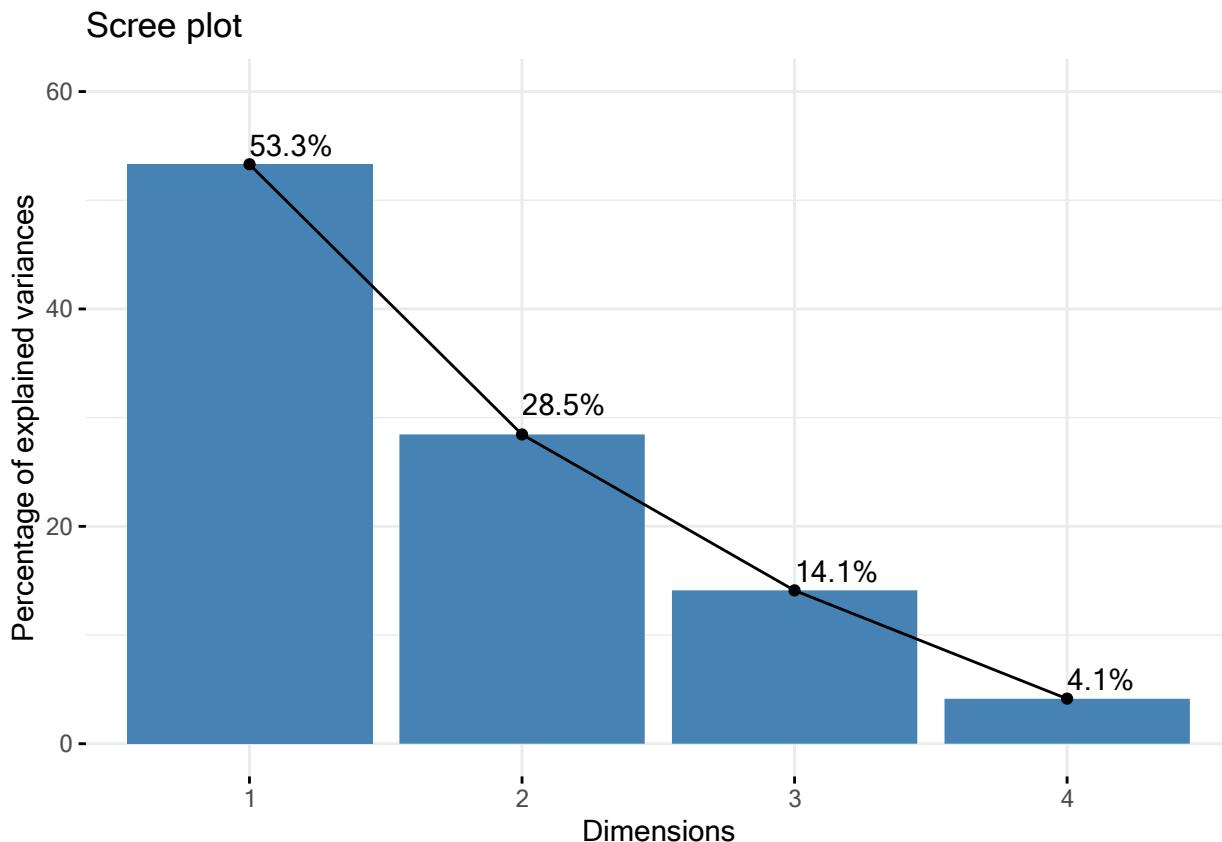
As we can see in the first two plots tax and years\_sell/mileage have almost an angle of 90 degrees, meaning that probably these variable are not that much related. Instead, years\_sell and mileage are positively related. Price although it doesn't completely go in the opposite direction we can see that is inversely related to years\_sell and mileage.

## 2.1 Eigenvalues and dominant axes analysis

```
res.pca$eig[, 1:3]
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
##  comp 1  2.1320997           53.302492           53.30249
##  comp 2  1.1381377           28.453443           81.75594
##  comp 3  0.5642237           14.105591           95.86153
##  comp 4  0.1655389            4.138473          100.00000
```

```
fviz_eig(res.pca, addlabels = TRUE, ylim = c(0, 60))
```



The eigenvalue of the first component is larger than the other ones. Starting from the second component with an eigenvalue of 1.138... the consequent eigenvalues get smaller.

According to the Kaiser criteria we will use the first 2 dimensions (the ones with an eigenvalue  $> 1$ ).

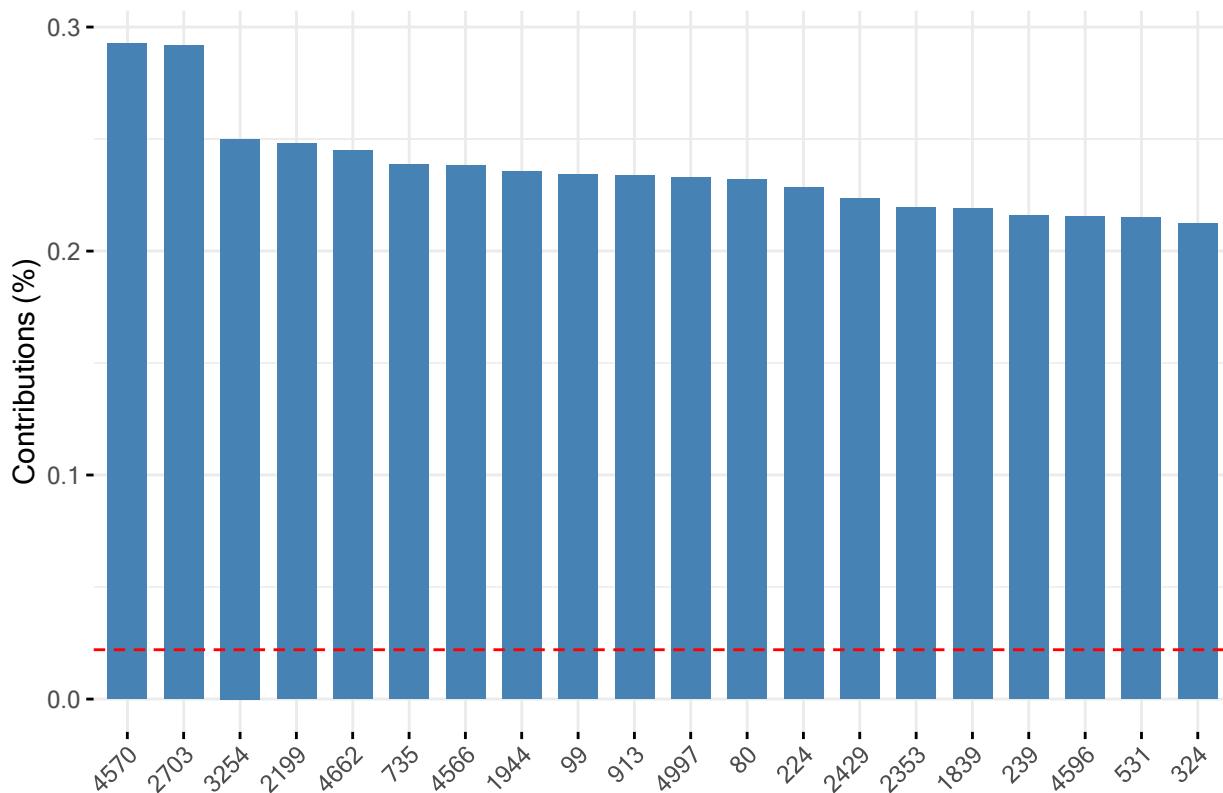
And according to Elbow's rule (based on selecting dimensions until the difference in variance of that of the next factorial plane is almost the same as that of the current plane) we can also use the first two dimensions (Dim.1 and Dim.2). The difference between the dimensions 2 and 3 is significantly less than the one we can see between dimensions 1 and 2.

With the two dimensions we capture/retain an 81.76% of the total variance.

## 2.2 Individuals point of view (contribution)

```
fviz_contrib(res.pca, choice = "ind", axes = 1:2, top = 20)
```

## Contribution of individuals to Dim-1-2



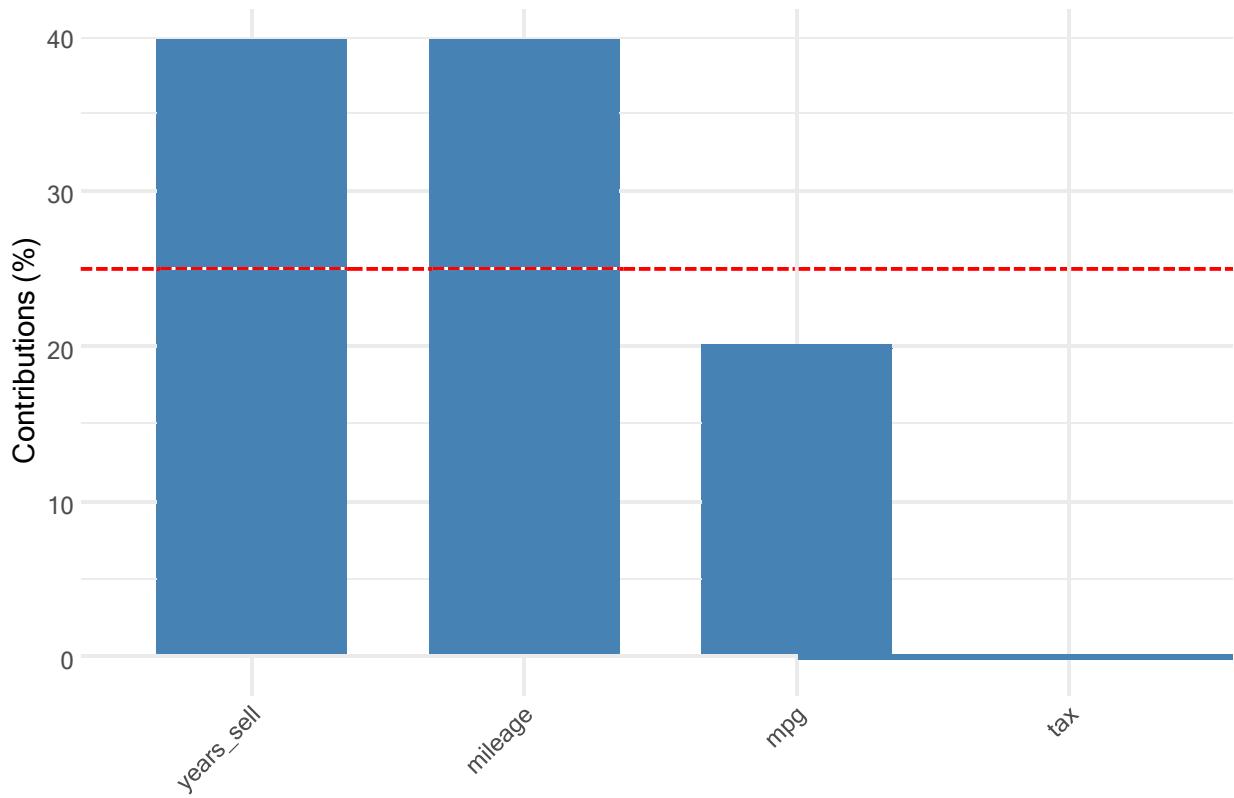
As we can see from the Contribution of individuals to Dim-1-2 plot, the 20 most contributive individuals in descending order are 4570, 2703, 3254, 2199, 4662, 735, 4566, 1944, 99, 913, 4997, 80, 224, 2429, 2353, 1839, 239, 4596, 531 and 324.

## 2.3 Interpreting the axes

### 2.3.1 Dimension 1

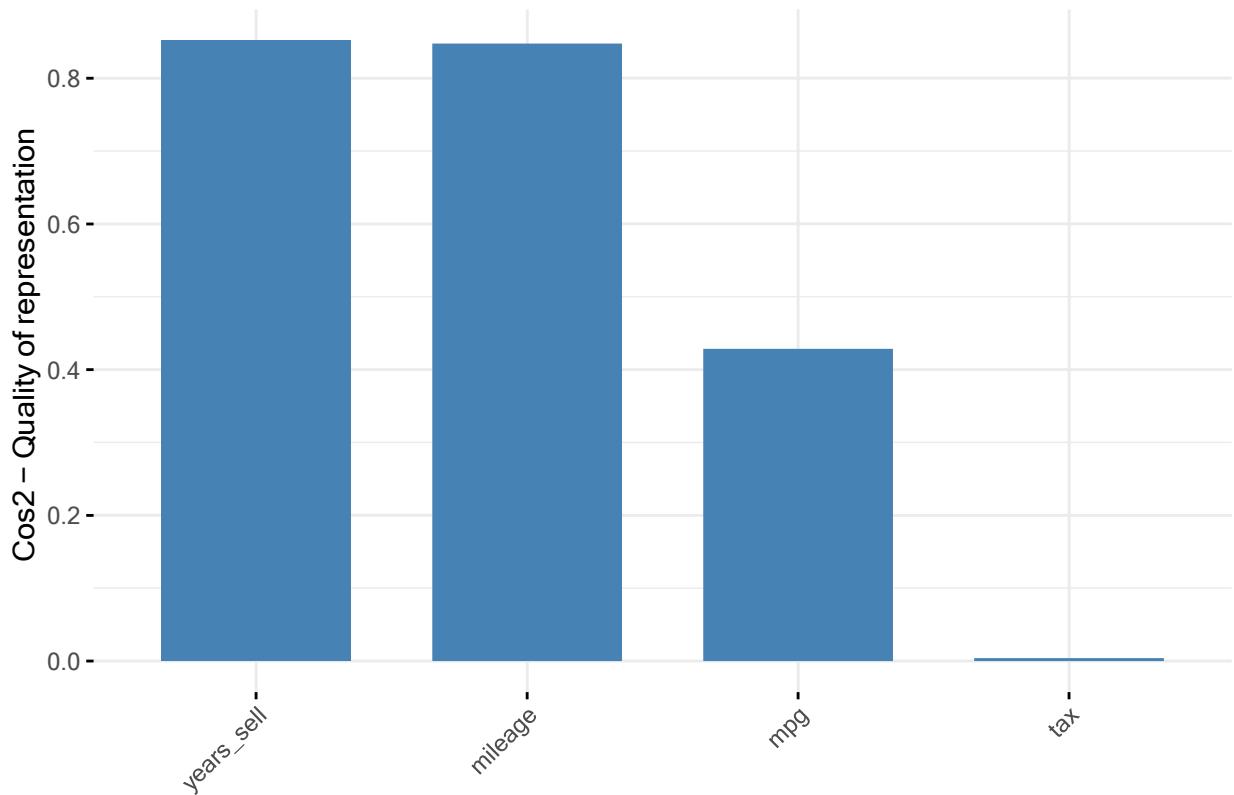
```
fviz_contrib(res.pca, choice = "var", axes = 1)
```

## Contribution of variables to Dim-1



```
fviz_cos2(res.pca, choice = "var", axes = 1)
```

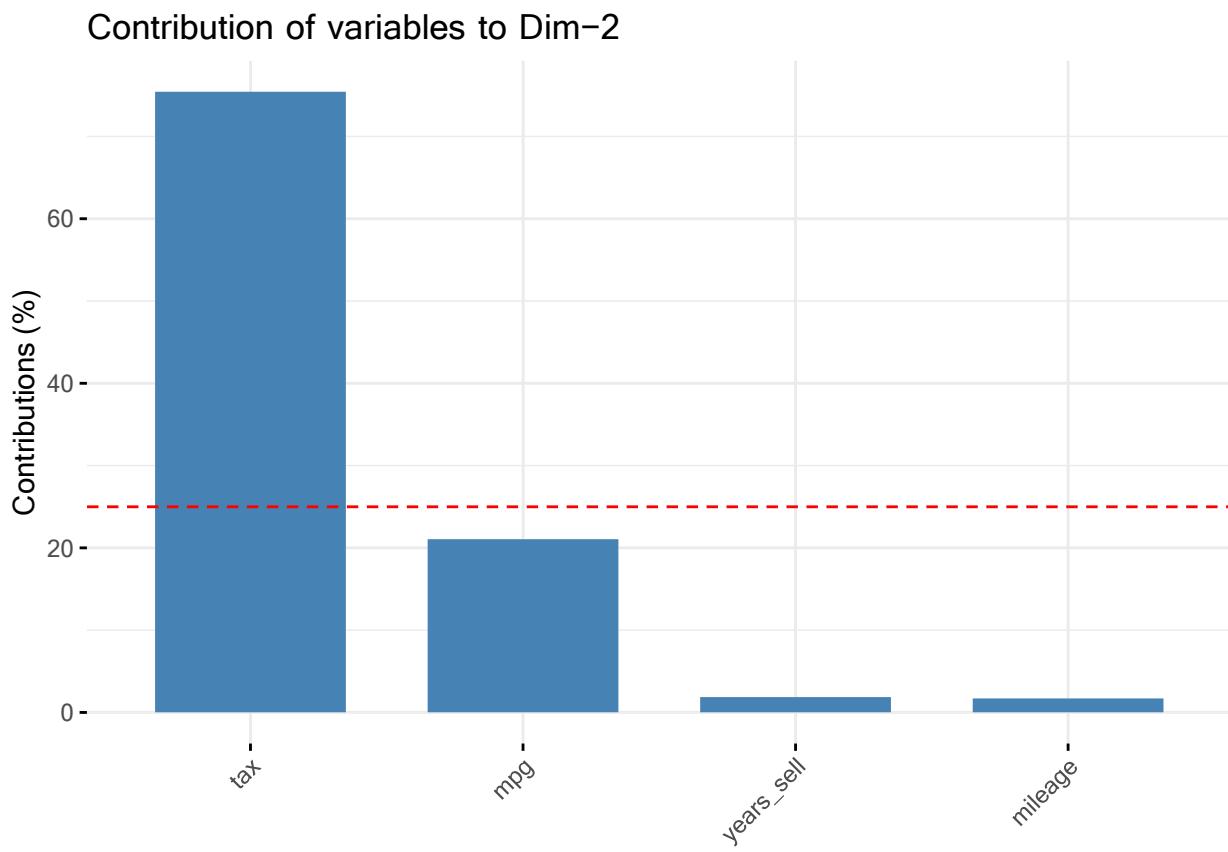
## Cos2 of variables to Dim-1



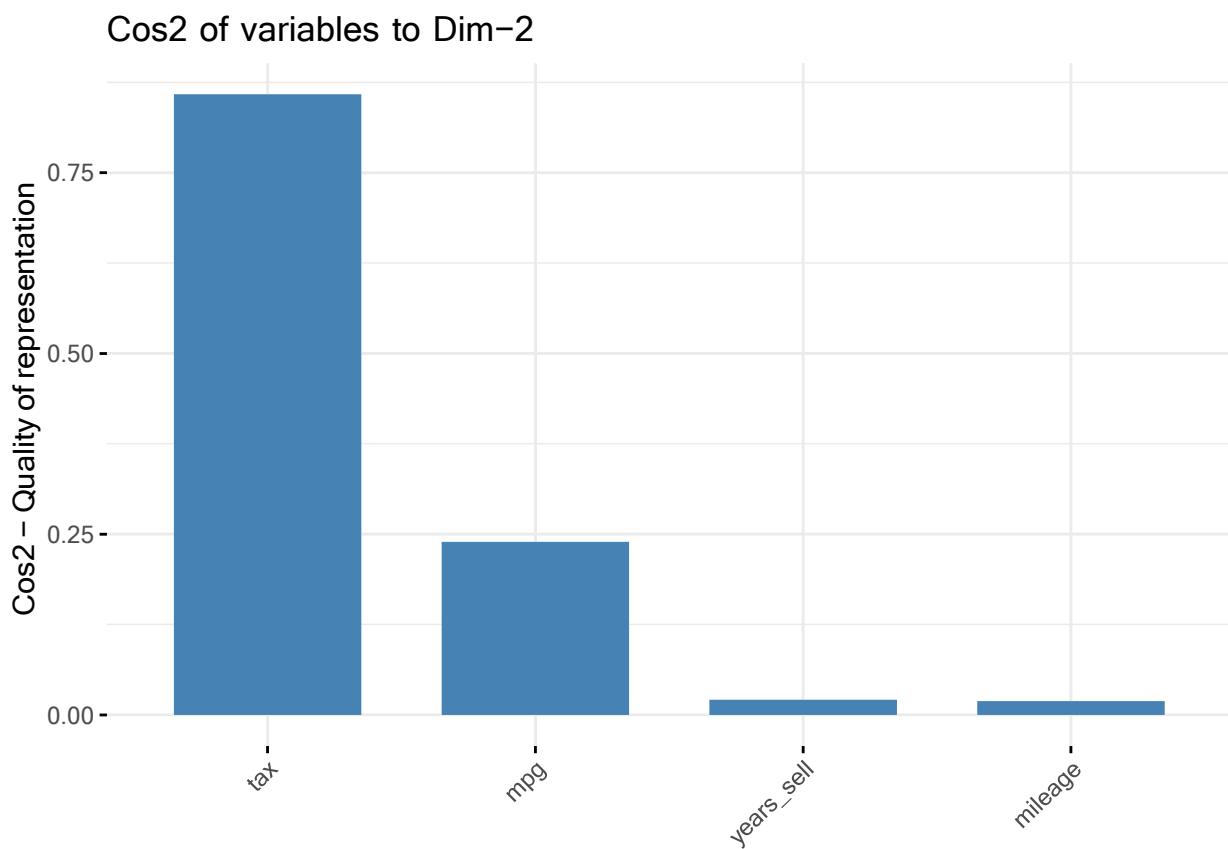
The years\_sell, mileage, mpg and tax variables have a positive first dimension. Dimension 1 shows a positive correlation to years\_sell and mileage. Price target variable (supplementary) is negatively associated to the first factorial axis. As we can see with cos2 years\_sell and mileage are the best represented in the first dimension.

### 2.3.2 Dimension 2

```
fviz_contrib(res.pca, choice = "var", axes = 2)
```



```
fviz_cos2(res.pca, choice = "var", axes = 2)
```

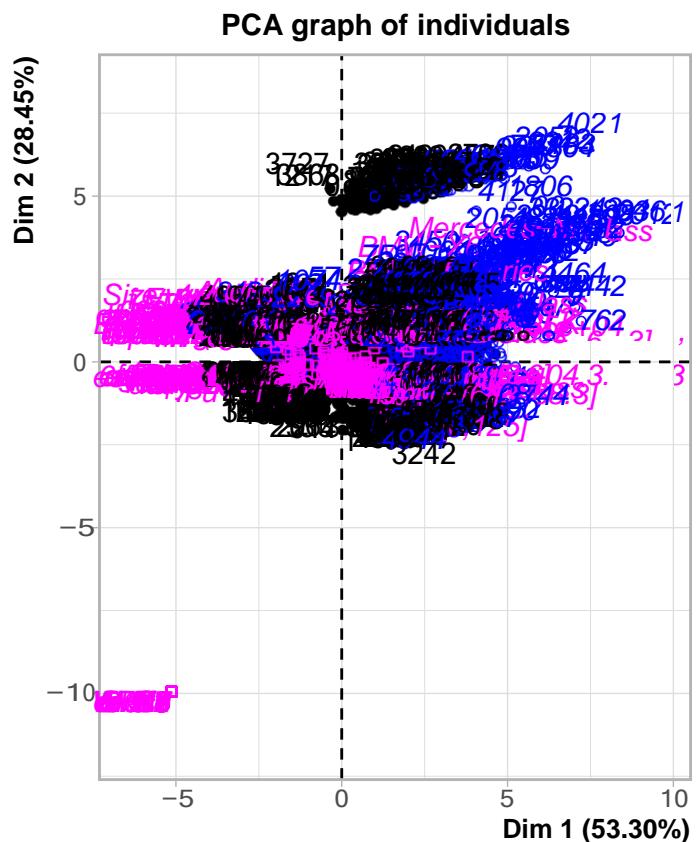


The tax, years\_sell and mileage variables have a positive second dimension and the mpg variable has a negative second dimension. Dimension 2 shows the greatest positive correlation to tax and the lowest negative correlation to mpg. As we can see with  $\cos 2$  tax is the best represented in the second dimension.

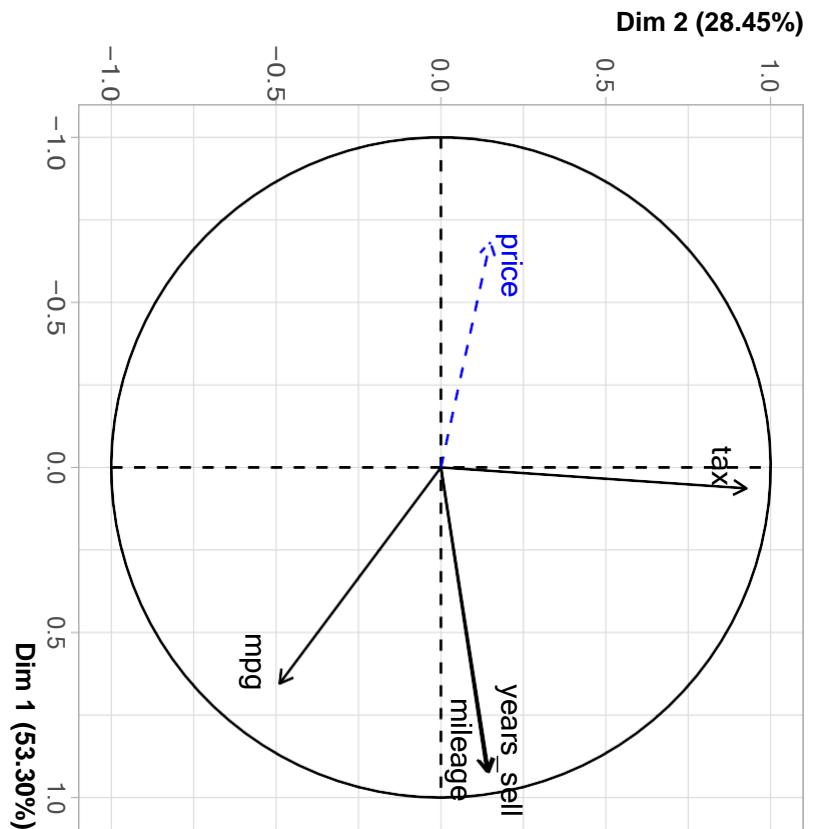
### 3 Hierarchical Clustering

As we have seen previously in this report we should pick 2 clusters. But we will pick 7 because it distributes the data more proportionately among the clusters. But you also can see in the table below that the number of individuals for each cluster is still quite unbalanced.

```
res.pca<-PCA(df[, c(vars_res, vars_dis, vars_con)], quali.sup=c(2:13), quanti.sup= c(1), ind.sup = 11 )
```

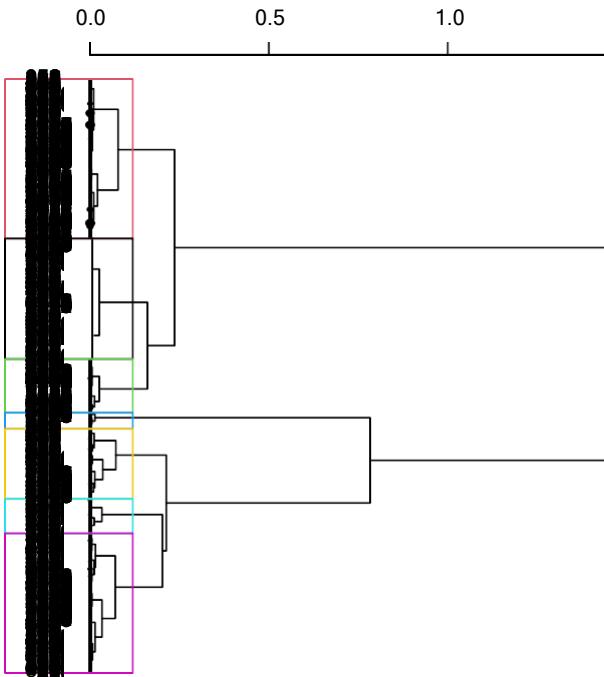
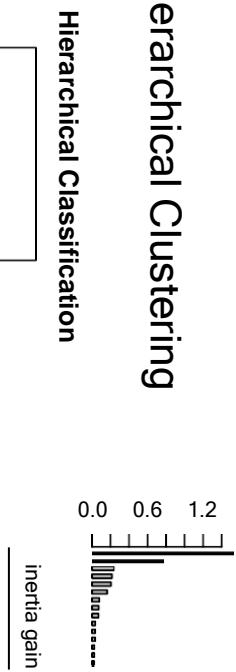


## PCA graph of variables

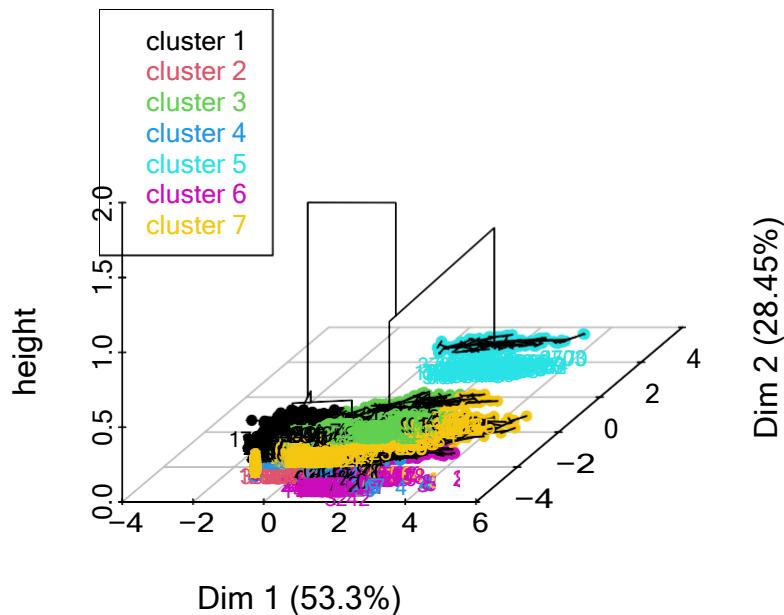


```
res.hcpc<-HCPC(res.pca, nb.clust=7, graph=T)
```

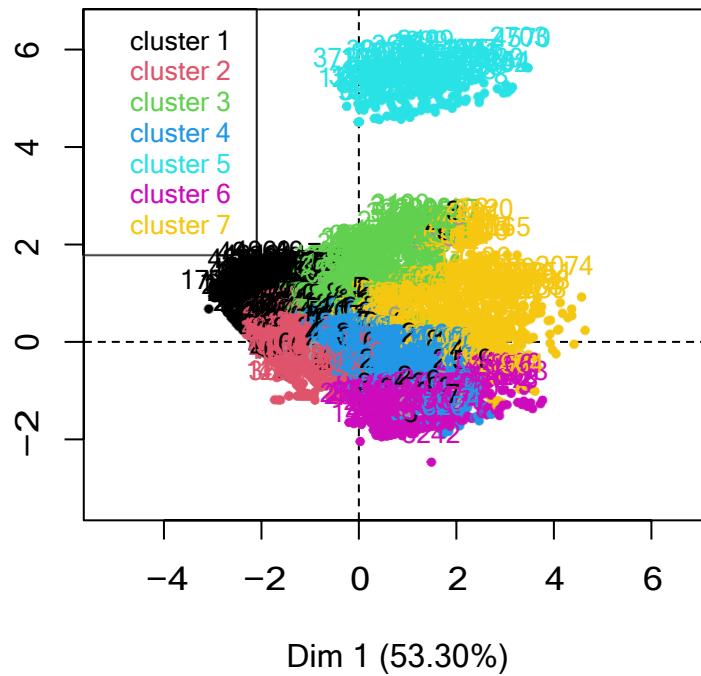
## Hierarchical Clustering



## Hierarchical clustering on the factor map



## Factor map



```
table(res.hpc$data.clust$clust)
```

```
##  
## 1 2 3 4 5 6 7  
## 1115 1085 572 950 123 263 429
```

```
res.hcpc$desc.var$test.chi2
```

```
##          p.value  df
## year      0.000000e+00 48
## engineSize 0.000000e+00 96
## aux_price  0.000000e+00 18
## aux_mileage 0.000000e+00 18
## aux_tax    0.000000e+00 12
## aux_mpg    0.000000e+00 18
## aux_years_sell 0.000000e+00 18
## model     1.400798e-299 462
## fuelType   4.027676e-107 18
## transmission 1.786437e-77 12
## manufacturer 1.922873e-34 18
## Audi      2.820428e-17  6
```

```
res.hcpc$desc.var$category
```

```
## $ '1 '
## 
## aux_years_sell=f.years_sell-[2, 3]
## aux_mpg=f.mpg-[8. 8, 44. 8]
## year=f.Year-2019
## aux_price=f.price-(26, 110]
## aux_mileage=f.mileage-[4, 6e+03]
## fuelType=f.Fuel-Petrol
## aux_tax=f.tax-(125, 145]
## year=f.Year-2020
## engineSize=f.EngSize-1. 5
## transmission=f.Trans-SemiAuto
## aux_mileage=f.mileage-(6e+03, 1. 65e+04]
## engineSize=f.EngSize-3
## model=VW-Sharan
## engineSize=f.EngSize-1. 3
## model=Audi-Q5
## manufacturer=Audi
## Audi=Audi Yes
## transmission=f.Trans-Automatic
## model=VW-Tiguan Allspace
## model=Audi-Q2
## model=Mercedes-X-CLASS
## model=VW-T-Roc
## model=Mercedes-GLC Class
## engineSize=f.EngSize-2. 3
## model=VW-Amarok
## model=VW-Touareg
## model=BMW-M4
## model=Audi-Q7
## model=VW-Caravelle
## model=BMW-Z4
## model=Mercedes-GLE Class
## aux_price=f.price-(19. 5, 26]
## model=BMW-M2
## model=Audi-Q3
## model=VW-Tiguan
## model=VW-Golf SV
## model=Mercedes-SL CLASS
## model=Audi-TT
## engineSize=f.EngSize-2
## engineSize=f.EngSize-2. 9
## model=BMW-X4
## model=Mercedes-GLS Class
```

	Cla/Mod	Mod/Cla	Global
54. 4671690	90. 7623318	40. 9521710	
78. 3303730	79. 1031390	24. 8181618	
52. 7296082	73. 6322870	34. 3178312	
58. 4562012	60. 4484305	25. 4132687	
55. 6203164	59. 9103139	26. 4712365	
38. 7080366	68. 2511211	43. 3325986	
32. 4631861	86. 9955157	65. 8584968	
63. 4551495	17. 1300448	6. 6343399	
47. 4766355	22. 7802691	11. 7919330	
32. 2816901	51. 3901345	39. 1227683	
34. 6787479	37. 7578475	26. 7577695	
39. 8706897	16. 5919283	10. 2270223	
90. 0000000	1. 6143498	0. 4408199	
56. 5789474	3. 8565022	1. 6751157	
55. 0000000	3. 9461883	1. 7632797	
31. 6666667	27. 2645740	21. 1593564	
31. 6666667	27. 2645740	21. 1593564	
30. 2573203	30. 5829596	24. 8402028	
85. 7142857	1. 0762332	0. 3085739	
48. 1012658	3. 4080717	1. 7412387	
90. 9090909	0. 8968610	0. 2424510	
50. 7936508	2. 8699552	1. 3885828	
42. 7272727	4. 2152466	2. 4245096	
90. 0000000	0. 8071749	0. 2204100	
78. 5714286	0. 9865471	0. 3085739	
58. 0645161	1. 6143498	0. 6832709	
88. 8888889	0. 7174888	0. 1983690	
66. 6666667	1. 0762332	0. 3967379	
100. 0000000	0. 5381166	0. 1322460	
75. 0000000	0. 8071749	0. 2644920	
46. 9387755	2. 0627803	1. 0800088	
28. 1840592	30. 7623318	26. 8238924	
100. 0000000	0. 4484305	0. 1102050	
36. 4285714	4. 5739910	3. 0857395	
33. 7579618	4. 7533632	3. 4604364	
47. 8260870	0. 9865471	0. 5069429	
44. 8275862	1. 1659193	0. 6391889	
44. 4444444	1. 0762332	0. 5951069	
26. 2526096	45. 1121076	42. 2305488	
53. 8461538	0. 6278027	0. 2865330	
50. 0000000	0. 7174888	0. 3526559	
54. 5454545	0. 5381166	0. 2424510	

```

## engineSize=f.EngSize-2.5
## model=Mercedes-S Class
## model=Mercedes-GL Class
## model=Mercedes-GLA Class
## model=VW-Scirocco
## model=Mercedes-E Class
## fuelType=f.Fuel-Hybrid
## year=f.Year-2018
## model=Audi-A6
## engineSize=f.EngSize-1.8
## model=BMW-4 Series
## model=BMW-1 Series
## model=BMW-5 Series
## engineSize=f.EngSize-1
## model=BMW-3 Series
## manufacturer=BMW
## Audi=Audi No
## model=Mercedes-A Class
## model=VW-Up
## year=f.Year-2013
## aux_mpg=f.mpg-(44.8, 53.3]
## model=VW-Polo
## engineSize=f.EngSize-1.2
## engineSize=f.EngSize-1.6
## year=f.Year-2014
## engineSize=f.EngSize-1.4
## engineSize=f.EngSize-2.1
## aux_tax=f.tax-[0, 125]
## aux_tax=f.tax-(145, 580]
## year=f.Year-2015
## transmission=f.Trans-Manual
## aux_price=f.price-(14, 19.5]
## aux_years_sell=f.years_sell-(3, 5]
## fuelType=f.Fuel-Diesel
## aux_years_sell=f.years_sell-(6, 23]
## year=f.Year-2017
## aux_years_sell=f.years_sell-(5, 6]
## year=f.Year-2016
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05]
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04]
## aux_mpg=f.mpg-(61.4, 88.3]
## aux_price=f.price-[0.65, 14]
## aux_mpg=f.mpg-(53.3, 61.4]
## aux_years_sell=f.years_sell-[2, 3]
## aux_mpg=f.mpg-[8.8, 44.8]
## year=f.Year-2019
## aux_price=f.price-(26, 110]
## aux_mileage=f.mileage-[4, 6e+03]
## fuelType=f.Fuel-Petrol
## aux_tax=f.tax-(125, 145]
## year=f.Year-2020
## engineSize=f.EngSize-1.5
## transmission=f.Trans-SemiAuto
## aux_mileage=f.mileage-(6e+03, 1.65e+04]
## engineSize=f.EngSize-3
## model=VW-Sharan
## engineSize=f.EngSize-1.3
## model=Audi-Q5
## manufacturer=Audi
## Audi=Audi Yes
## transmission=f.Trans-Automatic
## model=VW-Tiguan Allspace
## model=Audi-Q2

```

	66.6666667	0.3587444	0.1322460
	0.0000000	0.0000000	0.2644920
	0.0000000	0.0000000	0.2644920
	14.8148148	1.0762332	1.7853207
	4.7619048	0.0896861	0.4628609
	17.1597633	2.6008969	3.7249284
	9.5238095	0.3587444	0.9257218
	20.1257862	8.6098655	10.5135552
	10.7142857	0.5381166	1.2342958
	0.0000000	0.0000000	0.3967379
	12.2222222	0.9865471	1.9836897
	13.8888889	2.2421525	3.9673793
	5.6338028	0.3587444	1.5649107
	15.7107232	5.6502242	8.8384395
	10.9523810	2.0627803	4.6286092
	18.0921053	14.7982063	20.1013886
	22.6726307	72.7354260	78.8406436
	9.9206349	2.2421525	5.5543311
	0.0000000	0.0000000	1.8734847
	0.0000000	0.0000000	1.8955257
	17.7591463	20.8968610	28.9177871
	8.5798817	2.6008969	7.4498567
	0.0000000	0.0000000	2.6669605
	5.2287582	1.4349776	6.7445449
	0.0000000	0.0000000	3.5706414
	2.9411765	0.7174888	5.9951510
	2.7472527	0.8968610	8.0229226
	0.0000000	0.0000000	5.7747410
	11.2665113	13.0044843	28.3667622
	0.0000000	0.0000000	7.7363897
	12.2935780	18.0269058	36.0370289
	8.3050847	8.7892377	26.0083756
	7.8866769	9.2376682	28.7855411
	13.7753077	31.1210762	55.5212696
	0.0000000	0.0000000	13.3568437
	0.8443908	0.6278027	18.2719859
	0.0000000	0.0000000	16.9054441
	0.0000000	0.0000000	16.9054441
	0.0000000	0.0000000	19.9030196
	2.1328958	2.3318386	26.8679744
	0.0000000	0.0000000	21.7544633
	0.0000000	0.0000000	21.7544633
	0.0000000	0.0000000	24.5095878
	p.value	v.test	
	0.000000e+00	Inf	
	0.000000e+00	Inf	
	3.070798e-216	31.386725	
	3.426577e-192	29.571981	
	6.734426e-172	27.949238	
	1.961580e-83	19.351980	
	1.168407e-73	18.155182	
	7.800606e-50	14.842324	
	7.124526e-35	12.319374	
	9.589759e-22	9.581228	
	1.019223e-20	9.334028	
	1.358027e-14	7.700248	
	1.121696e-09	6.091054	
	2.304898e-09	5.974717	
	5.055685e-09	5.845329	
	1.798473e-08	5.630347	
	1.798473e-08	5.630347	
	4.847435e-07	5.032255	
	2.660194e-06	4.695451	
	5.445340e-06	4.546855	

```

## model=Mercedes-X-CLASS
## model=VW-T-Roc
## model=Mercedes-GLC Class
## engineSize=f. EngSize-2. 3
## model=VW-Amarok
## model=VW-Touareg
## model=BMW-M4
## model=Audi-Q7
## model=VW-Caravelle
## model=BMW-Z4
## model=Mercedes-GLE Class
## aux_price=f. price-(19. 5, 26]
## model=BMW-M2
## model=Audi-Q3
## model=VW-Tiguan
## model=VW-Golf SV
## model=Mercedes-SL CLASS
## model=Audi-TT
## engineSize=f. EngSize-2
## engineSize=f. EngSize-2. 9
## model=BMW-X4
## model=Mercedes-GLS Class
## engineSize=f. EngSize-2. 5
## model=Mercedes-S Class
## model=Mercedes-GL Class
## model=Mercedes-GLA Class
## model=VW-Scirocco
## model=Mercedes-E Class
## fuelType=f. Fuel-Hybrid
## year=f. Year-2018
## model=Audi-A6
## engineSize=f. EngSize-1. 8
## model=BMW-4 Series
## model=BMW-1 Series
## model=BMW-5 Series
## engineSize=f. EngSize-1
## model=BMW-3 Series
## manufacturer=BMW
## Audi=Audi No
## model=Mercedes-A Class
## model=VW-Up
## year=f. Year-2013
## aux_mpg=f. mpg-(44. 8, 53. 3]
## model=VW-Polo
## engineSize=f. EngSize-1. 2
## engineSize=f. EngSize-1. 6
## year=f. Year-2014
## engineSize=f. EngSize-1. 4
## engineSize=f. EngSize-2. 1
## aux_tax=f. tax-[0, 125]
## aux_tax=f. tax-(145, 580]
## year=f. Year-2015
## transmission=f. Trans-Manual
## aux_price=f. price-(14, 19. 5]
## aux_years_sell=f. years_sell-(3, 5]
## fuelType=f. Fuel-Diesel
## aux_years_sell=f. years_sell-(6, 23]
## year=f. Year-2017
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]
## aux_mpg=f. mpg-(61. 4, 88. 3]
## aux_price=f. price-[0. 65, 14]
6. 861728e-06 4. 497933
7. 186481e-06 4. 488089
2. 544234e-05 4. 210840
2. 567567e-05 4. 208777
3. 500141e-05 4. 138218
8. 202679e-05 3. 938400
9. 516327e-05 3. 902604
2. 149759e-04 3. 700735
2. 180825e-04 3. 697093
3. 652986e-04 3. 563962
6. 895772e-04 3. 393690
7. 097020e-04 3. 385803
8. 904053e-04 3. 323045
1. 519800e-03 3. 170876
8. 520028e-03 2. 630736
1. 674905e-02 2. 392171
1. 806534e-02 2. 364276
2. 516003e-02 2. 238937
2. 519116e-02 2. 238459
2. 697003e-02 2. 211951
3. 087581e-02 2. 158670
3. 827764e-02 2. 071869
3. 955736e-02 2. 058342
3. 373433e-02 -2. 123233
3. 373433e-02 -2. 123233
3. 348628e-02 -2. 126204
2. 342657e-02 -2. 266405
1. 910976e-02 -2. 343383
1. 584254e-02 -2. 412523
1. 545434e-02 -2. 421553
1. 029583e-02 -2. 565732
6. 171855e-03 -2. 738509
3. 691949e-03 -2. 903349
3. 548332e-04 -3. 571581
2. 786018e-05 -4. 190287
6. 959690e-06 -4. 494918
4. 345937e-07 -5. 053145
1. 880334e-07 -5. 210795
1. 798473e-08 -5. 630347
1. 540776e-09 -6. 040038
2. 988633e-11 -6. 647130
2. 240132e-11 -6. 689448
3. 708387e-12 -6. 947871
5. 875836e-15 -7. 806585
8. 864362e-16 -8. 041641
1. 542555e-20 -9. 290023
5. 476292e-21 -9. 399615
5. 662246e-24 -10. 097580
7. 469337e-33 -11. 938346
6. 184918e-34 -12. 143846
1. 781394e-43 -13. 825793
9. 278471e-46 -14. 199109
8. 518581e-51 -14. 990130
9. 039267e-60 -16. 305368
8. 266194e-72 -17. 919775
5. 628185e-80 -18. 937229
2. 417339e-81 -19. 102272
1. 252978e-100 -21. 295376
2. 997051e-105 -21. 788279
2. 997051e-105 -21. 788279
2. 039927e-126 -23. 916968
7. 040396e-134 -24. 623712
5. 656096e-140 -25. 186326
5. 656096e-140 -25. 186326

```

```

## aux_mpg=f.mpg-(53.3,61.4] 6.907205e-161 -27.028119
## Cla/Mod Mod/Cla Global
## $ '2 '
## aux_years_sell=f.years_sell-[2,3] 45.209903 77.4193548 40.9521710
## year=f.Year-2019 46.885035 67.2811060 34.3178312
## aux_mpg=f.mpg-(44.8,53.3] 43.597561 52.7188940 28.9177871
## aux_tax=f.tax-(125,145] 32.061580 88.2949309 65.8584968
## aux_mileage=f.mileage-[4,6e+03] 44.046628 48.7557604 26.4712365
## aux_mileage=f.mileage-(6e+03,1.65e+04] 42.998353 48.1105991 26.7577695
## year=f.Year-2018 44.863732 19.7235023 10.5135552
## aux_price=f.price-(19.5,26] 35.168447 39.4470046 26.8238924
## engineSize=f.EngSize-1 41.645885 15.3917051 8.8384395
## fuelType=f.Fuel-Diesel 28.384279 65.8986175 55.5212696
## Audi=Audi No 26.251048 86.5437788 78.8406436
## model=BMW-2 Series 51.538462 6.1751152 2.8653295
## aux_mpg=f.mpg-(53.3,61.4] 31.205036 31.9815668 24.5095878
## aux_price=f.price-(26,110] 30.182134 32.0737327 25.4132687
## model=VW-T-Cross 76.000000 1.7511521 0.5510249
## manufacturer=BMW 30.482456 25.6221198 20.1013886
## year=f.Year-2020 36.544850 10.1382488 6.6343399
## model=Mercedes-A Class 36.111111 8.3870968 5.5543311
## engineSize=f.EngSize-1.6 34.313725 9.6774194 6.7445449
## model=BMW-5 Series 45.070423 2.9493088 1.5649107
## model=VW-Arteon 59.259259 1.4746544 0.5951069
## engineSize=f.EngSize-1.3 43.421053 3.0414747 1.6751157
## engineSize=f.EngSize-2 26.670146 47.0967742 42.2305488
## model=BMW-X2 59.090909 1.1981567 0.4849019
## model=BMW-4 Series 40.000000 3.3179724 1.9836897
## model=VW-T-Roc 42.857143 2.4884793 1.3885828
## model=VW-Up 38.823529 3.0414747 1.8734847
## engineSize=f.EngSize-2.2 70.000000 0.6451613 0.2204100
## model=Mercedes-B Class 41.379310 2.2119816 1.2783778
## manufacturer=Mercedes 26.672039 30.5069124 27.3528764
## transmission=f.Trans-SemiAuto 25.971831 42.4884793 39.1227683
## model=Mercedes-C Class 29.673591 9.2165899 7.4278157
## model=Mercedes-S Class 58.333333 0.6451613 0.2644920
## model=BMW-X3 42.105263 1.4746544 0.8375579
## model=Audi-A8 53.333333 0.7373272 0.3306149
## model=BMW-3 Series 30.952381 5.9907834 4.6286092
## model=Mercedes-E Class 30.769231 4.7926267 3.7249284
## model=Audi-Q2 34.177215 2.4884793 1.7412387
## model=Mercedes-X-CLASS 0.000000 0.0000000 0.2424510
## model=Mercedes-GLS Class 0.000000 0.0000000 0.2424510
## model=BMW-7 Series 0.000000 0.0000000 0.2424510
## model=Mercedes-GL Class 0.000000 0.0000000 0.2644920
## model=BMW-Z4 0.000000 0.0000000 0.2644920
## transmission=f.Trans-Manual 22.079511 33.2718894 36.0370289
## engineSize=f.EngSize-0 0.000000 0.0000000 0.2865330
## model=Audi-A3 17.032967 2.8571429 4.0114613
## model=VW-Tiguan Allspace 0.000000 0.0000000 0.3085739
## model=VW-Amarok 0.000000 0.0000000 0.3085739
## model=BMW-X4 0.000000 0.0000000 0.3526559
## model=Mercedes-V Class 0.000000 0.0000000 0.3967379
## model=Audi-Q7 0.000000 0.0000000 0.3967379
## model=VW-Sharan 0.000000 0.0000000 0.4408199
## model=Mercedes-SL CLASS 3.448276 0.0921659 0.6391889
## model=VW-Scirocco 0.000000 0.0000000 0.4628609
## model=BMW-X5 0.000000 0.0000000 0.5289839
## model=Audi-TT 0.000000 0.0000000 0.5951069
## model=VW-Touareg 0.000000 0.0000000 0.6832709
## model=Audi-A1 10.000000 1.1981567 2.8653295
## model=VW-Tiguan 9.554140 1.3824885 3.4604364
## model=Mercedes-GLE Class 0.000000 0.0000000 1.0800088

```

```

## model=Audi-Q5
## aux_years_sell=f.years_sell-(3, 5]
## engineSize=f.EngSize-1.4
## aux_mpg=f.mpg-(61.4, 88.3]
## engineSize=f.EngSize-2.1
## aux_price=f.price-(14, 19.5]
## model=Audi-Q3
## year=f.Year-2013
## engineSize=f.EngSize-1.2
## manufacturer=Audi
## Audi=Audi Yes
## engineSize=f.EngSize-3
## fuelType=f.Fuel-Petrol
## year=f.Year-2014
## aux_price=f.price-[0.65, 14]
## aux_tax=f.tax-[0, 125]
## year=f.Year-2015
## aux_tax=f.tax-(145, 580]
## year=f.Year-2017
## aux_years_sell=f.years_sell-(6, 23]
## aux_years_sell=f.years_sell-(5, 6]
## year=f.Year-2016
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04]
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05]
## aux_mpg=f.mpg-[8.8, 44.8]
## aux_years_sell=f.years_sell-[2, 3]
## year=f.Year-2019
## aux_mpg=f.mpg-(44.8, 53.3]
## aux_tax=f.tax-(125, 145]
## aux_mileage=f.mileage-[4, 6e+03]
## aux_mileage=f.mileage-(6e+03, 1.65e+04]
## year=f.Year-2018
## aux_price=f.price-(19.5, 26]
## engineSize=f.EngSize-1
## fuelType=f.Fuel-Diesel
## Audi=Audi No
## model=BMW-2 Series
## aux_mpg=f.mpg-(53.3, 61.4]
## aux_price=f.price-(26, 110]
## model=VW-T-Cross
## manufacturer=BMW
## year=f.Year-2020
## model=Mercedes-A Class
## engineSize=f.EngSize-1.6
## model=BMW-5 Series
## model=VW-Arteon
## engineSize=f.EngSize-1.3
## engineSize=f.EngSize-2
## model=BMW-X2
## model=BMW-4 Series
## model=VW-T-Roc
## model=VW-Up
## engineSize=f.EngSize-2.2
## model=Mercedes-B Class
## manufacturer=Mercedes
## transmission=f.Trans-SemiAuto
## model=Mercedes-C Class
## model=Mercedes-S Class
## model=BMW-X3
## model=Audi-A8
## model=BMW-3 Series
## model=Mercedes-E Class
## model=Audi-Q2
3. 750000 0. 2764977 1. 7632797
18. 759571 22. 5806452 28. 7855411
9. 926471 2. 4884793 5. 9951510
16. 818642 15. 2995392 21. 7544633
11. 538462 3. 8709677 8. 0229226
17. 288136 18. 8018433 26. 0083756
4. 285714 0. 5529954 3. 0857395
0. 000000 0. 0000000 1. 8955257
1. 652893 0. 1843318 2. 6669605
15. 208333 13. 4562212 21. 1593564
15. 208333 13. 4562212 21. 1593564
10. 560345 4. 5161290 10. 2270223
17. 955239 32. 5345622 43. 3325986
0. 000000 0. 0000000 3. 5706414
10. 638298 9. 6774194 21. 7544633
0. 000000 0. 0000000 5. 7747410
0. 000000 0. 0000000 7. 7363897
9. 867910 11. 7050691 28. 3667622
3. 739445 2. 8571429 18. 2719859
0. 000000 0. 0000000 13. 3568437
0. 000000 0. 0000000 16. 9054441
0. 000000 0. 0000000 16. 9054441
2. 789171 3. 1336406 26. 8679744
0. 000000 0. 0000000 19. 9030196
0. 000000 0. 0000000 24. 8181618
p. value v. test
1. 802046e-174 28. 160111
5. 785152e-146 25. 727312
5. 114833e-82 19. 183186
5. 517244e-81 19. 059142
2. 653358e-75 18. 361865
4. 931594e-69 17. 560640
1. 619144e-26 10. 656894
1. 053248e-25 10. 481266
1. 769201e-16 8. 236771
1. 917669e-15 7. 946557
1. 538197e-13 7. 383813
5. 604083e-12 6. 889368
1. 283188e-10 6. 429151
1. 358003e-08 5. 678602
5. 858820e-08 5. 423057
3. 459745e-07 5. 096509
3. 737191e-07 5. 081879
7. 910134e-06 4. 467602
2. 131118e-05 4. 250691
9. 003901e-05 3. 915977
1. 041471e-04 3. 880721
1. 738740e-04 3. 754232
2. 086889e-04 3. 708259
5. 088414e-04 3. 476058
6. 544851e-04 3. 407968
9. 088825e-04 3. 317311
2. 079432e-03 3. 078647
2. 927648e-03 2. 975234
3. 234960e-03 2. 944482
7. 990628e-03 2. 652466
9. 439525e-03 2. 595712
1. 166698e-02 2. 522060
1. 300157e-02 2. 483726
1. 359369e-02 2. 467825
1. 603097e-02 2. 408210
1. 708746e-02 2. 384820
3. 786949e-02 2. 076265
3. 790334e-02 2. 075899

```

```

## model=Mercedes-X-CLASS
## model=Mercedes-GLS Class
## model=BMW-7 Series
## model=Mercedes-GL Class
## model=BMW-Z4
## transmission=f. Trans-Manual
## engineSize=f. EngSize-0
## model=Audi-A3
## model=VW-Tiguan Allspace
## model=VW-Amarok
## model=BMW-X4
## model=Mercedes-V Class
## model=Audi-Q7
## model=VW-Sharan
## model=Mercedes-SL CLASS
## model=VW-Scirocco
## model=BMW-X5
## model=Audi-TT
## model=VW-Touareg
## model=Audi-A1
## model=VW-Tiguan
## model=Mercedes-GLE Class
## model=Audi-Q5
## aux_years_sell=f. years_sell-(3, 5]
## engineSize=f. EngSize-1.4
## aux_mpg=f. mpg-(61.4, 88.3]
## engineSize=f. EngSize-2.1
## aux_price=f. price-(14, 19.5]
## model=Audi-Q3
## year=f. Year-2013
## engineSize=f. EngSize-1.2
## manufacturer=Audi
## Audi=Audi Yes
## engineSize=f. EngSize-3
## fuelType=f. Fuel-Petrol
## year=f. Year-2014
## aux_price=f. price-[0.65, 14]
## aux_tax=f. tax-[0, 125]
## year=f. Year-2015
## aux_tax=f. tax-(145, 580]
## year=f. Year-2017
## aux_years_sell=f. years_sell-(6, 23]
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## aux_mileage=f. mileage-(1.65e+04, 3.35e+04]
## aux_mileage=f. mileage-(3.35e+04, 1.16e+05]
## aux_mpg=f. mpg-[8.8, 44.8]
## $ '3 '
## aux_mileage=f. mileage-(1.65e+04, 3.35e+04]
## aux_tax=f. tax-(145, 580]
## year=f. Year-2017
## aux_mpg=f. mpg-(44.8, 53.3]
## aux_years_sell=f. years_sell-(3, 5]
## engineSize=f. EngSize-3
## engineSize=f. EngSize-1.4
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## fuelType=f. Fuel-Petrol
## model=VW-Tiguan
## aux_price=f. price-(14, 19.5]
## model=Mercedes-GLE Class
## model=Audi-Q3
4. 927928e-02 -1. 966167
4. 927928e-02 -1. 966167
4. 927928e-02 -1. 966167
3. 746575e-02 -2. 080652
3. 746575e-02 -2. 080652
2. 921754e-02 -2. 180539
2. 848225e-02 -2. 190579
2. 274406e-02 -2. 277707
2. 165130e-02 -2. 296430
2. 165130e-02 -2. 296430
1. 250870e-02 -2. 497459
7. 224689e-03 -2. 686306
7. 224689e-03 -2. 686306
4. 171619e-03 -2. 864884
3. 929126e-03 -2. 883797
3. 169581e-03 -2. 950794
1. 389668e-03 -3. 196789
6. 089000e-04 -3. 427619
2. 024412e-04 -3. 715950
4. 621518e-05 -4. 073986
3. 130672e-06 -4. 662055
1. 406442e-06 -4. 824090
8. 774816e-07 -4. 917295
1. 467824e-07 -5. 256551
1. 444467e-09 -6. 050445
1. 261615e-09 -6. 072210
6. 619362e-10 -6. 174928
2. 177960e-10 -6. 348237
1. 795469e-10 -6. 377888
4. 790495e-11 -6. 577306
2. 206914e-12 -7. 020743
1. 538197e-13 -7. 383813
1. 538197e-13 -7. 383813
1. 747156e-14 -7. 667988
9. 832501e-17 -8. 306791
2. 330472e-20 -9. 246002
4. 688957e-32 -11. 784554
6. 672042e-33 -11. 947733
2. 348438e-44 -13. 970839
7. 624179e-50 -14. 843858
6. 061011e-67 -17. 285390
8. 130952e-79 -18. 796079
5. 837475e-102 -21. 438574
5. 837475e-102 -21. 438574
1. 558050e-118 -23. 147012
1. 906170e-122 -23. 532289
4. 069569e-158 -26. 791403
## Cla/Mod Mod/Cla Global
30. 1886792 64. 3356643 26. 86797443
25. 8741259 58. 2167832 28. 36676218
29. 1917973 42. 3076923 18. 27198589
23. 6280488 54. 1958042 28. 91778708
23. 6600306 54. 0209790 28. 78554111
33. 6206897 27. 2727273 10. 22702226
41. 9117647 19. 9300699 5. 99515098
22. 2946545 29. 8951049 16. 90544413
22. 2946545 29. 8951049 16. 90544413
16. 6327569 57. 1678322 43. 33259863
33. 7579618 9. 2657343 3. 46043641
18. 1355932 37. 4125874 26. 00837558
44. 8979592 3. 8461538 1. 08000882
30. 0000000 7. 3426573 3. 08573948

```

```

## engineSize=f.EngSize-1.8
## year=f. Year-2015
## model=VW-Scirocco
## model=Mercedes-V Class
## model=VW-Touareg
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05]
## model=BMW-M3
## manufacturer=Audi
## Audi=Audi Yes
## model=BMW-X3
## aux_mpg=f.mpg-[8.8, 44.8]
## model=Audi-Q5
## engineSize=f.EngSize-4.2 or more
## model=Mercedes-GLS Class
## model=Audi-SQ5
## model=VW-Shuttle
## model=Mercedes-CLS Class
## aux_price=f.price-(19.5, 26]
## model=BMW-3 Series
## aux_years_sell=f.years_sell-(6, 23]
## model=BMW-7 Series
## model=Mercedes-E Class
## model=VW-T-Cross
## model=Mercedes-A Class
## model=BMW-5 Series
## model=VW-T-Roc
## model=Mercedes-GLA Class
## model=Mercedes-GLC Class
## model=Audi-A3
## model=VW-Passat
## Audi=Audi No
## manufacturer=Mercedes
## model=Mercedes-B Class
## engineSize=f.EngSize-1.6
## model=VW-Up
## year=f. Year-2013
## engineSize=f.EngSize-1.2
## aux_price=f.price-[0.65, 14]
## engineSize=f.EngSize-1.3
## model=Mercedes-C Class
## engineSize=f.EngSize-2.1
## aux_mpg=f.mpg-(53.3, 61.4]
## aux_price=f.price-(26, 110]
## fuelType=f.Fuel-Diesel
## model=VW-Polo
## aux_tax=f.tax-[0, 125]
## engineSize=f.EngSize-1.5
## year=f. Year-2020
## engineSize=f.EngSize-1
## aux_mileage=f.mileage-(6e+03, 1.65e+04]
## aux_tax=f.tax-(125, 145]
## aux_mpg=f.mpg-(61.4, 88.3]
## aux_mileage=f.mileage-[4, 6e+03]
## year=f. Year-2019
## aux_years_sell=f.years_sell-[2, 3]
##
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04]
## aux_tax=f.tax-(145, 580]
## year=f. Year-2017
## aux_mpg=f.mpg-(44.8, 53.3]
## aux_years_sell=f.years_sell-(3, 5]
## engineSize=f.EngSize-3
## engineSize=f.EngSize-1.4
## aux_years_sell=f.years_sell-(5, 6]
61. 1111111 1. 9230769 0. 39673793
21. 0826211 12. 9370629 7. 73638968
52. 3809524 1. 9230769 0. 46286092
55. 5555556 1. 7482517 0. 39673793
41. 9354839 2. 2727273 0. 68327088
16. 1683278 25. 5244755 19. 90301962
80. 0000000 0. 6993007 0. 11020498
15. 6250000 26. 2237762 21. 15935640
15. 6250000 26. 2237762 21. 15935640
31. 5789474 2. 0979021 0. 83755786
15. 2753108 30. 0699301 24. 81816178
25. 0000000 3. 4965035 1. 76327970
57. 1428571 0. 6993007 0. 15428697
45. 4545455 0. 8741259 0. 24245096
100. 0000000 0. 3496503 0. 04408199
44. 4444444 0. 6993007 0. 19836897
33. 3333333 1. 0489510 0. 39673793
14. 4617913 30. 7692308 26. 82389244
17. 6190476 6. 4685315 4. 62860921
15. 1815182 16. 0839161 13. 35684373
36. 3636364 0. 6993007 0. 24245096
7. 6923077 2. 2727273 3. 72492837
0. 0000000 0. 0000000 0. 55102491
8. 3333333 3. 6713287 5. 55433106
4. 2253521 0. 5244755 1. 56491073
3. 1746032 0. 3496503 1. 38858276
3. 7037037 0. 5244755 1. 78532070
4. 5454545 0. 8741259 2. 42450959
6. 0439560 1. 9230769 4. 01146132
2. 5641026 0. 3496503 1. 71919771
11. 7975958 73. 7762238 78. 84064360
9. 8307816 21. 3286713 27. 35287635
0. 0000000 0. 0000000 1. 27837778
6. 2091503 3. 3216783 6. 74454485
1. 1764706 0. 1748252 1. 87348468
1. 1627907 0. 1748252 1. 89552568
2. 4793388 0. 5244755 2. 66696055
8. 9159068 15. 3846154 21. 75446330
0. 0000000 0. 0000000 1. 67511572
5. 6379822 3. 3216783 7. 42781574
4. 6703297 2. 9720280 8. 02292264
8. 0935252 15. 7342657 24. 50958783
8. 1526453 16. 4335664 25. 41326868
9. 4481937 41. 6083916 55. 52126956
2. 0710059 1. 2237762 7. 44985673
0. 0000000 0. 0000000 5. 77474102
2. 6168224 2. 4475524 11. 79193300
0. 0000000 0. 0000000 6. 63433987
0. 4987531 0. 3496503 8. 83843950
4. 7775947 10. 1398601 26. 75776945
7. 9986613 41. 7832168 65. 85849680
0. 0000000 0. 0000000 21. 75446330
0. 0000000 0. 0000000 26. 47123650
0. 0000000 0. 0000000 34. 31783117
0. 0000000 0. 0000000 40. 95217104
p. value v. test
2. 435623e-91 20. 268584
4. 327399e-58 16. 067268
1. 456410e-47 14. 487356
3. 473193e-42 13. 610404
4. 006287e-42 13. 599964
2. 048401e-36 12. 602391
2. 290340e-36 12. 593584
1. 091130e-16 8. 294424

```

```

## year=f. Year-2016
## fuelType=f. Fuel-Petrol
## model=VW-Tiguan
## aux_price=f. price-(14, 19.5]
## model=Mercedes-GLE Class
## model=Audi-Q3
## engineSize=f. EngSize-1.8
## year=f. Year-2015
## model=VW-Scirocco
## model=Mercedes-V Class
## model=VW-Touareg
## aux_mileage=f. mileage-(3.35e+04, 1.16e+05]
## model=BMW-M3
## manufacturer=Audi
## Audi=Audi Yes
## model=BMW-X3
## aux_mpg=f. mpg-[8.8, 44.8]
## model=Audi-Q5
## engineSize=f. EngSize-4.2 or more
## model=Mercedes-GLS Class
## model=Audi-SQ5
## model=VW-Shuttle
## model=Mercedes-CLS Class
## aux_price=f. price-(19.5, 26]
## model=BMW-3 Series
## aux_years_sell=f. years_sell-(6, 23]
## model=BMW-7 Series
## model=Mercedes-E Class
## model=VW-T-Cross
## model=Mercedes-A Class
## model=BMW-5 Series
## model=VW-T-Roc
## model=Mercedes-GLA Class
## model=Mercedes-GLC Class
## model=Audi-A3
## model=VW-Passat
## Audi=Audi No
## manufacturer=Mercedes
## model=Mercedes-B Class
## engineSize=f. EngSize-1.6
## model=VW-Up
## year=f. Year-2013
## engineSize=f. EngSize-1.2
## aux_price=f. price-[0.65, 14]
## engineSize=f. EngSize-1.3
## model=Mercedes-C Class
## engineSize=f. EngSize-2.1
## aux_mpg=f. mpg-(53.3, 61.4]
## aux_price=f. price-(26, 110]
## fuelType=f. Fuel-Diesel
## model=VW-Polo
## aux_tax=f. tax-[0, 125]
## engineSize=f. EngSize-1.5
## year=f. Year-2020
## engineSize=f. EngSize-1
## aux_mileage=f. mileage-(6e+03, 1.65e+04]
## aux_tax=f. tax-(125, 145]
## aux_mpg=f. mpg-(61.4, 88.3]
## aux_mileage=f. mileage-[4, 6e+03]
## year=f. Year-2019
## aux_years_sell=f. years_sell-[2, 3]
##
## $ '4'
## 
```

Cla/Mod	Mod/Cla	Global
---------	---------	--------

```

## aux_mpg=f.mpg-(61.4, 88.3]      58.4599797 60.7368421 21.75446330
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 49.7949139 63.8947368 26.86797443
## year=f.Year-2017                56.0916767 48.9473684 18.27198589
## aux_years_sell=f.years_sell-(3,5] 43.2618683 59.4736842 28.78554111
## aux_price=f.price-[0.65,14]      41.6413374 43.2631579 21.75446330
## model=VW-Polo                  52.6627219 18.7368421 7.44985673
## engineSize=f.EngSize-1.2        76.0330579 9.6842105 2.66696055
## aux_years_sell=f.years_sell-(5,6] 39.3741851 31.7894737 16.90544413
## year=f.Year-2016                39.3741851 31.7894737 16.90544413
## transmission=f.Trans-Manual    31.3149847 53.8947368 36.03702887
## aux_mpg=f.mpg-(53.3,61.4]       33.5431655 39.2631579 24.50958783
## aux_price=f.price-(14,19.5]      30.9322034 38.4210526 26.00837558
## engineSize=f.EngSize-2.1        39.8351648 15.2631579 8.02292264
## engineSize=f.EngSize-1          37.6558603 15.8947368 8.83843950
## fuelType=f.Fuel-Diesel         25.1290195 66.6315789 55.52126956
## aux_tax=f.tax-(125,145]         24.0963855 75.7894737 65.85849680
## manufacturer=VW                26.0533708 39.0526316 31.38637866
## model=VW-Up                   45.8823529 4.1052632 1.87348468
## model=Audi-A1                  36.9230769 5.0526316 2.86532951
## engineSize=f.EngSize-1.6        28.4313725 9.1578947 6.74454485
## model=Audi-A3                  30.7692308 5.8947368 4.01146132
## model=Mercedes-A Class         28.5714286 7.5789474 5.55433106
## model=BMW-1 Series             29.4444444 5.5789474 3.96737933
## model=Mercedes-E Class         29.5857988 5.2631579 3.72492837
## Audi=Audi No                  21.6102880 81.3684211 78.84064360
## model=BMW-i3                  75.0000000 0.3157895 0.08816399
## engineSize=f.EngSize-2.9        0.0000000 0.0000000 0.28653295
## model=VW-Tiguan Allspace       0.0000000 0.0000000 0.30857395
## model=VW-Amarok                0.0000000 0.0000000 0.30857395
## manufacturer=Audi              18.4375000 18.6315789 21.15935640
## Audi=Audi Yes                 18.4375000 18.6315789 21.15935640
## model=Audi-A8                  0.0000000 0.0000000 0.33061494
## model=BMW-X4                  0.0000000 0.0000000 0.35265594
## model=VW-Arteon                3.7037037 0.1052632 0.59510690
## model=BMW-4 Series             11.1111111 1.0526316 1.98368966
## engineSize=f.EngSize-1.8        0.0000000 0.0000000 0.39673793
## model=Mercedes-V Class         0.0000000 0.0000000 0.39673793
## model=Mercedes-CLS Class       0.0000000 0.0000000 0.39673793
## model=Audi-Q7                  0.0000000 0.0000000 0.39673793
## model=VW-Sharan                0.0000000 0.0000000 0.44081993
## model=Mercedes-GLC Class       10.9090909 1.2631579 2.42450959
## model=BMW-X2                  0.0000000 0.0000000 0.48490192
## model=BMW-X5                  0.0000000 0.0000000 0.52898391
## model=VW-T-Cross               0.0000000 0.0000000 0.55102491
## aux_tax=f.tax-(145,580]         17.8710179 24.2105263 28.36676218
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05] 17.0542636 16.2105263 19.90301962
## model=VW-Touareg              0.0000000 0.0000000 0.68327088
## model=BMW-X3                  0.0000000 0.0000000 0.83755786
## model=VW-T-Roc                 3.1746032 0.2105263 1.38858276
## model=BMW-2 Series             7.6923077 1.0526316 2.86532951
## model=Audi-Q2                  3.7974684 0.3157895 1.74123870
## model=Mercedes-GLE Class       0.0000000 0.0000000 1.08000882
## model=Audi-Q3                  7.1428571 1.0526316 3.08573948
## manufacturer=BMW               15.3508772 14.7368421 20.10138858
## transmission=f.Trans-Automatic 15.4392192 18.3157895 24.84020278
## aux_years_sell=f.years_sell-(6,23] 12.7062706 8.1052632 13.35684373
## engineSize=f.EngSize-1.3        0.0000000 0.0000000 1.67511572
## model=Audi-Q5                  0.0000000 0.0000000 1.76327970
## aux_mileage=f.mileage-(6e+03, 1.65e+04] 15.2388797 19.4736842 26.75776945
## model=VW-Tiguan                4.4585987 0.7368421 3.46043641
## year=f.Year-2013                0.0000000 0.0000000 1.89552568
## engineSize=f.EngSize-2          16.2317328 32.7368421 42.23054882
## year=f.Year-2014                2.4691358 0.4210526 3.57064139
## fuelType=f.Fuel-Petrol          15.4628688 32.0000000 43.33259863

```

```

## aux_price=f. price-(19. 5, 26]
## transmission=f. Trans-SemiAuto
## aux_tax=f. tax-[0, 125]
## year=f. Year-2020
## engineSize=f. EngSize-3
## aux_price=f. price-(26, 110]
## aux_mpg=f. mpg-[8. 8, 44. 8]
## aux_mileage=f. mileage-[4, 6e+03]
## aux_mpg=f. mpg-(44. 8, 53. 3]
## year=f. Year-2019
## aux_years_sell=f. years_sell-[2, 3]
## aux_mpg=f. mpg-(61. 4, 88. 3]
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]
## year=f. Year-2017
## aux_years_sell=f. years_sell-(3, 5]
## aux_price=f. price-[0. 65, 14]
## model=VW-Polo
## engineSize=f. EngSize-1. 2
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## transmission=f. Trans-Manual
## aux_mpg=f. mpg-(53. 3, 61. 4]
## aux_price=f. price-(14, 19. 5]
## engineSize=f. EngSize-2. 1
## engineSize=f. EngSize-1
## fuelType=f. Fuel-Diesel
## aux_tax=f. tax-(125, 145]
## manufacturer=VW
## model=VW-Up
## model=Audi-A1
## engineSize=f. EngSize-1. 6
## model=Audi-A3
## model=Mercedes-A Class
## model=BMW-1 Series
## model=Mercedes-E Class
## Audi=Audi No
## model=BMW-i3
## engineSize=f. EngSize-2. 9
## model=VW-Tiguan Allspace
## model=VW-Amarok
## manufacturer=Audi
## Audi=Audi Yes
## model=Audi-A8
## model=BMW-X4
## model=VW-Arteon
## model=BMW-4 Series
## engineSize=f. EngSize-1. 8
## model=Mercedes-V Class
## model=Mercedes-CLS Class
## model=Audi-Q7
## model=VW-Sharan
## model=Mercedes-GLC Class
## model=BMW-X2
## model=BMW-X5
## model=VW-T-Cross
## aux_tax=f. tax-(145, 580]
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]
## model=VW-Touareg
## model=BMW-X3
## model=VW-T-Roc
## model=BMW-2 Series
## model=Audi-Q2
## model=Mercedes-GLE Class
13. 1470830 16. 8421053 26. 82389244
14. 8732394 27. 7894737 39. 12276835
0. 0000000 0. 0000000 5. 77474102
0. 0000000 0. 0000000 6. 63433987
0. 8620690 0. 4210526 10. 22702226
1. 2142238 1. 4736842 25. 41326868
0. 0000000 0. 0000000 24. 81816178
0. 3330558 0. 4210526 26. 47123650
0. 0000000 0. 0000000 28. 91778708
0. 3853565 0. 6315789 34. 31783117
0. 3229279 0. 6315789 40. 95217104
p. value v. test
4. 614655e-204 30. 481278
3. 483871e-167 27. 558670
1. 288506e-141 25. 335803
1. 165581e-112 22. 556265
2. 108484e-65 17. 079519
3. 088334e-41 13. 449770
1. 494476e-39 13. 159817
2. 108409e-38 12. 958310
2. 108409e-38 12. 958310
7. 814524e-37 12. 678160
2. 205144e-30 11. 455567
1. 717482e-21 9. 520862
8. 819806e-18 8. 588388
7. 196956e-16 8. 067131
5. 299809e-15 7. 819584
1. 261159e-13 7. 410193
1. 635732e-08 5. 646683
2. 277319e-07 5. 175147
2. 233091e-05 4. 240212
1. 265265e-03 3. 223744
1. 470743e-03 3. 180395
3. 086880e-03 2. 958951
5. 915979e-03 2. 752402
6. 806730e-03 2. 706155
3. 058544e-02 2. 162425
3. 280477e-02 2. 134465
4. 693882e-02 -1. 986852
3. 708209e-02 -2. 084859
3. 708209e-02 -2. 084859
3. 058544e-02 -2. 162425
3. 058544e-02 -2. 162425
2. 929345e-02 -2. 179515
2. 313937e-02 -2. 271126
1. 584832e-02 -2. 412390
1. 511297e-02 -2. 429661
1. 443566e-02 -2. 446236
1. 443566e-02 -2. 446236
1. 443566e-02 -2. 446236
1. 443566e-02 -2. 446236
9. 003667e-03 -2. 611915
5. 716365e-03 -2. 763622
5. 614355e-03 -2. 769493
3. 500078e-03 -2. 920021
2. 763301e-03 -2. 992911
1. 247111e-03 -3. 227880
1. 124804e-03 -3. 257293
6. 683313e-04 -3. 402251
1. 272461e-04 -3. 831729
5. 903356e-05 -4. 016641
3. 690155e-05 -4. 126070
1. 491090e-05 -4. 330001
9. 334924e-06 -4. 432032

```

```

## model=Audi-Q3
## manufacturer=BMW
## transmission=f. Trans-Automatic
## aux_years_sell=f. years_sell-(6, 23]
## engineSize=f. EngSize-1.3
## model=Audi-Q5
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]
## model=VW-Tiguan
## year=f. Year-2013
## engineSize=f. EngSize-2
## year=f. Year-2014
## fuelType=f. Fuel-Petrol
## aux_price=f. price-(19. 5, 26]
## transmission=f. Trans-SemiAuto
## aux_tax=f. tax-[0, 125]
## year=f. Year-2020
## engineSize=f. EngSize-3
## aux_price=f. price-(26, 110]
## aux_mpg=f. mpg-[8. 8, 44. 8]
## aux_mileage=f. mileage-[4, 6e+03]
## aux_mpg=f. mpg-(44. 8, 53. 3]
## year=f. Year-2019
## aux_years_sell=f. years_sell-[2, 3]
##
## $ '5 '
##
## aux_tax=f. tax-(145, 580]
## aux_years_sell=f. years_sell-(6, 23]
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]
## engineSize=f. EngSize-3
## aux_mpg=f. mpg-[8. 8, 44. 8]
## year=f. Year-2015
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## model=BMW-X5
## aux_price=f. price-(19. 5, 26]
## aux_mpg=f. mpg-(44. 8, 53. 3]
## model=Audi-Q5
## year=f. Year-2014
## year=f. Year-2013
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]
## engineSize=f. EngSize-2
## model=BMW-6 Series
## model=Mercedes-M Class
## model=BMW-X6
## manufacturer=Audi
## Audi=Audi Yes
## model=Audi-Q3
## transmission=f. Trans-Automatic
## model=Audi-A8
## model=BMW-X4
## model=Mercedes-CLS Class
## model=Audi-Q7
## model=Mercedes-S Class
## model=BMW-Z4
## model=Mercedes-GLC Class
## model=Audi-A4
## engineSize=f. EngSize-1.2
## model=Audi-A1
## engineSize=f. EngSize-1.4
## manufacturer=Mercedes
## model=Mercedes-E Class
## aux_price=f. price-[0. 65, 14]
## Audi=Audi No
6. 862020e-06 -4. 497924
1. 919396e-06 -4. 761731
8. 433764e-08 -5. 357594
2. 059508e-08 -5. 606927
1. 485881e-08 -5. 663186
5. 699286e-09 -5. 825351
5. 265164e-09 -5. 838567
3. 154182e-09 -5. 923366
1. 351397e-09 -6. 061164
1. 753241e-11 -6. 725223
2. 614903e-12 -6. 997006
1. 125090e-15 -8. 012381
6. 033134e-16 -8. 088647
2. 775839e-16 -8. 182687
2. 289411e-28 -11. 046102
1. 206237e-32 -11. 898410
8. 602897e-44 -13. 878070
3. 219729e-113 -22. 613118
1. 641438e-135 -24. 775639
6. 059006e-137 -24. 908232
9. 160220e-163 -27. 187372
2. 242778e-187 -29. 195025
4. 818828e-240 -33. 085408

## Cla/Mod Mod/Cla Global
9. 5571096 100. 0000000 28. 36676218
9. 4059406 46. 3414634 13. 35684373
7. 0874862 52. 0325203 19. 90301962
9. 4827586 35. 7723577 10. 22702226
5. 6838366 52. 0325203 24. 81816178
9. 1168091 26. 0162602 7. 73638968
6. 2581486 39. 0243902 16. 90544413
6. 2581486 39. 0243902 16. 90544413
37. 5000000 7. 3170732 0. 52898391
4. 6836483 46. 3414634 26. 82389244
4. 4969512 47. 9674797 28. 91778708
13. 7500000 8. 9430894 1. 76327970
9. 2592593 12. 1951220 3. 57064139
11. 6279070 8. 1300813 1. 89552568
4. 1017227 40. 6504065 26. 86797443
3. 6534447 56. 9105691 42. 23054882
30. 0000000 2. 4390244 0. 22040996
50. 0000000 1. 6260163 0. 08816399
50. 0000000 1. 6260163 0. 08816399
4. 0625000 31. 7073171 21. 15935640
4. 0625000 31. 7073171 21. 15935640
7. 1428571 8. 1300813 3. 08573948
3. 9041704 35. 7723577 24. 84020278
20. 0000000 2. 4390244 0. 33061494
18. 7500000 2. 4390244 0. 35265594
16. 6666667 2. 4390244 0. 39673793
16. 6666667 2. 4390244 0. 39673793
16. 6666667 1. 6260163 0. 26449196
16. 6666667 1. 6260163 0. 26449196
0. 0000000 0. 0000000 2. 42450959
0. 0000000 0. 0000000 2. 53471457
0. 0000000 0. 0000000 2. 66696055
0. 0000000 0. 0000000 2. 86532951
0. 7352941 1. 6260163 5. 99515098
1. 7727639 17. 8861789 27. 35287635
0. 0000000 0. 0000000 3. 72492837
1. 5197568 12. 1951220 21. 75446330
2. 3483366 68. 2926829 78. 84064360

```

	p. value	v. test
## aux_price=f. price-(26, 110]	1. 5611448	14. 6341463 25. 41326868
## model=Mercedes-C Class	0. 2967359	0. 8130081 7. 42781574
## aux_tax=f. tax-[0, 125]	0. 0000000	0. 0000000 5. 77474102
## aux_years_sell=f. years_sell-(3, 5]	1. 3782542	14. 6341463 28. 78554111
## year=f. Year-2020	0. 0000000	0. 0000000 6. 63433987
## engineSize=f. EngSize-1. 6	0. 0000000	0. 0000000 6. 74454485
## model=VW-Polo	0. 0000000	0. 0000000 7. 44985673
## engineSize=f. EngSize-1	0. 0000000	0. 0000000 8. 83843950
## transmission=f. Trans-Manual	1. 2844037	17. 0731707 36. 03702887
## year=f. Year-2018	0. 0000000	0. 0000000 10. 51355521
## engineSize=f. EngSize-1. 5	0. 0000000	0. 0000000 11. 79193300
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]	0. 7413509	7. 3170732 26. 75776945
## aux_mpg=f. mpg-(61. 4, 88. 3]	0. 0000000	0. 0000000 21. 75446330
## aux_mpg=f. mpg-(53. 3, 61. 4]	0. 0000000	0. 0000000 24. 50958783
## aux_mileage=f. mileage-[4, 6e+03]	0. 0000000	0. 0000000 26. 47123650
## year=f. Year-2019	0. 0000000	0. 0000000 34. 31783117
## aux_years_sell=f. years_sell-[2, 3]	0. 0000000	0. 0000000 40. 95217104
## aux_tax=f. tax-(125, 145]	0. 0000000	0. 0000000 65. 85849680
##		
## aux_tax=f. tax-(145, 580]	6. 369576e-70	17. 676438
## aux_years_sell=f. years_sell-(6, 23]	1. 747650e-19	9. 028046
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]	9. 893635e-16	8. 028171
## engineSize=f. EngSize-3	1. 191617e-14	7. 716934
## aux_mpg=f. mpg-[8. 8, 44. 8]	5. 773887e-11	6. 549477
## year=f. Year-2015	4. 319601e-10	6. 242018
## aux_years_sell=f. years_sell-(5, 6]	3. 330832e-09	5. 914402
## year=f. Year-2016	3. 330832e-09	5. 914402
## model=BMW-X5	5. 690657e-09	5. 825604
## aux_price=f. price-(19. 5, 26]	2. 778016e-06	4. 686585
## aux_mpg=f. mpg-(44. 8, 53. 3]	6. 590133e-06	4. 506514
## model=Audi-Q5	9. 140160e-06	4. 436575
## year=f. Year-2014	3. 083755e-05	4. 167191
## year=f. Year-2013	1. 083519e-04	3. 871085
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]	8. 036222e-04	3. 351544
## engineSize=f. EngSize-2	9. 523471e-04	3. 304239
## model=BMW-6 Series	2. 125145e-03	3. 072162
## model=Mercedes-M Class	4. 297143e-03	2. 855483
## model=BMW-X6	4. 297143e-03	2. 855483
## manufacturer=Audi	5. 625656e-03	2. 768838
## Audi=Audi Yes	5. 625656e-03	2. 768838
## model=Audi-Q3	5. 644753e-03	2. 767734
## transmission=f. Trans-Automatic	6. 215898e-03	2. 736170
## model=Audi-A8	7. 527234e-03	2. 672571
## model=BMW-X4	9. 138136e-03	2. 606843
## model=Mercedes-CLS Class	1. 295435e-02	2. 485022
## model=Audi-Q7	1. 295435e-02	2. 485022
## model=Mercedes-S Class	4. 386434e-02	2. 015385
## model=BMW-Z4	4. 386434e-02	2. 015385
## model=Mercedes-GLC Class	4. 685259e-02	-1. 987630
## model=Audi-A4	4. 069296e-02	-2. 046645
## engineSize=f. EngSize-1. 2	3. 435337e-02	-2. 115899
## model=Audi-A1	2. 663509e-02	-2. 216824
## engineSize=f. EngSize-1. 4	2. 253878e-02	-2. 281163
## manufacturer=Mercedes	1. 392839e-02	-2. 459105
## model=Mercedes-E Class	8. 788103e-03	-2. 620189
## aux_price=f. price-[0. 65, 14]	6. 270384e-03	-2. 733297
## Audi=Audi No	5. 625656e-03	-2. 768838
## aux_price=f. price-(26, 110]	3. 637814e-03	-2. 907972
## model=Mercedes-C Class	8. 006980e-04	-3. 352553
## aux_tax=f. tax-[0, 125]	5. 994214e-04	-3. 431876
## aux_years_sell=f. years_sell-(3, 5]	2. 108001e-04	-3. 705709
## year=f. Year-2020	1. 909925e-04	-3. 730642
## engineSize=f. EngSize-1. 6	1. 648160e-04	-3. 767613
## model=VW-Polo	6. 389464e-05	-3. 997946

```

## engineSize=f.EngSize-1
## transmission=f.Trans-Manual
## year=f.Year-2018
## engineSize=f.EngSize-1.5
## aux_mileage=f.mileage-(6e+03, 1.65e+04]
## aux_mpg=f.mpg-(61.4, 88.3]
## aux_mpg=f.mpg-[53.3, 61.4]
## aux_mileage=f.mileage-[4, 6e+03]
## year=f.Year-2019
## aux_years_sell=f.years_sell-[2, 3]
## aux_tax=f.tax-(125, 145]
##
## $ '6 '
## aux_tax=f.tax-[0, 125]
## aux_mpg=f.mpg-[53.3, 61.4]
## aux_years_sell=f.years_sell-(5, 6]
## year=f.Year-2016
## engineSize=f.EngSize-2.1
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05]
## aux_price=f.price-(14, 19.5]
## aux_years_sell=f.years_sell-(6, 23]
## model=Mercedes-GLC Class
## model=Mercedes-GLA Class
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04]
## model=BMW-X1
## manufacturer=Mercedes
## fuelType=f.Fuel-Diesel
## year=f.Year-2015
## year=f.Year-2013
## model=Mercedes-GL Class
## year=f.Year-2014
## manufacturer=BMW
## model=VW-Touran
## aux_price=f.price-[0.65, 14]
## Audi=Audi No
## manufacturer=Audi
## Audi=Audi Yes
## model=VW-T-Roc
## engineSize=f.EngSize-1.3
## model=Audi-Q5
## model=VW-Up
## aux_years_sell=f.years_sell-(3, 5]
## engineSize=f.EngSize-1.5
## model=VW-Polo
## aux_price=f.price-(19.5, 26]
## fuelType=f.Fuel-Petrol
## model=VW-Golf
## engineSize=f.EngSize-3
## year=f.Year-2020
## manufacturer=VW
## engineSize=f.EngSize-1
## year=f.Year-2018
## aux_mileage=f.mileage-(6e+03, 1.65e+04]
## aux_price=f.price-(26, 110]
## aux_mpg=f.mpg-(61.4, 88.3]
## aux_mpg=f.mpg-[8.8, 44.8]
## aux_mileage=f.mileage-[4, 6e+03]
## aux_tax=f.tax-(145, 580]
## year=f.Year-2019
## aux_years_sell=f.years_sell-[2, 3]
## aux_tax=f.tax-(125, 145]
##
## aux_tax=f.tax-[0, 125]

9.679875e-06 -4.424203
3.084650e-06 -4.665101
9.552526e-07 -4.900639
1.583524e-07 -5.242573
3.867777e-08 -5.496785
4.916478e-14 -7.534118
5.527285e-16 -8.099308
2.041982e-17 -8.491380
1.452254e-23 -10.004772
2.226146e-29 -11.253487
1.435974e-59 -16.277063

## Cla/Mod Mod/Cla Global
100.0000000 99.6197719 5.7747410
16.54676259 69.9619772 24.5095878
16.81877445 49.0494297 16.9054441
16.81877445 49.0494297 16.9054441
21.70329670 30.0380228 8.0229226
14.17497231 48.6692015 19.9030196
11.69491525 52.4714829 26.0083756
13.20132013 30.4182510 13.3568437
24.54545455 10.2661597 2.4245096
27.16049383 8.3650190 1.7853207
9.51599672 44.1064639 26.8679744
28.57142857 7.6045627 1.5428697
8.78323932 41.4448669 27.3528764
7.26478761 69.5817490 55.5212696
11.96581197 15.9695817 7.7363897
18.60465116 6.0836502 1.8955257
41.66666667 1.9011407 0.2644920
12.96296296 7.9847909 3.5706414
7.78508772 26.9961977 20.1013886
17.64705882 2.2813688 0.7493939
7.39614995 27.7566540 21.7544633
6.17836176 84.0304183 78.8406436
4.37500000 15.9695817 21.1593564
4.37500000 15.9695817 21.1593564
0.00000000 0.0000000 1.3885828
0.00000000 0.0000000 1.6751157
0.00000000 0.0000000 1.7632797
0.00000000 0.0000000 1.8734847
4.13476263 20.5323194 28.7855411
2.99065421 6.0836502 11.7919330
2.07100592 2.6615970 7.4498567
3.86195563 17.8707224 26.8238924
4.01831129 30.0380228 43.3325986
1.53508772 2.6615970 10.0506943
1.29310345 2.2813688 10.2270223
0.00000000 0.0000000 6.6343399
2.87921348 15.5893536 31.3863787
0.00000000 0.0000000 8.8384395
0.00000000 0.0000000 10.5135552
1.56507414 7.2243346 26.7577695
0.43365134 1.9011407 25.4132687
0.00000000 0.0000000 21.7544633
0.08880995 0.3802281 24.8181618
0.00000000 0.0000000 26.4712365
0.00000000 0.0000000 28.3667622
0.00000000 0.0000000 34.3178312
0.00000000 0.0000000 40.9521710
0.03346720 0.3802281 65.8584968

## p.value v.test
0.000000e+00 Inf

```

```

## aux_mpg=f.mpg-(53.3, 61.4] 4. 289299e-58 16. 067817
## aux_years_sell=f.years_sell-(5,6] 1. 006143e-35 12. 476250
## year=f.Year-2016 1. 006143e-35 12. 476250
## engineSize=f.EngSize-2.1 1. 130251e-27 10. 901779
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05] 2. 738265e-27 10. 820982
## aux_price=f.price-(14, 19.5] 4. 361829e-21 9. 423527
## aux_years_sell=f.years_sell-(6,23] 9. 291914e-14 7. 450597
## model=Mercedes-GLC Class 6. 260056e-11 6. 537393
## model=Mercedes-GLA Class 5. 039742e-10 6. 217862
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 5. 874826e-10 6. 193753
## model=BMW-X1 1. 202966e-09 6. 079848
## manufacturer=Mercedes 3. 857900e-07 5. 075839
## fuelType=f.Fuel-Diesel 1. 633256e-06 4. 794200
## year=f.Year-2015 4. 209724e-06 4. 600750
## year=f.Year-2013 3. 116050e-05 4. 164815
## model=Mercedes-GL Class 3. 809536e-04 3. 552938
## year=f.Year-2014 4. 991480e-04 3. 481213
## manufacturer=BMW 5. 426874e-03 2. 780539
## model=VW-Touran 1. 502677e-02 2. 431733
## aux_price=f.price-[0.65, 14] 1. 780602e-02 2. 369627
## Audi=Audi No 3. 006261e-02 2. 169264
## manufacturer=Audi 3. 006261e-02 -2. 169264
## Audi=Audi Yes 3. 006261e-02 -2. 169264
## model=VW-T-Roc 2. 262183e-02 -2. 279762
## engineSize=f.EngSize-1.3 1. 028050e-02 -2. 566249
## model=Audi-Q5 8. 061550e-03 -2. 649481
## model=VW-Up 5. 946842e-03 -2. 750698
## aux_years_sell=f.years_sell-(3,5] 1. 779870e-03 -3. 124699
## engineSize=f.EngSize-1.5 1. 562562e-03 -3. 162807
## model=VW-Polo 7. 284769e-04 -3. 378631
## aux_price=f.price-(19.5, 26] 4. 797187e-04 -3. 491833
## fuelType=f.Fuel-Petrol 5. 383407e-06 -4. 549263
## model=VW-Golf 2. 706381e-06 -4. 691931
## engineSize=f.EngSize-3 3. 833616e-07 -5. 077039
## year=f.Year-2020 8. 223950e-09 -5. 763803
## manufacturer=VW 1. 832350e-09 -6. 012012
## engineSize=f.EngSize-1 1. 250816e-11 -6. 774217
## year=f.Year-2018 8. 092862e-14 -7. 468801
## aux_mileage=f.mileage-(6e+03, 1.65e+04] 1. 700696e-16 -8. 241497
## aux_price=f.price-(26, 110] 1. 636107e-27 -10. 868078
## aux_mpg=f.mpg-(61.4, 88.3] 1. 048457e-29 -11. 319687
## aux_mpg=f.mpg-[8.8, 44.8] 1. 819404e-32 -11. 864057
## aux_mileage=f.mileage-[4, 6e+03] 4. 304970e-37 -12. 724814
## aux_tax=f.tax-(145, 580] 3. 353107e-40 -13. 272261
## year=f.Year-2019 1. 502424e-50 -14. 952397
## aux_years_sell=f.years_sell-[2,3] 2. 588001e-63 -16. 796517
## aux_tax=f.tax-(125, 145] 1. 403298e-127 -24. 028430
##
## $ '7 '
##
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05] 45. 5149502 95. 8041958 19. 90301962
## aux_years_sell=f.years_sell-(6,23] 49. 5049505 69. 9300699 13. 35684373
## aux_price=f.price-[0.65, 14] 29. 8885512 68. 7645688 21. 75446330
## aux_tax=f.tax-(145, 580] 25. 5633256 76. 6899767 28. 36676218
## year=f.Year-2014 64. 8148148 24. 4755245 3. 57064139
## aux_mpg=f.mpg-(61.4, 88.3] 24. 7213779 56. 8764569 21. 75446330
## year=f.Year-2015 37. 0370370 30. 3030303 7. 73638968
## year=f.Year-2013 68. 6046512 13. 7529138 1. 89552568
## fuelType=f.Fuel-Diesel 13. 4974196 79. 2540793 55. 52126956
## transmission=f.Trans-Manual 14. 6177370 55. 7109557 36. 03702887
## aux_years_sell=f.years_sell-(5,6] 15. 2542373 27. 2727273 16. 90544413
## year=f.Year-2016 15. 2542373 27. 2727273 16. 90544413
## engineSize=f.EngSize-1.6 18. 9542484 13. 5198135 6. 74454485
## engineSize=f.EngSize-2.1 17. 8571429 15. 1515152 8. 02292264

```

```

## year=f. Year-2012
## model=VW-Passat
## model=Audi-A3
## model=Mercedes-SLK
## model=BMW-3 Series
## engineSize=f. EngSize-1.4
## model=BMW-1 Series
## model=VW-Touareg
## model=BMW-2 Series
## manufacturer=Mercedes
## model=Audi-Q5
## model=Mercedes-GLE Class
## model=Audi-Q2
## model=VW-T-Roc
## engineSize=f. EngSize-1.3
## engineSize=f. EngSize-1
## model=Mercedes-GLC Class
## engineSize=f. EngSize-3
## engineSize=f. EngSize-1.5
## aux_tax=f. tax-[0, 125]
## year=f. Year-2020
## aux_mpg=f. mpg-(44.8, 53.3]
## transmission=f. Trans-SemiAuto
## year=f. Year-2018
## year=f. Year-2017
## fuelType=f. Fuel-Petrol
## aux_mileage=f. mileage-(1.65e+04, 3.35e+04]
## aux_mpg=f. mpg-[8.8, 44.8]
## aux_years_sell=f. years_sell-(3, 5]
## aux_price=f. price-(19.5, 26]
## aux_price=f. price-(26, 110]
## aux_mileage=f. mileage-[4, 6e+03]
## aux_mileage=f. mileage-(6e+03, 1.65e+04]
## aux_tax=f. tax-(125, 145]
## year=f. Year-2019
## aux_years_sell=f. years_sell-[2, 3]
## aux_mileage=f. mileage-(3.35e+04, 1.16e+05]
## aux_years_sell=f. years_sell-(6, 23]
## aux_price=f. price-[0.65, 14]
## aux_tax=f. tax-(145, 580]
## year=f. Year-2014
## aux_mpg=f. mpg-(61.4, 88.3]
## year=f. Year-2015
## year=f. Year-2013
## fuelType=f. Fuel-Diesel
## transmission=f. Trans-Manual
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## engineSize=f. EngSize-1.6
## engineSize=f. EngSize-2.1
## year=f. Year-2012
## model=VW-Passat
## model=Audi-A3
## model=Mercedes-SLK
## model=BMW-3 Series
## engineSize=f. EngSize-1.4
## model=BMW-1 Series
## model=VW-Touareg
## model=BMW-2 Series
## manufacturer=Mercedes
## model=Audi-Q5
## model=Mercedes-GLE Class
## model=Audi-Q2
85.7142857 1.3986014 0.15428697
25.6410256 4.6620047 1.71919771
17.5824176 7.4592075 4.01146132
100.0000000 0.6993007 0.06612299
15.7142857 7.6923077 4.62860921
14.3382353 9.0909091 5.99515098
15.0000000 6.2937063 3.96737933
0.0000000 0.0000000 0.68327088
4.6153846 1.3986014 2.86532951
7.8968574 22.8438228 27.35287635
2.5000000 0.4662005 1.76327970
0.0000000 0.0000000 1.08000882
1.2658228 0.2331002 1.74123870
0.0000000 0.0000000 1.38858276
0.0000000 0.0000000 1.67511572
4.4887781 4.1958042 8.83843950
0.0000000 0.0000000 2.42450959
4.3103448 4.6620047 10.22702226
3.7383178 4.6620047 11.79193300
0.0000000 0.0000000 5.77474102
0.0000000 0.0000000 6.63433987
4.5731707 13.9860140 28.91778708
5.4084507 22.3776224 39.12276835
0.0000000 0.0000000 10.51355521
1.4475271 2.7972028 18.27198589
4.1709054 19.1142191 43.33259863
1.4766202 4.1958042 26.86797443
0.6216696 1.6317016 24.81816178
0.9188361 2.7972028 28.78554111
0.4930156 1.3986014 26.82389244
0.0000000 0.0000000 25.41326868
0.0000000 0.0000000 26.47123650
0.0000000 0.0000000 26.75776945
3.3467202 23.3100233 65.85849680
0.0000000 0.0000000 34.31783117
0.0000000 0.0000000 40.95217104
p.value v.test
9.891670e-299 36.941721
1.190092e-189 29.373713
2.196663e-109 22.220082
6.461240e-107 21.963313
3.367288e-71 17.841469
2.802331e-63 16.791796
2.895136e-50 14.908658
1.379575e-41 13.509230
4.718653e-27 10.770997
2.843931e-18 8.717507
1.360434e-08 5.678296
1.360434e-08 5.678296
1.152278e-07 5.300909
1.910522e-07 5.207839
4.523182e-06 4.585769
2.947729e-05 4.177467
4.960550e-04 3.482878
8.400591e-04 3.339247
3.183018e-03 2.949487
7.344760e-03 2.680795
1.501424e-02 2.432036
4.550310e-02 -1.999974
4.308395e-02 -2.022895
2.589983e-02 -2.227710
1.800941e-02 -2.365425
7.488184e-03 -2.674316
3.787305e-03 -2.895355

```

```

## model=VW-T-Roc
## engineSize=f. EngSize-1.3
## engineSize=f. EngSize-1
## model=Mercedes-GLC Class
## engineSize=f. EngSize-3
## engineSize=f. EngSize-1.5
## aux_tax=f. tax-[0, 125]
## year=f. Year-2020
## aux_mpg=f. mpg-(44.8, 53.3]
## transmission=f. Trans-SemiAuto
## year=f. Year-2018
## year=f. Year-2017
## fuelType=f. Fuel-Petrol
## aux_mileage=f. mileage-(1.65e+04, 3.35e+04]
## aux_mpg=f. mpg-[8.8, 44.8]
## aux_years_sell=f. years_sell-(3, 5]
## aux_price=f. price-(19.5, 26]
## aux_price=f. price-(26, 110]
## aux_mileage=f. mileage-[4, 6e+03]
## aux_mileage=f. mileage-(6e+03, 1.65e+04]
## aux_tax=f. tax-(125, 145]
## year=f. Year-2019
## aux_years_sell=f. years_sell-[2, 3]
1. 830504e-03 -3. 116438
4. 927869e-04 -3. 484647
1. 171853e-04 -3. 851941
1. 562498e-05 -4. 319688
1. 391587e-05 -4. 345184
1. 132372e-07 -5. 304089
2. 195269e-12 -7. 021482
3. 482496e-14 -7. 578992
2. 626474e-14 -7. 615512
1. 180600e-14 -7. 718118
1. 573773e-22 -9. 766111
1. 054585e-24 -10. 261136
1. 457598e-28 -11. 086577
7. 890038e-38 -12. 856673
3. 074791e-45 -14. 114894
6. 145270e-49 -14. 703235
1. 044903e-51 -15. 128860
1. 369910e-58 -16. 138426
1. 947890e-61 -16. 538190
3. 242714e-62 -16. 645860
1. 866950e-80 -18. 995250
4. 922035e-84 -19. 423107
1. 508111e-105 -21. 819710

```

We start with the description of the categorical variables that characterize the clusters:

Cluster 1: Cluster one has cars with few years\_sell, low mpg, medium taxes, low mileage and high price. Also, cars with semiAuto transmission and cars with petrol as fuel predominate. We can say this because these categories are overrepresented in the cluster.

Cluster 2: Cluster two has cars with few years\_sell, low mpg, medium taxes and high price. Also, cars with diesel as fuel and non-Audi cars predominate.

Cluster 3: Cluster three has cars with high taxes, few years\_sell, low mpg, high mileage and medium price. Also, cars with petrol as fuel predominate.

Cluster 4: Cluster four has cars with high mpg, high mileage, few years\_sell, low price and medium taxes. Also, cars with manual transmission and cars with diesel as fuel predominate.

Cluster 5: Cluster five has cars high taxes, high years\_sell, high mileage, low mpg and medium-high price.

Cluster 6: Cluster six has cars with low taxes, high mpg, high years\_sell, high mileage and low price. Also, non-Audi cars predominate.

Cluster 7: Cluster seven has cars with high mileage, high years\_sell, low price, high taxes and high mpg. Also, cars with diesel as fuel and cars with manual transmission predominate.

Cluster 8: Cluster of multivariate outliers.

```
res.hcpc$desc.var$quanti.var
```

```

##          Eta2 P-value
## price     0.4588663 0
## mileage   0.7666377 0
## tax       0.9299117 0
## mpg       0.7156411 0
## years_sell 0.7859983 0

```

```
res.hcpc$desc.var$quanti
```

```

## $ '1 '
##          v. test Mean in category Overall mean sd in category Overall sd
## price     36.976855 30102.827803 21432.746088 9015.830564 9014.199606
## tax      -2.662444 145.628274 146.384170 1.792052 10.914805
## mileage   -32.039923 5812.507623 19240.275517 4656.415718 16111.883519
## years_sell -35.496314 2.927354 4.523914 0.526530 1.729163

```

```

## mpg      -45.379436      39.915067      52.881019      5.380837      10.984473
##          p.value
## price     2.697200e-299
## tax       7.757544e-03
## mileage   3.033028e-225##
## years_sell 5.603063e-276##
## mpg      0.000000e+00##
## $ '2 '
##          v. test Mean in category Overall mean sd in category Overall sd
## price     8.589204    23483.266359    21432.746088    6917.8501498 9014.199606
## mpg       7.462959    55.052091     52.881019     6.0517425 10.984473
## tax       -3.231640    145.450007    146.384170     1.9954547 10.914805
## mileage   -29.232483   6766.554839   19240.275517   5011.2700906 16111.883519
## years_sell -29.935866   3.152995     4.523914     0.6240959 1.729163
##          p.value
## price     8.757425e-18
## mpg       8.460078e-14
## tax       1.230818e-03
## mileage   7.498773e-188##
## years_sell 6.721228e-197##
## $ '3 '
##          v. test Mean in category Overall mean sd in category Overall sd
## years_sell 14.983046    5.536713     4.523914     0.9700308 1.729163
## mileage    12.801364   27303.145105   19240.275517   9058.2672706 16111.883519
## tax       7.126435    149.424883    146.384170     5.3695727 10.914805
## price     -3.052076   20357.248252   21432.746088   6431.1260475 9014.199606
## mpg       -12.187832   47.647512     52.881019     6.2757653 10.984473
##          p.value
## years_sell 9.477153e-51
## mileage   1.610955e-37##
## tax       1.030021e-12
## price     2.272647e-03
## mpg       3.608931e-34##
## $ '4 '
##          v. test Mean in category Overall mean sd in category Overall sd
## mpg       37.392552    64.731383    52.881019     5.7548139 10.984473
## years_sell 16.885568    5.366316     4.523914     0.8120942 1.729163
## mileage   11.635725   24649.152848   19240.275517   8540.2117516 16111.883519
## tax       -5.614238    144.616203    146.384170     2.6134466 10.914805
## price     -24.094376   15166.460000   21432.746088   4574.8333275 9014.199606
##          p.value
## mpg       5.163404e-306
## years_sell 5.746074e-64
## mileage   2.712746e-31##
## tax       1.974307e-08
## price     2.863163e-128##
## $ '5 '
##          v. test Mean in category Overall mean sd in category Overall sd
## tax       56.608581    201.341463   146.384170     2.215354 10.914805
## years_sell 13.508968    6.601626     4.523914     1.124305 1.729163
## mileage   11.129509   35189.861789   19240.275517   12894.252868 16111.883519
## mpg       -8.894295    44.191057    52.881019     3.462083 10.984473
##          p.value
## tax       0.000000e+00
## years_sell 1.384490e-41
## mileage   9.013013e-29
## mpg       5.879132e-19
## 
## $ '6 '
##          v. test Mean in category Overall mean sd in category Overall sd

```

```

## years_sell 17.274191      6.311787      4.523914      1.093419e+00      1.729163
## mileage     16.556216    35206.794677  19240.275517      1.360068e+04  16111.883519
## mpg         5.038208      56.193536      52.881019      3.281298e+00      10.984473
## price       -9.151840    16494.889734  21432.746088      4.336134e+03   9014.199606
## tax         -32.732077    125.000000    146.384170      8.720414e-01      10.914805
##                               p.value
## years_sell 7.359876e-67
## mileage    1.443955e-61##
## mpg        4.699110e-07
## price      5.597194e-20
## tax        5.462959e-235##
## $ '7'
##                               v. test Mean in category Overall mean sd in category Overall sd
## mileage    39.657865      48598.22335  19240.275517      9424.304078  16111.883519
## years_sell 34.099871      7.23310      4.523914      1.118493      1.729163
## mpg        18.635327      62.28618      52.881019      7.687089      10.984473
## tax         3.072233      147.92488      146.384170      3.800915      10.914805
## price      -22.229546    12225.96503  21432.746088      3378.880783   9014.199606
##                               p.value
## mileage    0.000000e+00##
## years_sell 7.408703e-255##
## mpg        1.661350e-77
## tax        2.124637e-03
## price      1.779241e-109

```

Regarding quantitative variables:

Cluster 1: As we can see price mean is greater in cluster 1 than in the overall mean. Also, we can see that tax, mileage, years\_sell and mpg means are lower in cluster 1 than in the overall means. So we can say that cluster 1 is a cluster of expensive cars with low tax, low mileage, few years\_sell and low mpg.

Cluster 2: As we can see price and mpg means are greater in cluster 2 than in the overall means. Also, we can see that tax, mileage and years\_sell means are lower in cluster 2 than in the overall means. So we can say that cluster 2 is a cluster of expensive cars with high mpg, low tax, low mileage and few years\_sell.

Cluster 3: As we can see years\_sell, mileage and tax means are greater in cluster 3 than in the overall means. Also, we can see that price and mpg means are lower in cluster 3 than in the overall means. So we can say that cluster 3 is a cluster of cheap cars with high years\_sell, high mileage, high tax and low mpg.

Cluster 4: As we can see mpg, years\_sell and mileage means are greater in cluster 4 than in the overall means. Also, we can see that tax and price means are lower in cluster 4 than in the overall means. So we can say that cluster 4 is a cluster of cheap cars with high mpg, many years\_sell, high mileage and low tax.

Cluster 5: As we can see tax, years\_sell and mileage means are greater in cluster 5 than in the overall means. Also, we can see that mpg mean is lower in cluster 5 than in the overall mean. So we can say that cluster 5 is a cluster of cars with high tax, many years\_sell, high mileage and low mpg.

Cluster 6: As we can see years\_sell, mileage and mpg means are greater in cluster 6 than in the overall means. Also, we can see that price and tax means are lower in cluster 6 than in the overall means. So we can say that cluster 6 is a cluster of cheap cars with many years\_sell, high mileage, high mpg and low tax.

Cluster 7: As we can see mileage, years\_sell, mpg and tax means are greater in cluster 7 than in the overall means. Also, we can see that price mean is lower in cluster 7 than in the overall mean. So we can say that cluster 7 is a cluster of cheap cars with high mileage, many years\_sell, high mpg and high tax.

Cluster 8: Cluster of multivariate outliers.

We have created a new variable indicating the cluster to which the individual belongs after the Hierarchical clustering.

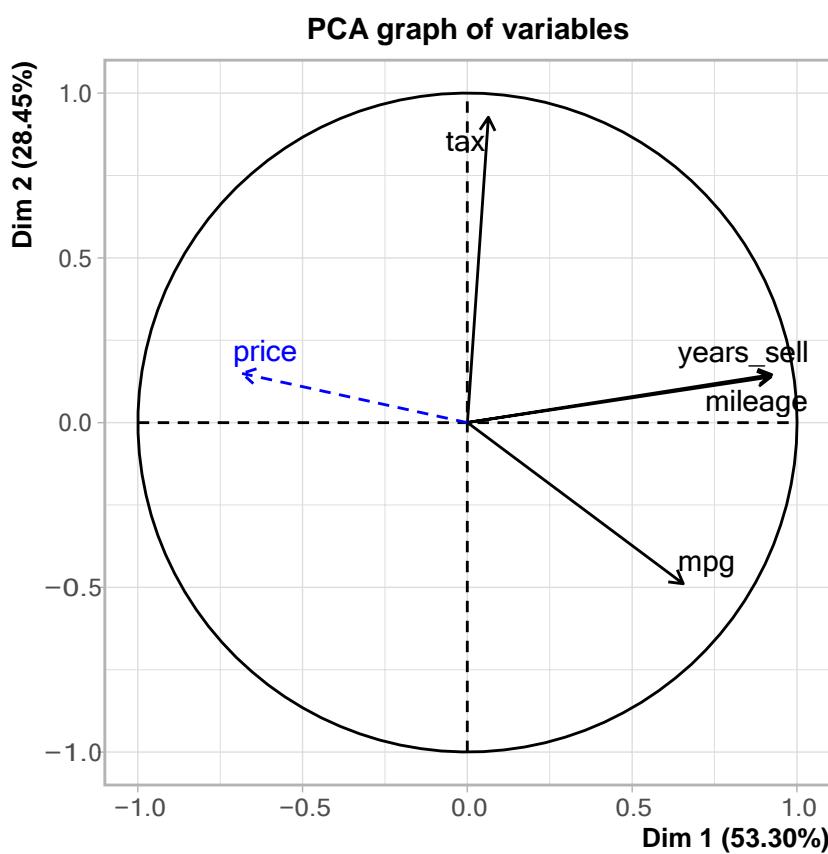
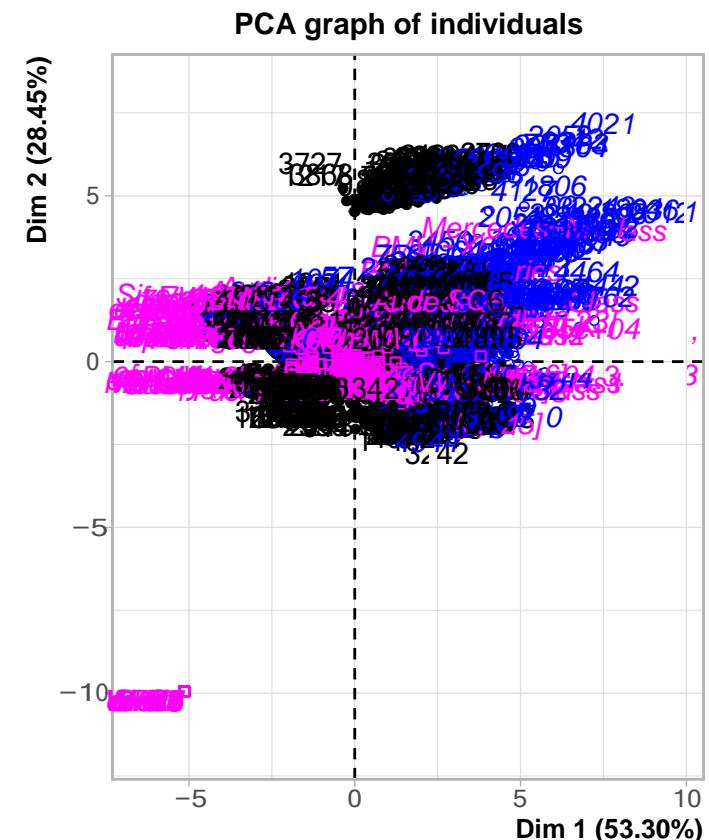
```

df$claHC<-8
df[ row.names(res.hcpc$data.clust), "claHC" ]<-res.hcpc$data.clust$clust
df$claHC<-factor(df$claHC)
levels( df$claHC ) <- paste0("claHCPC-", levels( df$claHC ) )

```

## 4 K-Means Classification

```
res.pca<-PCA(df[,c(vars_res, vars_dis, vars_con)], quali.sup=c(2:13), quanti.sup= c(1), ind.sup = 11, ncp =
```



```
ppcc<-res.pca$ind$coord[, 1:2]  
dim(ppcc)
```

```

## [1] 4537    2

dist<-dist(ppcc)
kc<-kmeans(dist, 7, iter.max = 30, trace=T)

## KMNS(*, k=7): iter= 1, indx=5
## QTRAN(): istep=4537, icoun=3##
QTRAN(): istep=9074, icoun=23##
QTRAN(): istep=13611, icoun=9
## QTRAN(): istep=18148, icoun=486##
QTRAN(): istep=22685, icoun=652##
QTRAN(): istep=27222, icoun=23 ##
QTRAN(): istep=31759, icoun=22 ##
QTRAN(): istep=36296, icoun=297##
QTRAN(): istep=40833, icoun=1209##
QTRAN(): istep=45370, icoun=2236##
KMNS(*, k=7): iter= 2, indx=3
## QTRAN(): istep=4537, icoun=175
## QTRAN(): istep=9074, icoun=361
## QTRAN(): istep=13611, icoun=320##
QTRAN(): istep=18148, icoun=2661##
QTRAN(): istep=22685, icoun=3244##
KMNS(*, k=7): iter= 3, indx=4537

```

We have created a new variable indicating the cluster to which the individual belongs after the K-Means classification.

```

df$claKM<-8
df[ -11, "claKM" ]<-kc$cluster
df$claKM<-factor(df$claKM)

100*(kc$betweenss/kc$totss)

```

```

## [1] 90.47151

```

We get an explicability of the 90.47%.

```
catdes(df, 21)
```

```

##
## Link between the cluster variable and the categorical variables (chi-square test)##
=====
##          p.value  df
## year      0.000000e+00 77
## aux_price 0.000000e+00 21##
aux_mileage 0.000000e+00 21 ##
aux_tax     0.000000e+00 14
## aux_mpg    0.000000e+00 21##
aux_years_sell 0.000000e+00 21 ##
mout       0.000000e+00 7
## clahC     0.000000e+00 49
## engineSize 1.482696e-310 147##
model      9.046692e-228 609
## fuelType   5.100134e-93 21##
transmission 3.798966e-71 14 ##
manufacturer 1.511546e-32 21 ##
Audi       6.895314e-13 7
##
## Description of each cluster by the categories
## =====
## $ '1 '

```

	Cla/Mod	Mod/Cla	Global
## aux_years_sell=f.years_sell-(3, 5]	38.5507246	73.2782369	27.60
## year=f.Year-2018	58.8235294	39.9449036	9.86
## claHC=claHCPC-2	34.2857143	51.2396694	21.70
## aux_mileage=f.mileage-(6e+03, 1.65e+04]	32.0512821	55.0964187	24.96
## claHC=claHCPC-3	44.0559441	34.7107438	11.44
## mout=NoMOut	16.0017633	100.0000000	90.74
## year=f.Year-2017	27.2829763	33.3333333	17.74
## aux_price=f.price-(19.5, 26]	23.6144578	40.4958678	24.90
## aux_tax=f.tax-(125, 145]	17.5863182	75.0688705	61.98
## manufacturer=Mercedes	20.2812731	37.7410468	27.02
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04]	20.3200000	34.9862259	25.00
## engineSize=f.EngSize-1.4	28.3737024	11.2947658	5.78
## aux_mpg=f.mpg-(53.3, 61.4]	19.1919192	31.4049587	23.76
## model=Mercedes-A Class	23.7037037	8.8154270	5.40
## model=Mercedes-E Class	25.5555556	6.3360882	3.60
## transmission=f.Trans-SemiAuto	16.8076110	43.8016529	37.84
## model=Mercedes-V Class	50.0000000	1.2396694	0.36
## fuelType=f.Fuel-Petrol	16.4514586	47.3829201	41.82
## model=Mercedes-C Class	20.4359673	10.3305785	7.34
## Audi=Audi No	15.2490421	82.2314050	78.30
## model=Mercedes-GLA Class	26.1904762	3.0303030	1.68
## model=VW-Shuttle	55.5555556	0.6887052	0.18
## aux_mpg=f.mpg-(44.8, 53.3]	16.7634253	31.8181818	27.56
## model=VW-Tiguan	21.3414634	4.8209366	3.28
## model=VW-Scirocco	33.3333333	1.1019284	0.48
## model=Audi-Q2	23.7500000	2.6170799	1.60
## aux_price=f.price-(14, 19.5]	16.2921348	27.9614325	24.92
## model=BMW-M3	50.0000000	0.4132231	0.12
## model=BMW-3 Series	10.2880658	3.4435262	4.86
## manufacturer=VW	13.0033003	27.1349862	30.30
## year=f.Year-2011	0.0000000	0.0000000	0.42
## year=f.Year-2010	0.0000000	0.0000000	0.42
## engineSize=f.EngSize-1.6	10.7648725	5.2341598	7.06
## model=VW-Sharan	0.0000000	0.0000000	0.44
## engineSize=f.EngSize-1.2	8.0000000	1.3774105	2.50
## model=Audi-A6	5.6338028	0.5509642	1.42
## model=VW-T-Cross	0.0000000	0.0000000	0.50
## model=BMW-X1	5.4054054	0.5509642	1.48
## model=Audi-A3	8.5427136	2.3415978	3.98
## manufacturer=BMW	12.0114395	17.3553719	20.98
## transmission=f.Trans-Automatic	12.3381569	22.3140496	26.26
## model=Audi-Q7	0.0000000	0.0000000	0.64
## fuelType=f.Fuel-Diesel	13.3004926	52.0661157	56.84
## Audi=Audi Yes	11.8894009	17.7685950	21.70
## manufacturer=Audi	11.8894009	17.7685950	21.70
## year=f.Year-2009 or before	0.0000000	0.0000000	0.84
## aux_price=f.price-(26, 110]	11.6800000	20.1101928	25.00
## model=VW-Golf	9.6311475	6.4738292	9.76
## model=VW-Passat	3.1914894	0.4132231	1.88
## claHC=claHCPC-4	9.6842105	12.6721763	19.00
## aux_tax=f.tax-(145, 580]	11.0349127	24.3801653	32.08
## year=f.Year-2013	0.0000000	0.0000000	2.42
## claHC=claHCPC-5	0.0000000	0.0000000	2.46
## year=f.Year-2019	10.2692549	22.5895317	31.94
## claHC=claHCPC-6	1.5209125	0.5509642	5.26
## year=f.Year-2014	0.0000000	0.0000000	3.94
## aux_tax=f.tax-[0, 125]	1.3468013	0.5509642	5.94
## aux_mpg=f.mpg-[8.8, 44.8]	7.3846154	13.2231405	26.00
## aux_years_sell=f.years_sell-[2, 3]	8.5505735	22.5895317	38.36
## aux_price=f.price-[0.65, 14]	6.5925338	11.4325069	25.18
## year=f.Year-2015	1.1933174	0.6887052	8.38
## year=f.Year-2020	0.0000000	0.0000000	6.42
## claHC=claHCPC-7	0.0000000	0.0000000	8.58

	p. value	v. test
## aux_mileage=f.mileage-[4, 6e+03]	5. 0319489	8. 6776860 25. 04
## claHC=claHCPC-8	0. 0000000	0. 0000000 9. 26
## mout=YesMOut	0. 0000000	0. 0000000 9. 26
## aux_years_sell=f.years_sell-(5, 6]	2. 9002320	3. 4435262 17. 24
## year=f.Year-2016	2. 9002320	3. 4435262 17. 24
## aux_years_sell=f.years_sell-(6, 23]	0. 5952381	0. 6887052 16. 80
## claHC=claHCPC-1	0. 5381166	0. 8264463 22. 30
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05]	0. 7200000	1. 2396694 25. 00
##		
## aux_years_sell=f.years_sell-(3, 5]	1. 385074e-173	28. 087687
## year=f.Year-2018	2. 093408e-134	24. 672838
## claHC=claHCPC-2	1. 227258e-82	19. 257250
## aux_mileage=f.mileage-(6e+03, 1. 65e+04]	1. 631200e-80	19. 002336
## claHC=claHCPC-3	8. 981878e-77	18. 544809
## mout=NoMOut	5. 767959e-34	12. 149554
## year=f.Year-2017	1. 309992e-28	11. 096126
## aux_price=f.price-(19. 5, 26]	8. 329662e-24	10. 059647
## aux_tax=f.tax-(125, 145]	8. 272241e-16	8. 050106
## manufacturer=Mercedes	8. 019335e-12	6. 838202
## aux_mileage=f.mileage-(1. 65e+04, 3. 35e+04]	7. 129743e-11	6. 517902
## engineSize=f.EngSize-1. 4	3. 393863e-10	6. 279627
## aux_mpg=f.mpg-(53. 3, 61. 4]	3. 423681e-07	5. 098493
## model=Mercedes-A Class	3. 748128e-05	4. 122481
## model=Mercedes-E Class	7. 601861e-05	3. 956617
## transmission=f.Trans-SemiAuto	3. 774053e-04	3. 555399
## model=Mercedes-V Class	4. 441305e-04	3. 512369
## fuelType=f.Fuel-Petrol	1. 069829e-03	3. 271488
## model=Mercedes-C Class	1. 369050e-03	3. 201098
## Audi=Audi No	4. 734126e-03	2. 824589
## model=Mercedes-GLA Class	4. 835981e-03	2. 817761
## model=VW-Shuttle	5. 349599e-03	2. 785192
## aux_mpg=f.mpg-(44. 8, 53. 3]	6. 041600e-03	2. 745515
## model=VW-Tiguan	1. 640450e-02	2. 399789
## model=VW-Scirocco	2. 089227e-02	2. 309926
## model=Audi-Q2	2. 697430e-02	2. 211889
## aux_price=f.price-(14, 19. 5]	4. 228892e-02	2. 030666
## model=BMW-M3	4. 853359e-02	1. 972666
## model=BMW-3 Series	4. 848498e-02	-1. 973093
## manufacturer=VW	4. 354934e-02	-2. 018403
## year=f.Year-2011	3. 681690e-02	-2. 087789
## year=f.Year-2010	3. 681690e-02	-2. 087789
## engineSize=f.EngSize-1. 6	3. 310545e-02	-2. 130802
## model=VW-Sharan	3. 144854e-02	-2. 151351
## engineSize=f.EngSize-1. 2	2. 781294e-02	-2. 199915
## model=Audi-A6	2. 156906e-02	-2. 297872
## model=VW-T-Cross	1. 959615e-02	-2. 333992
## model=BMW-X1	1. 547930e-02	-2. 420967
## model=Audi-A3	1. 029640e-02	-2. 565713
## manufacturer=BMW	8. 445558e-03	-2. 633718
## transmission=f.Trans-Automatic	8. 213313e-03	-2. 643173
## model=Audi-Q7	6. 491018e-03	-2. 721888
## fuelType=f.Fuel-Diesel	5. 120986e-03	-2. 799323
## Audi=Audi Yes	4. 734126e-03	-2. 824589
## manufacturer=Audi	4. 734126e-03	-2. 824589
## year=f.Year-2009 or before	1. 335148e-03	-3. 208316
## aux_price=f.price-(26, 110]	8. 147130e-04	-3. 347747
## model=VW-Golf	7. 700216e-04	-3. 363352
## model=VW-Passat	3. 374736e-04	-3. 584694
## claHC=claHCPC-4	1. 036027e-06	-4. 884669
## aux_tax=f.tax-(145, 580]	9. 658149e-07	-4. 898479
## year=f.Year-2013	4. 431858e-09	-5. 867211
## claHC=claHCPC-5	3. 210942e-09	-5. 920433
## year=f.Year-2019	2. 102194e-09	-5. 989707
## claHC=claHCPC-6	8. 715465e-14	-7. 459041

```

## year=f. Year-2014
## aux_tax=f. tax-[0, 125]
## aux_mpg=f. mpg-[8. 8, 44. 8]
## aux_years_sell=f. years_sell-[2, 3]
## aux_price=f. price-[0. 65, 14]
## year=f. Year-2015
## year=f. Year-2020
## claHC=claHCPC-7
## aux_mileage=f. mileage-[4, 6e+03]
## claHC=claHCPC-8
## mout=YesMOut
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## aux_years_sell=f. years_sell-(6, 23]
## claHC=claHCPC-1
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05] 2. 563329e-83 -19. 338186
##
## $ '2 '
##
## claHC=claHCPC-1
## aux_mpg=f. mpg-[8. 8, 44. 8]
## aux_years_sell=f. years_sell-[2, 3]
## aux_mileage=f. mileage-[4, 6e+03]
## aux_price=f. price-(26, 110]
## year=f. Year-2020
## year=f. Year-2019
## fuelType=f. Fuel-Petrol
## aux_tax=f. tax-(125, 145]
## mout=NoMOut
## transmission=f. Trans-SemiAuto
## engineSize=f. EngSize-3
## model=Audi-Q5
## Audi=Audi Yes
## manufacturer=Audi
## engineSize=f. EngSize-2
## model=VW-Tiguan Allspace
## engineSize=f. EngSize-1. 5
## model=VW-Amarok
## model=Mercedes-X-CLASS
## model=BMW-M4
## model=Audi-Q3
## model=BMW-M2
## model=BMW-Z4
## model=VW-T-Roc
## model=VW-Caravelle
## model=VW-Touareg
## engineSize=f. EngSize-2. 3
## model=Mercedes-GLE Class
## model=BMW-X2
## model=Audi-TT
## model=Audi-Q7
## engineSize=f. EngSize-2. 9
## model=Mercedes-GLC Class
## model=Audi-S4
## model=Audi-Q2
## engineSize=f. EngSize-1. 3
## model=VW-Tiguan
## model=VW-Sharan
## year=f. Year-2012
## model=Mercedes-E Class
## year=f. Year-2011
## year=f. Year-2010
## model=VW-Scirocco
## model=BMW-1 Series

```

	Cla/Mod	Mod/Cla	Global
61. 7937220	93. 9972715	22. 30	
49. 3076923	87. 4488404	26. 00	
37. 2784150	97. 5443383	38. 36	
44. 3290735	75. 7162347	25. 04	
42. 0800000	71. 7598909	25. 00	
72. 5856698	31. 7871760	6. 42	
30. 1815905	65. 7571623	31. 94	
23. 1468197	66. 0300136	41. 82	
19. 7160374	83. 3560709	61. 98	
16. 1560503	100. 0000000	90. 74	
21. 0359408	54. 2974079	37. 84	
25. 9786477	19. 9181446	11. 24	
43. 0107527	5. 4570259	1. 86	
20. 0921659	29. 7407913	21. 70	
20. 0921659	29. 7407913	21. 70	
17. 9413179	50. 8867667	41. 58	
71. 4285714	1. 3642565	0. 28	
21. 1573237	15. 9618008	11. 06	
62. 5000000	1. 3642565	0. 32	
72. 7272727	1. 0914052	0. 22	
72. 7272727	1. 0914052	0. 22	
28. 1690141	5. 4570259	2. 84	
100. 0000000	0. 6821282	0. 10	
60. 0000000	1. 2278308	0. 30	
33. 3333333	2. 8649386	1. 26	
75. 0000000	0. 8185539	0. 16	
40. 6250000	1. 7735334	0. 64	
58. 3333333	0. 9549795	0. 24	
30. 9090909	2. 3192360	1. 10	
40. 9090909	1. 2278308	0. 44	
35. 4838710	1. 5006821	0. 62	
34. 3750000	1. 5006821	0. 64	
43. 7500000	0. 9549795	0. 32	
23. 5294118	3. 8199181	2. 38	
75. 0000000	0. 4092769	0. 08	
25. 0000000	2. 7285130	1. 60	
25. 0000000	2. 5920873	1. 52	
21. 3414634	4. 7748977	3. 28	
31. 8181818	0. 9549795	0. 44	
0. 0000000	0. 0000000	0. 38	
9. 4444444	2. 3192360	3. 60	
0. 0000000	0. 0000000	0. 42	
0. 0000000	0. 0000000	0. 42	
0. 0000000	0. 0000000	0. 48	
9. 1370558	2. 4556617	3. 94	

## model=Audi-A3	9. 0452261	2. 4556617	3. 98
## engineSize=f. EngSize-1. 8	0. 0000000	0. 0000000	0. 52
## model=VW-Golf	11. 0655738	7. 3669850	9. 76
## model=Mercedes-B Class	3. 1746032	0. 2728513	1. 26
## model=VW-Up	4. 5454545	0. 5457026	1. 76
## year=f. Year-2009 or before	0. 0000000	0. 0000000	0. 84
## model=BMW-5 Series	3. 4883721	0. 4092769	1. 72
## model=BMW-3 Series	7. 4074074	2. 4556617	4. 86
## model=VW-Polo	8. 0459770	3. 8199181	6. 96
## engineSize=f. EngSize-1	8. 3538084	4. 6384720	8. 14
## model=Mercedes-C Class	7. 9019074	3. 9563438	7. 34
## manufacturer=Mercedes	11. 1769060	20. 6002729	27. 02
## model=Mercedes-A Class	6. 2962963	2. 3192360	5. 40
## Audi=Audi No	13. 1545338	70. 2592087	78. 30
## year=f. Year-2013	0. 0000000	0. 0000000	2. 42
## claHC=claHCPC-5	0. 0000000	0. 0000000	2. 46
## engineSize=f. EngSize-1. 2	0. 0000000	0. 0000000	2. 50
## engineSize=f. EngSize-1. 6	3. 6827195	1. 7735334	7. 06
## year=f. Year-2014	0. 0000000	0. 0000000	3. 94
## engineSize=f. EngSize-1. 4	1. 3840830	0. 5457026	5. 78
## year=f. Year-2018	3. 6511156	2. 4556617	9. 86
## claHC=claHCPC-6	0. 0000000	0. 0000000	5. 26
## aux_tax=f. tax-[0, 125]	0. 0000000	0. 0000000	5. 94
## aux_tax=f. tax-(145, 580]	7. 6059850	16. 6439291	32. 08
## aux_mpg=f. mpg-(44. 8, 53. 3]	6. 6763425	12. 5511596	27. 56
## engineSize=f. EngSize-2. 1	0. 4914005	0. 2728513	8. 14
## year=f. Year-2015	0. 0000000	0. 0000000	8. 38
## claHC=claHCPC-7	0. 0000000	0. 0000000	8. 58
## claHC=claHCPC-8	0. 0000000	0. 0000000	9. 26
## mout=YesMOut	0. 0000000	0. 0000000	9. 26
## transmission=f. Trans-Manual	6. 7409471	16. 5075034	35. 90
## claHC=claHCPC-2	4. 0552995	6. 0027285	21. 70
## claHC=claHCPC-3	0. 0000000	0. 0000000	11. 44
## fuelType=f. Fuel-Diesel	8. 4799437	32. 8785812	56. 84
## aux_price=f. price-(14, 19. 5]	2. 4879615	4. 2291951	24. 92
## aux_years_sell=f. years_sell-(6, 23]	0. 0000000	0. 0000000	16. 80
## aux_years_sell=f. years_sell-(5, 6]	0. 0000000	0. 0000000	17. 24
## year=f. Year-2016	0. 0000000	0. 0000000	17. 24
## year=f. Year-2017	0. 0000000	0. 0000000	17. 74
## claHC=claHCPC-4	0. 0000000	0. 0000000	19. 00
## aux_years_sell=f. years_sell-(3, 5]	1. 3043478	2. 4556617	27. 60
## aux_mpg=f. mpg-(61. 4, 88. 3]	0. 0000000	0. 0000000	22. 68
## aux_price=f. price-[0. 65, 14]	0. 3971406	0. 6821282	25. 18
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]	0. 2400000	0. 4092769	25. 00
## aux_mpg=f. mpg-(53. 3, 61. 4]	0. 0000000	0. 0000000	23. 76
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]	0. 0000000	0. 0000000	25. 00
##	p. value	v. test	
## claHC=claHCPC-1	0. 000000e+00	Inf	
## aux_mpg=f. mpg-[8. 8, 44. 8]	0. 000000e+00	Inf	
## aux_years_sell=f. years_sell-[2, 3]	1. 484659e-308	37. 548606	
## aux_mileage=f. mileage-[4, 6e+03]	1. 211501e-222	31. 852602	
## aux_price=f. price-(26, 110]	8. 380305e-190	29. 385637	
## year=f. Year-2020	2. 476815e-134	24. 666032	
## year=f. Year-2019	3. 311525e-93	20. 479035	
## fuelType=f. Fuel-Petrol	2. 169158e-46	14. 300605	
## aux_tax=f. tax-(125, 145]	5. 908612e-42	13. 571517	
## mout=NoMOut	2. 583412e-34	12. 215050	
## transmission=f. Trans-SemiAuto	1. 203069e-22	9. 793298	
## engineSize=f. EngSize-3	6. 706285e-14	7. 493496	
## model=Audi-Q5	2. 915968e-11	6. 650753	
## Audi=Audi Yes	3. 026714e-08	5. 539884	
## manufacturer=Audi	3. 026714e-08	5. 539884	
## engineSize=f. EngSize-2	3. 877292e-08	5. 496352	
## model=VW-Tiguan Allspace	2. 627088e-06	4. 698010	

```

## engineSize=f.EngSize-1.5
## model=VW-Amarok
## model=Mercedes-X-CLASS
## model=BMW-M4
## model=Audi-Q3
## model=BMW-M2
## model=BMW-Z4
## model=VW-T-Roc
## model=VW-Caravelle
## model=VW-Touareg
## engineSize=f.EngSize-2.3
## model=Mercedes-GLE Class
## model=BMW-X2
## model=Audi-TT
## model=Audi-Q7
## engineSize=f.EngSize-2.9
## model=Mercedes-GLC Class
## model=Audi-S4
## model=Audi-Q2
## engineSize=f.EngSize-1.3
## model=VW-Tiguan
## model=VW-Sharan
## year=f.Year-2012
## model=Mercedes-E Class
## year=f.Year-2011
## year=f.Year-2010
## model=VW-Scirocco
## model=BMW-1 Series
## model=Audi-A3
## engineSize=f.EngSize-1.8
## model=VW-Golf
## model=Mercedes-B Class
## model=VW-Up
## year=f.Year-2009 or before
## model=BMW-5 Series
## model=BMW-3 Series
## model=VW-Polo
## engineSize=f.EngSize-1
## model=Mercedes-C Class
## manufacturer=Mercedes
## model=Mercedes-A Class
## Audi=Audi No
## year=f.Year-2013
## claHC=claHCPC-5
## engineSize=f.EngSize-1.2
## engineSize=f.EngSize-1.6
## year=f.Year-2014
## engineSize=f.EngSize-1.4
## year=f.Year-2018
## claHC=claHCPC-6
## aux_tax=f.tax-[0, 125]
## aux_tax=f.tax-(145, 580]
## aux_mpg=f.mpg-(44.8, 53.3]
## engineSize=f.EngSize-2.1
## year=f.Year-2015
## claHC=claHCPC-7
## claHC=claHCPC-8
## mout=YesMOut
## transmission=f.Trans-Manual
## claHC=claHCPC-2
## claHC=claHCPC-3
## fuelType=f.Fuel-Diesel
## aux_price=f.price-(14, 19.5]
## aux_years_sell=f.years_sell-(6, 23]
1. 191822e-05 4. 379079
1. 631788e-05 4. 310103
2. 376760e-05 4. 226195
2. 376760e-05 4. 226195
2. 572944e-05 4. 208304
6. 692704e-05 3. 986954
7. 149032e-05 3. 971271
1. 888055e-04 3. 733542
2. 192125e-04 3. 695780
3. 702320e-04 3. 560441
6. 302370e-04 3. 418256
2. 138163e-03 3. 070339
2. 984574e-03 2. 969322
4. 058290e-03 2. 873595
5. 437668e-03 2. 779894
5. 716613e-03 2. 763608
9. 374435e-03 2. 598088
1. 164132e-02 2. 522834
1. 428945e-02 2. 449904
1. 697687e-02 2. 387208
1. 901836e-02 2. 345171
4. 240899e-02 2. 029484
4. 890371e-02 -1. 969430
3. 674265e-02 -2. 088612
3. 556827e-02 -2. 101829
3. 556827e-02 -2. 101829
2. 205622e-02 -2. 289398
1. 972403e-02 -2. 331557
1. 701913e-02 -2. 386294
1. 603619e-02 -2. 408091
1. 535439e-02 -2. 423910
3. 576024e-03 -2. 913325
2. 776146e-03 -2. 991495
1. 245909e-03 -3. 228156
8. 503236e-04 -3. 335872
4. 443153e-04 -3. 512259
1. 186598e-04 -3. 848879
6. 920615e-05 -3. 978999
4. 997198e-05 -4. 055758
1. 503975e-05 -4. 328105
1. 302203e-05 -4. 359735
3. 026714e-08 -5. 539884
3. 624228e-09 -5. 900490
2. 616958e-09 -5. 953982
1. 889361e-09 -6. 007044
4. 114052e-12 -6. 933207
1. 382098e-14 -7. 698002
1. 075051e-15 -8. 017972
7. 743685e-17 -8. 335094
2. 283801e-19 -8. 998713
7. 359986e-22 -9. 608519
2. 744227e-24 -10. 168384
8. 870024e-26 -10. 497498
1. 459091e-26 -10. 666573
5. 771741e-31 -11. 571130
1. 007490e-31 -11. 719935
2. 583412e-34 -12. 215050
2. 583412e-34 -12. 215050
1. 152172e-35 -12. 465452
1. 021641e-35 -12. 475033
9. 122395e-43 -13. 707766
2. 000911e-45 -14. 145149
3. 749743e-58 -16. 076149
1. 474281e-64 -16. 965662

```

```

## aux_years_sell=f.years_sell-(5, 6] 2. 048610e-66 -17. 215024
## year=f.Year-2016 2. 048610e-66 -17. 215024
## year=f.Year-2017 1. 541052e-68 -17. 495847
## claHC=claHCPC-4 5. 920260e-74 -18. 192477
## aux_years_sell=f.years_sell-(3, 5] 9. 761156e-84 -19. 387916
## aux_mpg=f.mpg-(61. 4, 88. 3] 2. 540477e-90 -20. 152852
## aux_price=f.price-[0. 65, 14] 1. 384530e-91 -20. 296366
## aux_mileage=f.mileage-(1. 65e+04, 3. 35e+04] 2. 035721e-94 -20. 614462
## aux_mpg=f.mpg-(53. 3, 61. 4] 2. 732967e-95 -20. 711417
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05] 4. 311633e-101 -21. 345303
##
## $ '3'
##
## aux_years_sell=f.years_sell-[2, 3] 50. 9906152 89. 72477064 38. 36
## year=f.Year-2019 56. 9818410 83. 48623853 31. 94
## claHC=claHCPC-2 61. 6589862 61. 37614679 21. 70
## aux_mpg=f.mpg-(44. 8, 53. 3] 45. 8635704 57. 98165138 27. 56
## aux_tax=f.tax-(125, 145] 32. 0103259 91. 00917431 61. 98
## aux_mileage=f.mileage-[4, 6e+03] 46. 4057508 53. 30275229 25. 04
## aux_mileage=f.mileage-(6e+03, 1. 65e+04] 39. 0224359 44. 67889908 24. 96
## mout=NoMOut 24. 0246859 100. 00000000 90. 74
## claHC=claHCPC-1 37. 6681614 38. 53211009 22. 30
## aux_price=f.price-(19. 5, 26] 35. 0200803 40. 00000000 24. 90
## aux_price=f.price-(26, 110] 32. 7200000 37. 52293578 25. 00
## engineSize=f.EngSize-1.5 37. 0705244 18. 80733945 11. 06
## engineSize=f.EngSize-1.3 67. 1052632 4. 67889908 1. 52
## engineSize=f.EngSize-1 38. 3292383 14. 31192661 8. 14
## model=VW-T-Cross 80. 00000000 1. 83486239 0. 50
## model=Mercedes-B Class 53. 9682540 3. 11926606 1. 26
## model=VW-Arteon 59. 2592593 1. 46788991 0. 54
## model=BMW-2 Series 36. 9565217 4. 67889908 2. 76
## Audi=Audi No 23. 0395913 82. 75229358 78. 30
## fuelType=f.Fuel-Petrol 24. 6293639 47. 24770642 41. 82
## model=VW-T-Roc 44. 4444444 2. 56880734 1. 26
## model=Audi-Q2 41. 2500000 3. 02752294 1. 60
## model=Mercedes-C Class 29. 1553134 9. 81651376 7. 34
## manufacturer=VW 24. 5544554 34. 12844037 30. 30
## transmission=f.Trans-SemiAuto 24. 0486258 41. 74311927 37. 84
## model=VW-Golf 27. 0491803 12. 11009174 9. 76
## model=VW-Sharan 50. 00000000 1. 00917431 0. 44
## engineSize=f.EngSize-2.2 60. 00000000 0. 55045872 0. 20
## model=VW-Golf SV 40. 00000000 0. 91743119 0. 50
## engineSize=f.EngSize-4 7. 1428571 0. 18348624 0. 56
## model=Mercedes-GLE Class 10. 9090909 0. 55045872 1. 10
## model=Mercedes-GL Class 0. 00000000 0. 00000000 0. 28
## model=Audi-A7 0. 00000000 0. 00000000 0. 28
## model=BMW-Z4 0. 00000000 0. 00000000 0. 30
## model=VW-Scirocco 4. 1666667 0. 09174312 0. 48
## model=VW-Tiguan 14. 6341463 2. 20183486 3. 28
## model=Mercedes-E Class 14. 4444444 2. 38532110 3. 60
## year=f.Year-2012 0. 00000000 0. 00000000 0. 38
## year=f.Year-2011 0. 00000000 0. 00000000 0. 42
## year=f.Year-2010 0. 00000000 0. 00000000 0. 42
## model=Audi-TT 3. 2258065 0. 09174312 0. 62
## model=Audi-Q7 3. 1250000 0. 09174312 0. 64
## model=BMW-1 Series 13. 7055838 2. 47706422 3. 94
## model=Mercedes-GLA Class 9. 5238095 0. 73394495 1. 68
## model=BMW-X5 3. 0303030 0. 09174312 0. 66
## model=Audi-Q3 11. 9718310 1. 55963303 2. 84
## engineSize=f.EngSize-1.8 0. 00000000 0. 00000000 0. 52
## engineSize=f.EngSize-4.2 or more 0. 00000000 0. 00000000 0. 56
## model=Audi-A1 10. 9489051 1. 37614679 2. 74
## transmission=f.Trans-Manual 19. 1643454 31. 55963303 35. 90
## aux_mpg=f.mpg-[8. 8, 44. 8] 18. 3076923 21. 83486239 26. 00

```

```

## fuelType=f. Fuel-Diesel
## Audi=Audi Yes
## manufacturer=Audi
## year=f. Year-2009 or before
## model=Audi-Q5
## engineSize=f. EngSize-3
## aux_mpg=f. mpg-(53. 3, 61. 4]
## aux_price=f. price-(14, 19. 5]
## year=f. Year-2013
## claHC=claHCPC-5
## engineSize=f. EngSize-1. 2
## engineSize=f. EngSize-1. 4
## year=f. Year-2014
## engineSize=f. EngSize-2. 1
## claHC=claHCPC-6
## aux_tax=f. tax-[0, 125]
## year=f. Year-2015
## claHC=claHCPC-7
## claHC=claHCPC-8
## mout=YesMOut
## aux_years_sell=f. years_sell-(3, 5]
## claHC=claHCPC-3
## aux_price=f. price-[0. 65, 14]
## aux_tax=f. tax-(145, 580]
## aux_mpg=f. mpg-(61. 4, 88. 3]
## year=f. Year-2017
## aux_years_sell=f. years_sell-(6, 23]
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## claHC=claHCPC-4
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]
## aux_years_sell=f. years_sell-[2, 3]
## year=f. Year-2019
## claHC=claHCPC-2
## aux_mpg=f. mpg-(44. 8, 53. 3]
## aux_tax=f. tax-(125, 145]
## aux_mileage=f. mileage-[4, 6e+03]
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]
## mout=NoMOut
## claHC=claHCPC-1
## aux_price=f. price-(19. 5, 26]
## aux_price=f. price-(26, 110]
## engineSize=f. EngSize-1. 5
## engineSize=f. EngSize-1. 3
## engineSize=f. EngSize-1
## model=VW-T-Cross
## model=Mercedes-B Class
## model=VW-Arteon
## model=BMW-2 Series
## Audi=Audi No
## fuelType=f. Fuel-Petrol
## model=VW-T-Roc
## model=Audi-Q2
## model=Mercedes-C Class
## manufacturer=VW
## transmission=f. Trans-SemiAuto
## model=VW-Golf
## model=VW-Sharan
## engineSize=f. EngSize-2. 2
## model=VW-Golf SV
## engineSize=f. EngSize-4
## model=Mercedes-GLE Class
19. 7044335 51. 37614679 56. 84
17. 3271889 17. 24770642 21. 70
17. 3271889 17. 24770642 21. 70
0. 0000000 0. 00000000 0. 84
5. 3763441 0. 45871560 1. 86
13. 3451957 6. 88073394 11. 24
15. 9090909 17. 33944954 23. 76
14. 5264848 16. 60550459 24. 92
0. 0000000 0. 00000000 2. 42
0. 0000000 0. 00000000 2. 46
0. 0000000 0. 00000000 2. 50
2. 7681661 0. 73394495 5. 78
0. 0000000 0. 00000000 3. 94
3. 9312039 1. 46788991 8. 14
0. 0000000 0. 00000000 5. 26
0. 0000000 0. 00000000 5. 94
0. 0000000 0. 00000000 8. 38
0. 0000000 0. 00000000 8. 58
0. 0000000 0. 00000000 9. 26
0. 0000000 0. 00000000 9. 26
8. 1159420 10. 27522936 27. 60
0. 1748252 0. 09174312 11. 44
5. 0833995 5. 87155963 25. 18
6. 1097257 8. 99082569 32. 08
2. 7336861 2. 84403670 22. 68
0. 6764374 0. 55045872 17. 74
0. 0000000 0. 00000000 16. 80
0. 0000000 0. 00000000 17. 24
0. 0000000 0. 00000000 17. 24
0. 0000000 0. 00000000 19. 00
1. 7600000 2. 01834862 25. 00
0. 0000000 0. 00000000 25. 00
## p. value v. test
0. 000000e+00 Inf
0. 000000e+00 Inf
1. 084861e-246 33. 544446
7. 228788e-131 24. 340951
3. 656820e-129 24. 179491
1. 434006e-118 23. 150590
9. 396875e-60 16. 302997
5. 850783e-53 15. 317404
4. 426993e-44 13. 925617
2. 826505e-36 12. 576974
1. 239861e-25 10. 465829
2. 824253e-18 8. 718294
1. 978169e-17 8. 495068
2. 447959e-15 7. 916243
9. 229130e-10 6. 122201
2. 432215e-08 5. 578056
3. 048507e-05 4. 169812
4. 020121e-05 4. 106320
4. 039098e-05 4. 105232
4. 274947e-05 4. 092094
6. 020425e-05 4. 012009
8. 535124e-05 3. 928855
5. 926305e-04 3. 434965
2. 026870e-03 3. 086267
2. 778798e-03 2. 991204
3. 827081e-03 2. 892074
4. 007645e-03 2. 877559
1. 132554e-02 2. 532491
4. 103438e-02 2. 043182
4. 753606e-02 -1. 981493
4. 026174e-02 -2. 051054

```

```

## model=Mercedes-GL Class
## model=Audi-A7
## model=BMW-Z4
## model=VW-Scirocco
## model=VW-Tiguan
## model=Mercedes-E Class
## year=f. Year-2012
## year=f. Year-2011
## year=f. Year-2010
## model=Audi-TT
## model=Audi-Q7
## model=BMW-1 Series
## model=Mercedes-GLA Class
## model=BMW-X5
## model=Audi-Q3
## engineSize=f. EngSize-1.8
## engineSize=f. EngSize-4.2 or more
## model=Audi-A1
## transmission=f. Trans-Manual
## aux_mpg=f. mpg-[8.8, 44.8]
## fuelType=f. Fuel-Diesel
## Audi=Audi Yes
## manufacturer=Audi
## year=f. Year-2009 or before
## model=Audi-Q5
## engineSize=f. EngSize-3
## aux_mpg=f. mpg-(53.3, 61.4]
## aux_price=f. price-(14, 19.5]
## year=f. Year-2013
## claHC=claHCPC-5
## engineSize=f. EngSize-1.2
## engineSize=f. EngSize-1.4
## year=f. Year-2014
## engineSize=f. EngSize-2.1
## claHC=claHCPC-6
## aux_tax=f. tax-[0, 125]
## year=f. Year-2015
## claHC=claHCPC-7
## claHC=claHCPC-8
## mout=YesMOut
## aux_years_sell=f. years_sell-(3, 5]
## claHC=claHCPC-3
## aux_price=f. price-[0.65, 14]
## aux_tax=f. tax-(145, 580]
## aux_mpg=f. mpg-(61.4, 88.3]
## year=f. Year-2017
## aux_years_sell=f. years_sell-(6, 23]
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
## claHC=claHCPC-4
## aux_mileage=f. mileage-(1.65e+04, 3.35e+04]
## aux_mileage=f. mileage-(3.35e+04, 1.16e+05]
## $ '4 '
## claHC=claHCPC-5
## aux_tax=f. tax-(145, 580]
## aux_years_sell=f. years_sell-(6, 23]
## engineSize=f. EngSize-3
## aux_mileage=f. mileage-(3.35e+04, 1.16e+05]
## aux_mpg=f. mpg-[8.8, 44.8]
## year=f. Year-2015
## aux_years_sell=f. years_sell-(5, 6]
## year=f. Year-2016
3. 181900e-02 -2. 146677
3. 181900e-02 -2. 146677
2. 486298e-02 -2. 243525
2. 351124e-02 -2. 265023
1. 971529e-02 -2. 331723
1. 179387e-02 -2. 518253
9. 263520e-03 -2. 602173
5. 652506e-03 -2. 767286
5. 652506e-03 -2. 767286
5. 103131e-03 -2. 800451
4. 089077e-03 -2. 871207
3. 451131e-03 -2. 924407
3. 287546e-03 -2. 939488
3. 274209e-03 -2. 940748
2. 428369e-03 -3. 032127
1. 642387e-03 -3. 148272
1. 001377e-03 -3. 290139
9. 127842e-04 -3. 316115
6. 792431e-04 -3. 397823
3. 338111e-04 -3. 587541
4. 040164e-05 -4. 105171
4. 039098e-05 -4. 105232
4. 039098e-05 -4. 105232
3. 116758e-05 -4. 164763
1. 310892e-05 -4. 358279
6. 780780e-08 -5. 396885
7. 842093e-09 -5. 771818
1. 310574e-13 -7. 405094
7. 923887e-14 -7. 471578
4. 778595e-14 -7. 537829
2. 881109e-14 -7. 603552
1. 039480e-21 -9. 572900
3. 018658e-22 -9. 699876
4. 399203e-26 -10. 563505
1. 107366e-29 -11. 314894
1. 470741e-33 -12. 072783
9. 724836e-48 -14. 515077
6. 403888e-49 -14. 700444
5. 850783e-53 -15. 317404
5. 850783e-53 -15. 317404
1. 554844e-54 -15. 551475
7. 158559e-64 -16. 872592
3. 343542e-76 -18. 474002
1. 650812e-89 -20. 060001
6. 519345e-93 -20. 446011
2. 612390e-94 -20. 602388
3. 069886e-100 -21. 253346
3. 762842e-103 -21. 565809
3. 762842e-103 -21. 565809
5. 561127e-115 -22. 791551
2. 488876e-119 -23. 225972
1. 033657e-157 -26. 756636
## Cla/Mod Mod/Cla Global
100. 00000000 99. 1935484 2. 46
7. 73067332 100. 0000000 32. 08
6. 90476190 46. 7741935 16. 80
7. 82918149 35. 4838710 11. 24
5. 20000000 52. 4193548 25. 00
4. 92307692 51. 6129032 26. 00
7. 63723150 25. 8064516 8. 38
5. 56844548 38. 7096774 17. 24
5. 56844548 38. 7096774 17. 24

```

		p. value	v. test
## model=BMW-X5	27. 27272727	7. 2580645	0. 66
## aux_price=f. price-(19. 5, 26]	4. 57831325	45. 9677419	24. 90
## aux_mpg=f. mpg-(44. 8, 53. 3]	4. 28156749	47. 5806452	27. 56
## mout=NoMOut	2. 73308354	100. 0000000	90. 74
## model=Audi-Q5	11. 82795699	8. 8709677	1. 86
## year=f. Year-2014	7. 61421320	12. 0967742	3. 94
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]	4. 00000000	40. 3225806	25. 00
## engineSize=f. EngSize-2	3. 41510342	57. 2580645	41. 58
## year=f. Year-2013	8. 26446281	8. 0645161	2. 42
## model=BMW-6 Series	30. 00000000	2. 4193548	0. 20
## model=Audi-Q3	7. 04225352	8. 0645161	2. 84
## model=Audi-A8	18. 75000000	2. 4193548	0. 32
## model=Mercedes-CLS Class	16. 66666667	2. 4193548	0. 36
## model=BMW-X4	16. 66666667	2. 4193548	0. 36
## Audi=Audi Yes	3. 59447005	31. 4516129	21. 70
## manufacturer=Audi	3. 59447005	31. 4516129	21. 70
## model=BMW-X6	28. 57142857	1. 6129032	0. 14
## transmission=f. Trans-Automatic	3. 42726580	36. 2903226	26. 26
## model=Mercedes-M Class	25. 00000000	1. 6129032	0. 16
## transmission=f. Trans-SemiAuto	3. 06553911	46. 7741935	37. 84
## model=VW-Golf	3. 89344262	15. 3225806	9. 76
## model=Mercedes-GLC Class	0. 00000000	0. 0000000	2. 38
## engineSize=f. EngSize-1. 2	0. 00000000	0. 0000000	2. 50
## model=Audi-A4	0. 00000000	0. 0000000	2. 72
## model=Audi-A1	0. 00000000	0. 0000000	2. 74
## engineSize=f. EngSize-1. 4	0. 69204152	1. 6129032	5. 78
## manufacturer=Mercedes	1. 62842339	17. 7419355	27. 02
## Audi=Audi No	2. 17113665	68. 5483871	78. 30
## model=Mercedes-E Class	0. 00000000	0. 0000000	3. 60
## aux_price=f. price-(26, 110]	1. 44000000	14. 5161290	25. 00
## claHC=claHCPC-6	0. 00000000	0. 0000000	5. 26
## model=Mercedes-C Class	0. 27247956	0. 8064516	7. 34
## aux_price=f. price-[0. 65, 14]	1. 27084988	12. 9032258	25. 18
## aux_years_sell=f. years_sell-(3, 5]	1. 30434783	14. 5161290	27. 60
## aux_tax=f. tax-[0, 125]	0. 00000000	0. 0000000	5. 94
## year=f. Year-2020	0. 00000000	0. 0000000	6. 42
## claHC=claHCPC-7	0. 23310023	0. 8064516	8. 58
## model=VW-Polo	0. 00000000	0. 0000000	6. 96
## engineSize=f. EngSize-1. 6	0. 00000000	0. 0000000	7. 06
## engineSize=f. EngSize-1	0. 00000000	0. 0000000	8. 14
## claHC=claHCPC-8	0. 00000000	0. 0000000	9. 26
## mout=YesMOut	0. 00000000	0. 0000000	9. 26
## transmission=f. Trans-Manual	1. 16991643	16. 9354839	35. 90
## year=f. Year-2018	0. 00000000	0. 0000000	9. 86
## engineSize=f. EngSize-1. 5	0. 00000000	0. 0000000	11. 06
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]	0. 72115385	7. 2580645	24. 96
## claHC=claHCPC-3	0. 00000000	0. 0000000	11. 44
## claHC=claHCPC-4	0. 00000000	0. 0000000	19. 00
## aux_mpg=f. mpg-(61. 4, 88. 3]	0. 08818342	0. 8064516	22. 68
## claHC=claHCPC-2	0. 00000000	0. 0000000	21. 70
## claHC=claHCPC-1	0. 00000000	0. 0000000	22. 30
## aux_mpg=f. mpg-(53. 3, 61. 4]	0. 00000000	0. 0000000	23. 76
## aux_mileage=f. mileage-[4, 6e+03]	0. 00000000	0. 0000000	25. 04
## year=f. Year-2019	0. 00000000	0. 0000000	31. 94
## aux_years_sell=f. years_sell-[2, 3]	0. 00000000	0. 0000000	38. 36
## aux_tax=f. tax-(125, 145]	0. 00000000	0. 0000000	61. 98
##			
## claHC=claHCPC-5	7. 271869e-248	33. 624846	
## aux_tax=f. tax-(145, 580]	2. 094315e-63	16. 809070	
## aux_years_sell=f. years_sell-(6, 23]	4. 184290e-15	7. 849281	
## engineSize=f. EngSize-3	6. 029798e-13	7. 199807	
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]	3. 985859e-11	6. 604602	
## aux_mpg=f. mpg-[8. 8, 44. 8]	8. 282529e-10	6. 139416	
## year=f. Year-2015	4. 579058e-09	5. 861789	

```

## aux_years_sell=f.years_sell-(5, 6] 9.535900e-09 5.738784
## year=f.Year-2016 9.535900e-09 5.738784
## model=BMW-X5 6.544170e-08 5.403257
## aux_price=f.price-(19.5, 26] 2.737784e-07 5.140658
## aux_mpg=f.mpg-(44.8, 53.3] 1.682204e-06 4.788276
## mout=NoMOut 4.991816e-06 4.565131
## model=Audi-Q5 1.763827e-05 4.292865
## year=f.Year-2014 1.134463e-04 3.859872
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 1.518566e-04 3.788014
## engineSize=f.EngSize-2 3.972387e-04 3.541912
## year=f.Year-2013 9.325408e-04 3.310126
## model=BMW-6 Series 1.640267e-03 3.148650
## model=Audi-Q3 3.269376e-03 2.941205
## model=Audi-A8 7.103710e-03 2.691943
## model=Mercedes-CLS Class 1.009465e-02 2.572570
## model=BMW-X4 1.009465e-02 2.572570
## Audi=Audi Yes 1.057283e-02 2.556509
## manufacturer=Audi 1.057283e-02 2.556509
## model=BMW-X6 1.229328e-02 2.503612
## transmission=f.Trans-Automatic 1.286622e-02 2.487451
## model=Mercedes-M Class 1.625249e-02 2.403196
## transmission=f.Trans-SemiAuto 4.038975e-02 2.049741
## model=VW-Golf 4.667769e-02 1.989213
## model=Mercedes-GLC Class 4.857224e-02 -1.972327
## engineSize=f.EngSize-1.2 4.161974e-02 -2.037302
## model=Audi-A4 3.133905e-02 -2.152741
## model=Audi-A1 3.054011e-02 -2.163014
## engineSize=f.EngSize-1.4 2.701161e-02 -2.211350
## manufacturer=Mercedes 1.526728e-02 -2.425976
## Audi=Audi No 1.057283e-02 -2.556509
## model=Mercedes-E Class 1.000905e-02 -2.575516
## aux_price=f.price-(26, 110] 4.362603e-03 -2.850679
## claHC=claHCPC-6 1.129091e-03 -3.256213
## model=Mercedes-C Class 8.388829e-04 -3.339636
## aux_price=f.price-[0.65, 14] 7.452915e-04 -3.372352
## aux_years_sell=f.years_sell-(3, 5] 5.128147e-04 -3.473972
## aux_tax=f.tax-[0, 125] 4.567325e-04 -3.504927
## year=f.Year-2020 2.401405e-04 -3.672552
## claHC=claHCPC-7 1.781239e-04 -3.748178
## model=VW-Polo 1.160457e-04 -3.854333
## engineSize=f.EngSize-1.6 1.013765e-04 -3.887274
## engineSize=f.EngSize-1 2.333207e-05 -4.230357
## claHC=claHCPC-8 4.991816e-06 -4.565131
## mout=YesMOut 4.991816e-06 -4.565131
## transmission=f.Trans-Manual 2.785021e-06 -4.686069
## year=f.Year-2018 2.167893e-06 -4.737108
## engineSize=f.EngSize-1.5 4.019794e-07 -5.068018
## aux_mileage=f.mileage-(6e+03, 1.65e+04] 2.944867e-07 -5.126942
## claHC=claHCPC-3 2.346213e-07 -5.169579
## claHC=claHCPC-4 3.117802e-12 -6.972309
## aux_mpg=f.mpg-(61.4, 88.3] 3.527652e-13 -7.272550
## claHC=claHCPC-2 4.358821e-14 -7.549813
## claHC=claHCPC-1 1.653648e-14 -7.675042
## aux_mpg=f.mpg-(53.3, 61.4] 1.514369e-15 -7.975767
## aux_mileage=f.mileage-[4, 6e+03] 1.791654e-16 -8.235262
## year=f.Year-2019 9.147008e-22 -9.586108
## aux_years_sell=f.years_sell-[2, 3] 3.323699e-27 -10.803212
## aux_tax=f.tax-(125, 145] 6.429876e-54 -15.460300
##
## $ '5 '
## claHC=claHCPC-4 55.36842105 68.0465718 19.00
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 44.64000000 72.1862872 25.00
## year=f.Year-2017 46.33596392 53.1694696 17.74

```

```

## aux_years_sell=f.years_sell-(3, 5]
## aux_mpg=f.mpg-(53.3, 61.4]
## claHC=claHCPC-3
## mout=NoMOut
## aux_years_sell=f.years_sell-(5, 6]
## year=f.Year-2016
## engineSize=f.EngSize-1.2
## model=VW-Polo
## engineSize=f.EngSize-1.4
## aux_price=f.price-(14, 19.5]
## aux_price=f.price-[0.65, 14]
## transmission=f.Trans-Manual
## aux_mpg=f.mpg-(61.4, 88.3]
## engineSize=f.EngSize-1
## manufacturer=VW
## model=Audi-A1
## model=VW-Up
## engineSize=f.EngSize-2.1
## fuelType=f.Fuel-Petrol
## aux_tax=f.tax-(145, 580]
## manufacturer=Mercedes
## year=f.Year-2012
## model=Audi-Q2
## model=Audi-Q7
## year=f.Year-2018
## year=f.Year-2011
## year=f.Year-2010
## aux_price=f.price-(19.5, 26]
## model=BMW-X2
## aux_tax=f.tax-[0, 125]
## model=VW-T-Cross
## fuelType=f.Fuel-Diesel
## transmission=f.Trans-SemiAuto
## model=BMW-X5
## year=f.Year-2009 or before
## model=Audi-Q5
## engineSize=f.EngSize-3
## model=VW-T-Roc
## manufacturer=BMW
## year=f.Year-2015
## transmission=f.Trans-Automatic
## engineSize=f.EngSize-1.3
## engineSize=f.EngSize-2
## year=f.Year-2013
## claHC=claHCPC-5
## engineSize=f.EngSize-1.5
## aux_mpg=f.mpg-(44.8, 53.3]
## aux_mileage=f.mileage-(6e+03, 1.65e+04]
## year=f.Year-2014
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05]
## year=f.Year-2020
## aux_years_sell=f.years_sell-(6, 23]
## claHC=claHCPC-7
## claHC=claHCPC-8
## mout=YesMOut
## aux_price=f.price-(26, 110]
## aux_mpg=f.mpg-[8.8, 44.8]
## claHC=claHCPC-2
## claHC=claHCPC-1
## aux_mileage=f.mileage-[4, 6e+03]
## year=f.Year-2019
## aux_years_sell=f.years_sell-[2, 3]
## claHC=claHCPC-4

34.13043478 60.9314360 27.60
29.62962963 45.5368693 23.76
37.58741259 27.8137128 11.44
17.03769010 100.0000000 90.74
30.39443155 33.8939198 17.24
30.39443155 33.8939198 17.24
60.00000000 9.7024580 2.50
38.21839080 17.2056921 6.96
35.29411765 13.1953428 5.78
22.55216693 36.3518758 24.92
22.47815727 36.6106080 25.18
19.83286908 46.0543338 35.90
21.51675485 31.5653299 22.68
26.28992629 13.8421734 8.14
20.06600660 39.3272962 30.30
28.46715328 5.0452781 2.74
30.68181818 3.4928849 1.76
21.37592138 11.2548512 8.14
16.78622669 45.4075032 41.82
16.95760599 35.1875809 32.08
13.76757957 24.0620957 27.02
0.00000000 0.0000000 0.38
7.50000000 0.7761966 1.60
3.12500000 0.1293661 0.64
12.17038540 7.7619664 9.86
0.00000000 0.0000000 0.42
0.00000000 0.0000000 0.42
13.49397590 21.7335058 24.90
0.00000000 0.0000000 0.44
10.77441077 4.1397154 5.94
0.00000000 0.0000000 0.50
14.28571429 52.5226391 56.84
13.58350951 33.2470893 37.84
0.00000000 0.0000000 0.66
0.00000000 0.0000000 0.84
4.30107527 0.5174644 1.86
10.67615658 7.7619664 11.24
1.58730159 0.1293661 1.26
11.91611058 16.1707633 20.98
9.30787589 5.0452781 8.38
12.18583397 20.6985770 26.26
0.00000000 0.0000000 1.52
12.07311207 32.4708926 41.58
0.00000000 0.0000000 2.42
0.00000000 0.0000000 2.46
6.69077758 4.7865459 11.06
9.94194485 17.7231565 27.56
9.29487179 15.0064683 24.96
0.00000000 0.0000000 3.94
7.76000000 12.5485123 25.00
0.00000000 0.0000000 6.42
4.64285714 5.0452781 16.80
0.00000000 0.0000000 8.58
0.00000000 0.0000000 9.26
0.00000000 0.0000000 9.26
3.28000000 5.3040103 25.00
3.07692308 5.1746442 26.00
0.00000000 0.0000000 21.70
0.00000000 0.0000000 22.30
0.15974441 0.2587322 25.04
0.06261741 0.1293661 31.94
0.05213764 0.1293661 38.36
p.value v.test
1.617461e-249 33.737740

```

```

## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 2.371996e-206 30.653522
## year=f.Year-2017 5.759700e-139 25.094160
## aux_years_sell=f.years_sell-(3, 5] 4.434566e-101 21.343988
## aux_mpg=f.mpg-(53.3, 61.4] 2.904133e-48 14.597716
## claHC=claHCPC-3 3.626343e-44 13.939863
## mout=NoMOut 2.553771e-36 12.584989
## aux_years_sell=f.years_sell-(5, 6] 3.236744e-35 12.382842
## year=f.Year-2016 3.236744e-35 12.382842
## engineSize=f.EngSize-1.2 9.486287e-31 11.528425
## model=VW-Polo 3.988946e-27 10.786452
## engineSize=f.EngSize-1.4 8.908322e-18 8.587240
## aux_price=f.price-(14, 19.5] 1.113548e-14 7.725569
## aux_price=f.price-[0.65, 14] 1.334814e-14 7.702450
## transmission=f.Trans-Manual 2.729878e-10 6.313390
## aux_mpg=f.mpg-(61.4, 88.3] 5.115225e-10 6.215528
## engineSize=f.EngSize-1 3.920704e-09 5.887505
## manufacturer=VW 5.694044e-09 5.825504
## model=Audi-A1 8.756013e-05 3.922706
## model=VW-Up 3.001921e-04 3.615134
## engineSize=f.EngSize-2.1 9.322438e-04 3.310215
## fuelType=f.Fuel-Petrol 2.834284e-02 2.192508
## aux_tax=f.tax-(145, 580] 4.533316e-02 2.001550
## manufacturer=Mercedes 4.261749e-02 -2.027439
## year=f.Year-2012 4.087513e-02 -2.044794
## model=Audi-Q2 3.645996e-02 -2.091760
## model=Audi-Q7 3.592831e-02 -2.097738
## year=f.Year-2018 2.993163e-02 -2.170994
## year=f.Year-2011 2.917168e-02 -2.181159
## year=f.Year-2010 2.917168e-02 -2.181159
## aux_price=f.price-(19.5, 26] 2.549105e-02 -2.233879
## model=BMW-X2 2.464272e-02 -2.246958
## aux_tax=f.tax-[0, 125] 1.735642e-02 -2.379069
## model=VW-T-Cross 1.485162e-02 -2.435977
## fuelType=f.Fuel-Diesel 8.598250e-03 -2.627628
## transmission=f.Trans-SemiAuto 3.982427e-03 -2.879550
## model=BMW-X5 3.842619e-03 -2.890801
## year=f.Year-2009 or before 8.372495e-04 -3.340177
## model=Audi-Q5 7.810505e-04 -3.359424
## engineSize=f.EngSize-3 5.526711e-04 -3.453837
## model=VW-T-Roc 3.236648e-04 -3.595584
## manufacturer=BMW 2.591861e-04 -3.653008
## year=f.Year-2015 1.261954e-04 -3.833768
## transmission=f.Trans-Automatic 9.907158e-05 -3.892854
## engineSize=f.EngSize-1.3 2.576068e-06 -4.702015
## engineSize=f.EngSize-2 1.640134e-08 -5.646221
## year=f.Year-2013 1.140644e-09 -6.088372
## claHC=claHCPC-5 8.078108e-10 -6.143385
## engineSize=f.EngSize-1.5 4.416257e-11 -6.589394
## aux_mpg=f.mpg-(44.8, 53.3] 5.285358e-12 -6.897694
## aux_mileage=f.mileage-(6e+03, 1.65e+04] 3.760733e-13 -7.263905
## year=f.Year-2014 2.067961e-15 -7.937200
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05] 2.889192e-20 -9.222994
## year=f.Year-2020 5.382751e-25 -10.325872
## aux_years_sell=f.years_sell-(6, 23] 2.191259e-26 -10.628708
## claHC=claHCPC-7 1.424301e-33 -12.075423
## claHC=claHCPC-8 2.553771e-36 -12.584989
## mout=YesMOut 2.553771e-36 -12.584989
## aux_price=f.price-(26, 110] 1.812783e-54 -15.541642
## aux_mpg=f.mpg-[8.8, 44.8] 2.125306e-59 -16.253048
## claHC=claHCPC-2 5.279934e-91 -20.230468
## claHC=claHCPC-1 6.997160e-94 -20.554623
## aux_mileage=f.mileage-[4, 6e+03] 1.278290e-102 -21.509149
## year=f.Year-2019 3.338700e-141 -25.298254
## aux_years_sell=f.years_sell-[2, 3] 4.566760e-179 -28.532993

```

```

## 
## $ '6 '
## 
## aux_years_sell=f.years_sell-(5, 6]          Cla/Mod   Mod/Cla Global
## year=f.Year-2016                            41.9953596 50.4178273 17.24
## claHC=claHCPC-6                           41.9953596 50.4178273 17.24
## aux_tax=f.tax-[0, 125]                      69.2015209 25.3481894 5.26
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05] 60.9427609 25.2089136 5.94
## claHC=claHCPC-4                           30.0800000 52.3676880 25.00
## aux_mpg=f.mpg-(61.4, 88.3]                 33.3684211 44.1504178 19.00
## aux_price=f.price-(14, 19.5]                30.7760141 48.6072423 22.68
## fuelType=f.Fuel-Diesel                     28.6516854 49.7214485 24.92
## year=f.Year-2015                           20.0211119 79.2479109 56.84
## mout=NoMOout                            38.9021480 22.7019499 8.38
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 15.8254353 100.0000000 90.74
## claHC=claHCPC-7                           25.2000000 43.8718663 25.00
## engineSize=f.EngSize-2.1                  35.1981352 21.0306407 8.58
## aux_years_sell=f.years_sell-(6, 23]          35.6265356 20.1949861 8.14
## aux_price=f.price-[0.65, 14]                24.0476190 28.1337047 16.80
## aux_mpg=f.mpg-(53.3, 61.4]                 21.4455917 37.6044568 25.18
## transmission=f.Trans-Manual               21.3804714 35.3760446 23.76
## model=Mercedes-GLA Class                  18.8857939 47.2144847 35.90
## model=Mercedes-X1                         36.9047619 4.3175487 1.68
## model=Mercedes-GL Class                  33.7837838 3.4818942 1.48
## manufacturer=Mercedes                     57.1428571 1.1142061 0.28
## model=BMW-1 Series                       17.3205033 32.5905292 27.02
## model=VW-Jetta                           22.8426396 6.2674095 3.94
## model=Audi-No                            100.0000000 0.4178273 0.06
## engineSize=f.EngSize-1.6                  19.5467422 9.6100279 7.06
## model=Mercedes-A Class                  20.3703704 7.6601671 5.40
## Audi=Audi No                           14.9936143 81.7548747 78.30
## year=f.Year-2017                          17.0236753 21.0306407 17.74
## model=Mercedes-CL Class                  24.5614035 1.9498607 1.14
## model=Mercedes-GLC Class                 21.0084034 3.4818942 2.38
## model=Audi-Q5                           7.5268817 0.9749304 1.86
## model=BMW-X5                            3.0303030 0.1392758 0.66
## fuelType=f.Fuel-Hybrid                   5.4545455 0.4178273 1.10
## year=f.Year-2011                          0.0000000 0.0000000 0.42
## year=f.Year-2010                          0.0000000 0.0000000 0.42
## manufacturer=VW                           12.8052805 27.0194986 30.30
## model=BMW-X2                            0.0000000 0.0000000 0.44
## model=VW-T-Cross                         0.0000000 0.0000000 0.50
## model=VW-Arteon                          0.0000000 0.0000000 0.54
## Audi=Audi Yes                           12.0737327 18.2451253 21.70
## manufacturer=Audi                         12.0737327 18.2451253 21.70
## engineSize=f.EngSize-4.2 or more          0.0000000 0.0000000 0.56
## engineSize=f.EngSize-4                  0.0000000 0.0000000 0.56
## model=Mercedes-GLE Class                3.6363636 0.2785515 1.10
## model=Audi-Q7                           0.0000000 0.0000000 0.64
## year=f.Year-2009 or before              0.0000000 0.0000000 0.84
## model=Audi-Q2                           1.2500000 0.1392758 1.60
## model=VW-T-Roc                           0.0000000 0.0000000 1.26
## aux_years_sell=f.years_sell-(3, 5]          11.1594203 21.4484680 27.60
## year=f.Year-2013                          2.4793388 0.4178273 2.42
## engineSize=f.EngSize-1.3                  0.0000000 0.0000000 1.52
## engineSize=f.EngSize-1                  6.3882064 3.6211699 8.14
## transmission=f.Trans-SemiAuto           11.0465116 29.1086351 37.84
## claHC=claHCPC-5                           0.0000000 0.0000000 2.46
## engineSize=f.EngSize-3                  6.7615658 5.2924791 11.24
## aux_price=f.price-(19.5, 26]              6.5060241 11.2813370 24.90
## year=f.Year-2020                          0.0000000 0.0000000 6.42
## aux_mpg=f.mpg-(44.8, 53.3]                6.7489115 12.9526462 27.56
## year=f.Year-2018                          0.6085193 0.4178273 9.86
## aux_tax=f.tax-(125, 145]                  9.7773475 42.2005571 61.98
## claHC=claHCPC-8                           0.0000000 0.0000000 9.26

```

## mout=YesMOut	0. 0000000	0. 0000000	9. 26
## fuelType=f. Fuel-Petrol	6. 8866571	20. 0557103	41. 82
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]	2. 1634615	3. 7604457	24. 96
## aux_mpg=f. mpg-[8. 8, 44. 8]	1. 6923077	3. 0640669	26. 00
## aux_price=f. price-(26, 110]	0. 8000000	1. 3927577	25. 00
## claHC=claHCPC-2	0. 0000000	0. 0000000	21. 70
## claHC=claHCPC-1	0. 0000000	0. 0000000	22. 30
## aux_mileage=f. mileage-[4, 6e+03]	0. 0000000	0. 0000000	25. 04
## year=f. Year-2019	0. 0000000	0. 0000000	31. 94
## aux_years_sell=f. years_sell-[2, 3]	0. 0000000	0. 0000000	38. 36
##	p. value	v. test	
## aux_years_sell=f. years_sell-(5, 6]	8. 691034e-114	22. 670845	
## year=f. Year-2016	8. 691034e-114	22. 670845	
## claHC=claHCPC-6	1. 737909e-98	21. 063006	
## aux_tax=f. tax-[0, 125]	1. 218668e-83	19. 376496	
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]	3. 238225e-66	17. 188496	
## claHC=claHCPC-4	9. 493930e-65	16. 991494	
## aux_mpg=f. mpg-(61. 4, 88. 3]	1. 379169e-62	16. 696959	
## aux_price=f. price-(14, 19. 5]	3. 869470e-55	15. 640292	
## fuelType=f. Fuel-Diesel	6. 289805e-42	13. 566934	
## year=f. Year-2015	3. 678725e-39	13. 091580	
## mout=NoMOut	1. 441926e-33	12. 074411	
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]	3. 381328e-33	12. 004097	
## claHC=claHCPC-7	3. 442139e-30	11. 416921	
## engineSize=f. EngSize-2. 1	1. 424220e-29	11. 292802	
## aux_years_sell=f. years_sell-(6, 23]	1. 237768e-16	8. 279420	
## aux_price=f. price-[0. 65, 14]	1. 321879e-15	7. 992539	
## aux_mpg=f. mpg-(53. 3, 61. 4]	2. 745718e-14	7. 609776	
## transmission=f. Trans-Manual	1. 739455e-11	6. 726372	
## model=Mercedes-GLA Class	2. 466582e-07	5. 160221	
## model=BMW-X1	2. 383677e-05	4. 225541	
## model=Mercedes-GL Class	2. 593923e-04	3. 652804	
## manufacturer=Mercedes	3. 501408e-04	3. 575065	
## model=BMW-1 Series	1. 140457e-03	3. 253368	
## model=VW-Jetta	2. 950579e-03	2. 972840	
## engineSize=f. EngSize-1. 6	5. 523694e-03	2. 774792	
## model=Mercedes-A Class	5. 614217e-03	2. 769501	
## Audi=Audi No	1. 393252e-02	2. 458998	
## year=f. Year-2017	1. 418642e-02	2. 452508	
## model=Mercedes-CL Class	3. 998521e-02	2. 053902	
## model=Mercedes-GLC Class	4. 586719e-02	1. 996613	
## model=Audi-Q5	4. 695495e-02	-1. 986706	
## model=BMW-X5	4. 466727e-02	-2. 007774	
## fuelType=f. Fuel-Hybrid	4. 412896e-02	-2. 012864	
## year=f. Year-2011	3. 829501e-02	-2. 071683	
## year=f. Year-2010	3. 829501e-02	-2. 071683	
## manufacturer=VW	3. 758395e-02	-2. 079364	
## model=BMW-X2	3. 277266e-02	-2. 134857	
## model=VW-T-Cross	2. 053675e-02	-2. 316394	
## model=VW-Arteon	1. 503620e-02	-2. 431506	
## Audi=Audi Yes	1. 393252e-02	-2. 458998	
## manufacturer=Audi	1. 393252e-02	-2. 458998	
## engineSize=f. EngSize-4. 2 or more	1. 286528e-02	-2. 487477	
## engineSize=f. EngSize-4	1. 286528e-02	-2. 487477	
## model=Mercedes-GLE Class	1. 196062e-02	-2. 513304	
## model=Audi-Q7	6. 892807e-03	-2. 701979	
## year=f. Year-2009 or before	1. 444788e-03	-3. 185551	
## model=Audi-Q2	5. 788200e-05	-4. 021283	
## model=VW-T-Roc	5. 369399e-05	-4. 038937	
## aux_years_sell=f. years_sell-(3, 5]	4. 856318e-05	-4. 062436	
## year=f. Year-2013	1. 085192e-05	-4. 399463	
## engineSize=f. EngSize-1. 3	6. 941591e-06	-4. 495472	
## engineSize=f. EngSize-1	1. 892702e-07	-5. 209578	
## transmission=f. Trans-SemiAuto	1. 245606e-07	-5. 286674	

```

## claHC=claHCPC-5
## engineSize=f. EngSize-3
## aux_price=f. price-(19. 5, 26]
## year=f. Year-2020
## aux_mpg=f. mpg-(44. 8, 53. 3]
## year=f. Year-2018
## aux_tax=f. tax-(125, 145]
## claHC=claHCPC-8
## mout=YesMOut
## fuelType=f. Fuel-Petrol
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]
## aux_mpg=f. mpg-[8. 8, 44. 8]
## aux_price=f. price-(26, 110]
## claHC=claHCPC-2
## claHC=claHCPC-1
## aux_mileage=f. mileage-[4, 6e+03]
## year=f. Year-2019
## aux_years_sell=f. years_sell-[2, 3]
##
## $ '7 '
##
## claHC=claHCPC-7
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]
## aux_years_sell=f. years_sell-(6, 23]
## aux_price=f. price-[0. 65, 14]
## year=f. Year-2014
## aux_tax=f. tax-(145, 580]
## year=f. Year-2013
## year=f. Year-2015
## aux_mpg=f. mpg-(61. 4, 88. 3]
## fuelType=f. Fuel-Diesel
## mout=NoMOut
## transmission=f. Trans-Manual
## engineSize=f. EngSize-1. 6
## claHC=claHCPC-6
## model=BMW-1 Series
## aux_tax=f. tax-[0, 125]
## model=Audi-A3
## engineSize=f. EngSize-2. 1
## year=f. Year-2012
## model=VW-Passat
## model=BMW-3 Series
## model=Audi-A1
## model=BMW-6 Series
## Audi=Audi Yes
## manufacturer=Audi
## model=VW-CC
## Audi=Audi No
## year=f. Year-2009 or before
## model=Audi-Q5
## model=Mercedes-C Class
## model=Mercedes-GLE Class
## model=VW-T-Roc
## manufacturer=Mercedes
## engineSize=f. EngSize-1. 3
## engineSize=f. EngSize-3
## model=Audi-Q2
## engineSize=f. EngSize-1. 5
## model=Mercedes-GLC Class
## engineSize=f. EngSize-1
## claHC=claHCPC-5
## aux_mpg=f. mpg-(44. 8, 53. 3]
## year=f. Year-2020
## transmission=f. Trans-SemiAuto

```

	4. 054885e-09	-5. 881939
3. 729325e-09	-5. 895773	
1. 598754e-22	-9. 764515	
4. 023639e-23	-9. 903399	
7. 426252e-24	-10. 070943	
1. 154380e-30	-11. 511511	
3. 700223e-31	-11. 609209	
1. 441926e-33	-12. 074411	
1. 441926e-33	-12. 074411	
3. 561132e-40	-13. 267750	
2. 672666e-60	-16. 379652	
7. 926833e-70	-17. 664099	
1. 047479e-80	-19. 025567	
5. 675164e-84	-19. 415794	
1. 268838e-86	-19. 726829	
5. 048643e-99	-21. 121482	
1. 229076e-132	-24. 507490	
1. 460402e-167	-27. 590159	

	Cla/Mod	Mod/Cla	Global
64. 56876457	74. 2627346	8. 58	
28. 48000000	95. 4423592	25. 00	
35. 95238095	80. 9651475	16. 80	
21. 12787927	71. 3136729	25. 18	
56. 34517766	29. 7587131	3. 94	
16. 20947631	69. 7050938	32. 08	
60. 33057851	19. 5710456	2. 42	
26. 73031026	30. 0268097	8. 38	
16. 84303351	51. 2064343	22. 68	
10. 59113300	80. 6970509	56. 84	
8. 22129160	100. 0000000	90. 74	
11. 58774373	55. 7640751	35. 90	
16. 14730878	15. 2815013	7. 06	
17. 11026616	12. 0643432	5. 26	
17. 25888325	9. 1152815	3. 94	
15. 15151515	12. 0643432	5. 94	
15. 07537688	8. 0428954	3. 98	
11. 54791155	12. 6005362	8. 14	
31. 57894737	1. 6085791	0. 38	
13. 82978723	3. 4852547	1. 88	
11. 11111111	7. 2386059	4. 86	
12. 40875912	4. 5576408	2. 74	
30. 00000000	0. 8042895	0. 20	
8. 94009217	26. 0053619	21. 70	
8. 94009217	26. 0053619	21. 70	
27. 27272727	0. 8042895	0. 22	
7. 04980843	73. 9946381	78. 30	
0. 00000000	0. 0000000	0. 84	
2. 15053763	0. 5361930	1. 86	
4. 63215259	4. 5576408	7. 34	
0. 00000000	0. 0000000	1. 10	
0. 00000000	0. 0000000	1. 26	
5. 77350111	20. 9115282	27. 02	
0. 00000000	0. 0000000	1. 52	
4. 44839858	6. 7024129	11. 24	
0. 00000000	0. 0000000	1. 60	
3. 97830018	5. 8981233	11. 06	
0. 00000000	0. 0000000	2. 38	
2. 94840295	3. 2171582	8. 14	
0. 00000000	0. 0000000	2. 46	
4. 93468795	18. 2305630	27. 56	
0. 00000000	0. 0000000	6. 42	
4. 22832981	21. 4477212	37. 84	

## claHC=claHCPC-8	0. 00000000	0. 0000000	9. 26
## mout=YesMOut	0. 00000000	0. 0000000	9. 26
## year=f. Year-2018	0. 00000000	0. 0000000	9. 86
## claHC=claHCPC-4	1. 57894737	4. 0214477	19. 00
## aux_mpg=f. mpg-[8. 8, 44. 8]	1. 92307692	6. 7024129	26. 00
## fuelType=f. Fuel-Petrol	3. 25203252	18. 2305630	41. 82
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]	1. 36000000	4. 5576408	25. 00
## year=f. Year-2017	0. 11273957	0. 2680965	17. 74
## aux_price=f. price-(19. 5, 26]	0. 80321285	2. 6809651	24. 90
## claHC=claHCPC-2	0. 00000000	0. 0000000	21. 70
## claHC=claHCPC-1	0. 00000000	0. 0000000	22. 30
## aux_price=f. price-(26, 110]	0. 24000000	0. 8042895	25. 00
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]	0. 00000000	0. 0000000	24. 96
## aux_mileage=f. mileage-[4, 6e+03]	0. 00000000	0. 0000000	25. 04
## aux_years_sell=f. years_sell-(3, 5]	0. 07246377	0. 2680965	27. 60
## year=f. Year-2019	0. 00000000	0. 0000000	31. 94
## aux_tax=f. tax-(125, 145]	2. 19425621	18. 2305630	61. 98
## aux_years_sell=f. years_sell-[2, 3]	0. 00000000	0. 0000000	38. 36
	p. value	v. test	
## claHC=claHCPC-7	1. 166017e-254	34. 086580	
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]	2. 144778e-206	30. 656803	
## aux_years_sell=f. years_sell-(6, 23]	8. 459310e-184	28. 911906	
## aux_price=f. price-[0. 65, 14]	2. 761398e-85	19. 570467	
## year=f. Year-2014	6. 360082e-78	18. 686634	
## aux_tax=f. tax-(145, 580]	3. 248724e-54	15. 504213	
## year=f. Year-2013	3. 500240e-53	15. 350767	
## year=f. Year-2015	1. 170398e-37	12. 826148	
## aux_mpg=f. mpg-(61. 4, 88. 3]	3. 899697e-36	12. 551516	
## fuelType=f. Fuel-Diesel	6. 738721e-24	10. 080493	
## mout=NoMOut	4. 072547e-17	8. 410791	
## transmission=f. Trans-Manual	4. 858579e-16	8. 114982	
## engineSize=f. EngSize-1. 6	1. 076520e-08	5. 718210	
## claHC=claHCPC-6	8. 345514e-08	5. 359494	
## model=BMW-1 Series	3. 155896e-06	4. 660404	
## aux_tax=f. tax-[0, 125]	3. 165253e-06	4. 659794	
## model=Audi-A3	1. 886396e-04	3. 733764	
## engineSize=f. EngSize-2. 1	2. 069129e-03	3. 080126	
## year=f. Year-2012	2. 225386e-03	3. 058378	
## model=VW-Passat	3. 054078e-02	2. 163005	
## model=BMW-3 Series	3. 526198e-02	2. 105337	
## model=Audi-A1	3. 756744e-02	2. 079543	
## model=BMW-6 Series	3. 776305e-02	2. 077417	
## Audi=Audi Yes	3. 932414e-02	2. 060779	
## manufacturer=Audi	3. 932414e-02	2. 060779	
## model=VW-CC	4. 991869e-02	1. 960660	
## Audi=Audi No	3. 932414e-02	-2. 060779	
## year=f. Year-2009 or before	3. 799926e-02	-2. 074863	
## model=Audi-Q5	3. 180642e-02	-2. 146835	
## model=Mercedes-C Class	2. 521705e-02	-2. 238061	
## model=Mercedes-GLE Class	1. 372919e-02	-2. 464272	
## model=VW-T-Roc	7. 327630e-03	-2. 681576	
## manufacturer=Mercedes	4. 858145e-03	-2. 816292	
## engineSize=f. EngSize-1. 3	2. 635569e-03	-3. 007326	
## engineSize=f. EngSize-3	2. 355187e-03	-3. 041352	
## model=Audi-Q2	1. 923030e-03	-3. 101869	
## engineSize=f. EngSize-1. 5	4. 142375e-04	-3. 530843	
## model=Mercedes-GLC Class	8. 774561e-05	-3. 922196	
## engineSize=f. EngSize-1	6. 672803e-05	-3. 987661	
## claHC=claHCPC-5	6. 383834e-05	-3. 998154	
## aux_mpg=f. mpg-(44. 8, 53. 3]	1. 421468e-05	-4. 340519	
## year=f. Year-2020	6. 535990e-12	-6. 867449	
## transmission=f. Trans-SemiAuto	2. 003552e-12	-7. 034236	
## claHC=claHCPC-8	4. 072547e-17	-8. 410791	
## mout=YesMOut	4. 072547e-17	-8. 410791	

```

## year=f. Year-2018
## claHC=claHCPC-4
## aux_mpg=f. mpg-[8. 8, 44. 8]
## fuelType=f. Fuel-Petrol
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]
## year=f. Year-2017
## aux_price=f. price-(19. 5, 26]
## claHC=claHCPC-2
## claHC=claHCPC-1
## aux_price=f. price-(26, 110]
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]
## aux_mileage=f. mileage-[4, 6e+03]
## aux_years_sell=f. years_sell-(3, 5]
## year=f. Year-2019
## aux_tax=f. tax-(125, 145]
## aux_years_sell=f. years_sell-[2, 3]
##
## $ '8 '
##
## claHC=claHCPC-8
## mout=YesMOut
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05]
## aux_years_sell=f. years_sell-(6, 23]
## aux_tax=f. tax-(145, 580]
## aux_price=f. price-[0. 65, 14]
## year=f. Year-2009 or before
## year=f. Year-2011
## year=f. Year-2010
## engineSize=f. EngSize-4
## engineSize=f. EngSize-4. 2 or more
## transmission=f. Trans-Automatic
## engineSize=f. EngSize-3
## year=f. Year-2013
## fuelType=f. Fuel-Diesel
## model=Audi-Q8
## aux_mpg=f. mpg-[8. 8, 44. 8]
## year=f. Year-2012
## model=BMW-X7
## model=Audi-Q7
## year=f. Year-2015
## aux_mpg=f. mpg-(61. 4, 88. 3]
## manufacturer=BMW
## model=BMW-M5
## year=f. Year-2014
## model=BMW-8 Series
## model=BMW-i8
## model=Audi-RS6
## fuelType=f. Fuel-Hybrid
## engineSize=f. EngSize-1. 8
## model=Audi-A6
## model=BMW-X5
## model=Mercedes-M Class
## Audi=Audi Yes
## manufacturer=Audi
## model=Audi-SQ5
## model=Mercedes-SL CLASS
## model=Mercedes-CLC Class
## model=Audi-R8
## engineSize=f. EngSize-1. 6
## engineSize=f. EngSize-2. 5
## model=BMW-X3
## model=BMW-5 Series
## model=VW-Passat
## model=Audi-A4
3. 078771e-18 -8. 708517
9. 257139e-19 -8. 843735
3. 438861e-23 -9. 919087
8. 785181e-24 -10. 054404
9. 560380e-28 -10. 916998
8. 828911e-32 -11. 731114
1. 988650e-34 -12. 236312
4. 000788e-42 -13. 600065
1. 961284e-43 -13. 818868
8. 321861e-44 -13. 880451
2. 267259e-49 -14. 770588
1. 491166e-49 -14. 798801
2. 786072e-53 -15. 365563
4. 510084e-66 -17. 169276
1. 530673e-72 -18. 013354
4. 022746e-83 -19. 314930

          Cla/Mod      Mod/Cla Global
100. 000000 100. 0000000 9. 26
100. 000000 100. 0000000 9. 26
27. 760000 74. 9460043 25. 00
27. 857143 50. 5399568 16. 80
19. 763092 68. 4665227 32. 08
21. 604448 58. 7473002 25. 18
100. 000000 9. 0712743 0. 84
100. 000000 4. 5356371 0. 42
100. 000000 4. 5356371 0. 42
82. 142857 4. 9676026 0. 56
75. 000000 4. 5356371 0. 56
14. 166032 40. 1727862 26. 26
17. 437722 21. 1663067 11. 24
28. 925620 7. 5593952 2. 42
11. 365236 69. 7624190 56. 84
100. 000000 1. 7278618 0. 16
13. 384615 37. 5809935 26. 00
63. 157895 2. 5917927 0. 38
100. 000000 1. 5118790 0. 14
43. 750000 3. 0237581 0. 64
16. 229117 14. 6868251 8. 38
12. 962963 31. 7494600 22. 68
13. 060057 29. 5896328 20. 98
100. 000000 0. 8639309 0. 08
17. 766497 7. 5593952 3. 94
80. 000000 0. 8639309 0. 10
100. 000000 0. 6479482 0. 06
100. 000000 0. 6479482 0. 06
23. 636364 2. 8077754 1. 10
30. 769231 1. 7278618 0. 52
21. 126761 3. 2397408 1. 42
27. 272727 1. 9438445 0. 66
50. 000000 0. 8639309 0. 16
11. 520737 26. 9978402 21. 70
11. 520737 26. 9978402 21. 70
60. 000000 0. 6479482 0. 10
23. 684211 1. 9438445 0. 76
100. 000000 0. 4319654 0. 04
100. 000000 0. 4319654 0. 04
13. 314448 10. 1511879 7. 06
40. 000000 0. 8639309 0. 20
20. 833333 2. 1598272 0. 96
17. 441860 3. 2397408 1. 72
17. 021277 3. 4557235 1. 88
15. 441176 4. 5356371 2. 72

```

## model=BMW-X6	42.857143	0.6479482	0.14	
## model=BMW-3 Series	13.580247	7.1274298	4.86	
## model=Audi-RS5	66.666667	0.4319654	0.06	
## year=f. Year-2020	6.230530	4.3196544	6.42	
## model=VW-Up	3.409091	0.6479482	1.76	
## engineSize=f. EngSize-1.4	5.882353	3.6717063	5.78	
## aux_price=f. price-(26, 110]	7.760000	20.9503240	25.00	
## model=Mercedes-CL Class	1.754386	0.2159827	1.14	
## model=VW-Golf	6.557377	6.9114471	9.76	
## model=VW-Tiguan	4.268293	1.5118790	3.28	
## engineSize=f. EngSize-1.2	3.200000	0.8639309	2.50	
## Audi=Audi No	8.633461	73.0021598	78.30	
## model=Audi-Q2	1.250000	0.2159827	1.60	
## engineSize=f. EngSize-2	7.840308	35.2051836	41.58	
## model=VW-T-Roc	0.000000	0.0000000	1.26	
## year=f. Year-2017	6.538895	12.5269978	17.74	
## engineSize=f. EngSize-1.3	0.000000	0.0000000	1.52	
## model=Audi-Q3	1.408451	0.4319654	2.84	
## aux_mpg=f. mpg-(53.3, 61.4]	6.397306	16.4146868	23.76	
## claHC=claHCPC-5	0.000000	0.0000000	2.46	
## model=VW-Polo	2.873563	2.1598272	6.96	
## manufacturer=VW	6.006601	19.6544276	30.30	
## year=f. Year-2018	3.245436	3.4557235	9.86	
## engineSize=f. EngSize-1.5	3.254973	3.8876890	11.06	
## aux_price=f. price-(14, 19.5]	5.296950	14.2548596	24.92	
## transmission=f. Trans-SemiAuto	6.183932	25.2699784	37.84	
## aux_years_sell=f. years_sell-(3, 5]	5.362319	15.9827214	27.60	
## engineSize=f. EngSize-1	1.474201	1.2958963	8.14	
## fuelType=f. Fuel-Petrol	5.978001	26.9978402	41.82	
## claHC=claHCPC-6	0.000000	0.0000000	5.26	
## aux_mpg=f. mpg-(44.8, 53.3]	4.789550	14.2548596	27.56	
## aux_mileage=f. mileage-[4, 6e+03]	4.073482	11.0151188	25.04	
## claHC=claHCPC-7	0.000000	0.0000000	8.58	
## aux_mileage=f. mileage-(6e+03, 1.65e+04]	2.724359	7.3434125	24.96	
## claHC=claHCPC-3	0.000000	0.0000000	11.44	
## aux_mileage=f. mileage-(1.65e+04, 3.35e+04]	2.480000	6.6954644	25.00	
## aux_price=f. price-(19.5, 26]	2.248996	6.0475162	24.90	
## year=f. Year-2019	2.504696	8.6393089	31.94	
## aux_years_sell=f. years_sell-[2, 3]	3.128259	12.9589633	38.36	
## claHC=claHCPC-4	0.000000	0.0000000	19.00	
## claHC=claHCPC-2	0.000000	0.0000000	21.70	
## claHC=claHCPC-1	0.000000	0.0000000	22.30	
## aux_tax=f. tax-(125, 145]	3.581801	23.9740821	61.98	
## mout=NoMOut	0.000000	0.0000000	90.74	
##		p. value	v. test	
## claHC=claHCPC-8	0.000000e+00		Inf	
## mout=YesMOut	0.000000e+00		Inf	
## aux_mileage=f. mileage-(3.35e+04, 1.16e+05]	1.554868e-126	23.928299		
## aux_years_sell=f. years_sell-(6, 23]	4.901088e-71	17.820484		
## aux_tax=f. tax-(145, 580]	5.062511e-64	16.893041		
## aux_price=f. price-[0.65, 14]	5.767433e-59	16.191740		
## year=f. Year-2009 or before	6.914535e-45	14.057649		
## year=f. Year-2011	1.309612e-22	9.784717		
## year=f. Year-2010	1.309612e-22	9.784717		
## engineSize=f. EngSize-4	6.632477e-20	9.133492		
## engineSize=f. EngSize-4.2 or more	8.600628e-17	8.322666		
## transmission=f. Trans-Automatic	6.897170e-12	6.859770		
## engineSize=f. EngSize-3	7.943877e-11	6.501659		
## year=f. Year-2013	4.202440e-10	6.246317		
## fuelType=f. Fuel-Diesel	2.200919e-09	5.982238		
## model=Audi-Q8	5.115807e-09	5.843361		
## aux_mpg=f. mpg-[8.8, 44.8]	8.579695e-09	5.756655		
## year=f. Year-2012	1.006358e-08	5.729654		
## model=BMW-X7	5.601584e-08	5.431074		

```

## model=Audi-Q7
## year=f. Year-2015
## aux_mpg=f. mpg-(61. 4, 88. 3]
## manufacturer=BMW
## model=BMW-M5
## year=f. Year-2014
## model=BMW-8 Series
## model=BMW-i8
## model=Audi-RS6
## fuelType=f. Fuel-Hybrid
## engineSize=f. EngSize-1.8
## model=Audi-A6
## model=BMW-X5
## model=Mercedes-M Class
## Audi=Audi Yes
## manufacturer=Audi
## model=Audi-SQ5
## model=Mercedes-SL CLASS
## model=Mercedes-CLC Class
## model=Audi-R8
## engineSize=f. EngSize-1.6
## engineSize=f. EngSize-2.5
## model=BMW-X3
## model=BMW-5 Series
## model=VW-Passat
## model=Audi-A4
## model=BMW-X6
## model=BMW-3 Series
## model=Audi-RS5
## year=f. Year-2020
## model=VW-Up
## engineSize=f. EngSize-1.4
## aux_price=f. price-(26, 110]
## model=Mercedes-CL Class
## model=VW-Golf
## model=VW-Tiguan
## engineSize=f. EngSize-1.2
## Audi=Audi No
## model=Audi-Q2
## engineSize=f. EngSize-2
## model=VW-T-Roc
## year=f. Year-2017
## engineSize=f. EngSize-1.3
## model=Audi-Q3
## aux_mpg=f. mpg-(53. 3, 61. 4]
## claHC=claHCPC-5
## model=VW-Polo
## manufacturer=VW
## year=f. Year-2018
## engineSize=f. EngSize-1.5
## aux_price=f. price-(14, 19.5]
## transmission=f. Trans-SemiAuto
## aux_years_sell=f. years_sell-(3, 5]
## engineSize=f. EngSize-1
## fuelType=f. Fuel-Petrol
## claHC=claHCPC-6
## aux_mpg=f. mpg-(44. 8, 53. 3]
## aux_mileage=f. mileage-[4, 6e+03]
## claHC=claHCPC-7
## aux_mileage=f. mileage-(6e+03, 1. 65e+04]
## claHC=claHCPC-3
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04]
## aux_price=f. price-(19. 5, 26]
## year=f. Year-2019
3. 098228e-07 5. 117372
2. 091759e-06 4. 744351
2. 343987e-06 4. 721251
4. 253532e-06 4. 598593
7. 266462e-05 3. 967388
1. 422709e-04 3. 804187
3. 432952e-04 3. 580227
7. 893589e-04 3. 356499
7. 893589e-04 3. 356499
1. 539956e-03 3. 167047
2. 115019e-03 3. 073587
2. 351547e-03 3. 041817
2. 911931e-03 2. 976885
4. 048421e-03 2. 874364
4. 528525e-03 2. 838788
4. 528525e-03 2. 838788
7. 180295e-03 2. 688365
8. 421713e-03 2. 634678
8. 557952e-03 2. 629226
8. 557952e-03 2. 629226
9. 398039e-03 2. 597225
1. 082174e-02 2. 548403
1. 462839e-02 2. 441450
1. 638753e-02 2. 400168
1. 670834e-02 2. 393064
1. 911364e-02 2. 343307
2. 282115e-02 2. 276416
2. 361209e-02 2. 263383
2. 488450e-02 2. 243191
4. 530707e-02 -2. 001792
4. 031697e-02 -2. 050487
3. 364805e-02 -2. 124264
3. 250104e-02 -2. 138194
3. 001385e-02 -2. 169908
2. 525909e-02 -2. 237417
1. 623524e-02 -2. 403584
9. 418223e-03 -2. 596488
4. 528525e-03 -2. 838788
4. 062357e-03 -2. 873279
3. 294183e-03 -2. 938863
2. 107941e-03 -3. 074587
1. 438391e-03 -3. 186834
5. 850247e-04 -3. 438464
1. 178440e-04 -3. 850568
5. 578170e-05 -4. 029980
5. 515425e-06 -4. 544162
1. 482869e-06 -4. 813531
6. 133813e-08 -5. 414856
5. 937028e-08 -5. 420687
8. 133131e-09 -5. 765676
4. 645667e-09 -5. 859391
1. 998420e-09 -5. 997935
7. 634609e-10 -6. 152344
1. 688770e-11 -6. 730676
4. 072047e-12 -6. 934658
3. 836788e-12 -6. 943066
1. 077617e-12 -7. 120211
3. 386259e-15 -7. 875779
1. 070250e-19 -9. 081565
1. 424402e-24 -10. 232071
1. 931360e-26 -10. 640478
1. 025706e-26 -10. 699281
1. 077181e-28 -11. 113607
1. 216673e-35 -12. 461109

```

```

## aux_years_sell=f.years_sell-[2, 3] 1. 127852e-36 -12. 649363
## claHC=claHCPC-4 1. 934579e-45 -14. 147521
## claHC=claHCPC-2 1. 098165e-52 -15. 276416
## claHC=claHCPC-1 2. 479313e-54 -15. 521565
## aux_tax=f.tax-(125, 145] 1. 927642e-68 -17. 483090
## mout=NoMOut 0. 000000e+00 -Inf
##
##
## Link between the cluster variable and the quantitative variables
## =====
##          Eta2 P-value
## price      0.2982498 0
## mileage    0.6964806 0
## tax        0.6185937 0
## mpg        0.4436024 0
## years_sell 0.6366584 0
## totalMOE   0.3492559 0
##
## Description of each cluster by quantitative variables
## =====
## $ '1 '
##          v. test Mean in category Overall mean sd in category Overall sd
## mpg        5.707897 5.498795e+01 52.73678 9.1451284 1.149277e+01
## tax        -2.075033 1.461651e+02 146.98985 3.2692157 1.158247e+01
## years_sell -8.027318 4.196970e+00 4.78660 0.8486201 2.140435e+00
## mileage    -10.788987 1.494471e+04 22887.97553 7106.6113628 2.145413e+04
## totalMOE   -13.273108 6.749311e-02 0.28340 0.2508741 4.740089e-01
##          p. value
## mpg        1.143805e-08
## tax        3.798346e-02
## years_sell 9.962695e-16
## mileage   3.880427e-27##
## totalMOE  3.315410e-40##
## $ '2 '
##          v. test Mean in category Overall mean sd in category Overall sd
## price      29.011064 3.220756e+04 21470.24440 9252.6565545 1.084587e+04
## tax        -3.074651 1.457746e+02 146.98985 2.0564225 1.158247e+01
## totalMOE   -15.664948 3.001364e-02 0.28340 0.1706248 4.740089e-01
## mileage    -25.817761 3.986430e+03 22887.97553 3441.3981207 2.145413e+04
## years_sell -28.475797 2.706685e+00 4.78660 0.5063542 2.140435e+00
## mpg        -36.643573 3.836568e+01 52.73678 5.8090831 1.149277e+01
##          p. value
## price      4.771613e-185
## tax        2.107489e-03##
## totalMOE   2.626434e-55
## mileage   5.602856e-147##
## years_sell 2.336322e-178##
## mpg        5.792301e-294##
## $ '3 '
##          v. test Mean in category Overall mean sd in category Overall sd
## price      11.671565 2.486124e+04 21470.24440 7293.6990803 1.084587e+04
## tax        -5.277011 1.453526e+02 146.98985 1.7826732 1.158247e+01
## mpg        -11.381390 4.923287e+01 52.73678 5.9034624 1.149277e+01
## totalMOE   -21.379976 1.192661e-02 0.28340 0.1085558 4.740089e-01
## mileage    -28.877313 6.292056e+03 22887.97553 4570.4458112 2.145413e+04
## years_sell -30.359561 3.045872e+00 4.78660 0.4237301 2.140435e+00
##          p. value
## price      1.781158e-31
## tax        1.313083e-07
## mpg        5.176796e-30
## totalMOE   2.052350e-101
## mileage   2.301216e-183

```

```

## years_sell 1.878962e-202
##
## $ '4 '
##          v. test Mean in category Overall mean sd in category Overall sd
## tax      52.523839    2.009455e+02    146.98985    4.914306e+00 1.158247e+01
## years_sell 9.705326    6.629032e+00    4.78660    1.160282e+00 2.140435e+00
## mileage   6.627876    3.549942e+04   22887.97553    1.329314e+04 2.145413e+04
## totalMOE -6.549321    8.064516e-03    0.28340    8.943981e-02 4.740089e-01
## mpg      -8.236646    4.434113e+01    52.73678    3.828775e+00 1.149277e+01
##          p. value
## tax      0.000000e+00
## years_sell 2.861605e-22
## mileage   3.405509e-11
## totalMOE 5.779944e-11
## mpg      1.771051e-16
##
## $ '5 '
##          v. test Mean in category Overall mean sd in category Overall sd
## mpg      16.182267    5.888783e+01    52.73678    7.403577e+00 1.149277e+01
## totalMOE 9.731508    4.359638e-01    0.28340    0.5138196 4.740089e-01
## years_sell 8.094620    5.359638e+00    4.78660    0.7029974 2.140435e+00
## tax      -4.317522    1.453359e+02    146.98985    5.6585490 1.158247e+01
## price   -13.762222    1.653353e+04   21470.24440    5581.1974101 1.084587e+04
##          p. value
## mpg      6.727083e-59
## totalMOE 2.212886e-22
## years_sell 5.744333e-16
## tax      1.577907e-05
## price   4.301035e-43
##
## $ '6 '
##          v. test Mean in category Overall mean sd in category Overall sd
## mpg      22.79082    6.178377e+01    52.73678    8.0810658 1.149277e+01
## years_sell 18.05177    6.121170e+00    4.78660    0.8231151 2.140435e+00
## totalMOE 17.82348    5.752089e-01    0.28340    0.5109371 4.740089e-01
## mileage   15.26178    3.419728e+04   22887.97553    8964.2957270 2.145413e+04
## price   -16.27041    1.537512e+04   21470.24440    4106.3903423 1.084587e+04
## tax      -16.40878    1.404254e+02    146.98985    9.5662194 1.158247e+01
##          p. value
## mpg      5.655160e-115
## years_sell 7.640095e-73
## totalMOE 4.645423e-71
## mileage   1.374451e-52
## price   1.600768e-59
## tax      1.654976e-60
##
## $ '7 '
##          v. test Mean in category Overall mean sd in category Overall sd
## mileage  26.10795    5.079012e+04   22887.97553    9825.0311877 2.145413e+04
## years_sell 25.86314    7.544236e+00    4.78660    1.0617908 2.140435e+00
## totalMOE 16.83713    6.809651e-01    0.28340    0.4885687 4.740089e-01
## mpg      14.81617    6.121910e+01    52.73678    10.4800946 1.149277e+01
## price   -17.33829    1.210273e+04   21470.24440    3841.9262489 1.084587e+04
##          p. value
## mileage  2.961352e-150
## years_sell 1.731510e-147
## totalMOE 1.304120e-63
## mpg      1.151664e-49
## price   2.418448e-67
##
## $ '8 '
##          v. test Mean in category Overall mean sd in category Overall sd
## mileage  37.630812    5.863228e+04   22887.97553    31918.640845 2.145413e+04
## years_sell 27.162493    7.360691e+00    4.78660    3.588067 2.140435e+00

```

```

## totalMOE 20.252281 7.084233e-01 0.28340 0.605299 4.740089e-01
## tax 11.573894 1.529250e+02 146.98985 15.573121 1.158247e+01
## mpg -2.777691 5.132340e+01 52.73678 15.551143 1.149277e+01
## p.value
## mileage 0.000000e+00
## years_sell 1.802722e-162
## totalMOE 3.391667e-91
## tax 5.588725e-31
## mpg 5.474665e-03

```

We start with the description of the categorical variables that characterize the clusters:

Cluster 1: Cluster one has cars with medium-few years\_sell, medium taxes, medium mileage, medium price and medium mpg. Also, non-Audi cars predominate.

Cluster 2: Cluster two has cars with low mpg, few years\_sell, low mileage, high price and medium taxes. Also, cars with petrol as fuel, cars with semiAuto transmission, cars with engineSize 2 and non-Audi cars predominate.

Cluster 3: Cluster three has cars with few years\_sell, medium taxes, low mileage, high price and medium-low mpg. Also, non-Audi cars and cars with diesel as fuel predominate.

Cluster 4: Cluster four has cars with high taxes, many years\_sell, high mileage and low mpg. Also, non-Audi cars and cars with engineSize 2 predominate.

Cluster 5: Cluster five has cars with medium years\_sell, high mpg, low price and medium-high mileage. Also, cars with diesel as fuel predominate.

Cluster 6: Cluster six has cars with high mileage and many years\_sell. Also, cars with diesel as fuel and non-Audi cars predominate.

Cluster 7: Cluster seven has cars with high mileage, many years\_sell, low price, high taxes and high mpg. Also, cars with diesel as fuel and cars with manual transmission predominate.

Cluster 8: Cluster of multivariate outliers.

Regarding quantitative variables:

Cluster 1: As we can see mpg mean is greater in cluster 1 than in the overall mean. Also, we can see that tax, years\_sell and mileage means are lower in cluster 1 than in the overall means. So we can say that cluster 1 is a cluster of cars with high mpg, low tax, few years\_sell and low mileage.

Cluster 2: As we can see price mean is greater in cluster 2 than in the overall mean. Also, we can see that tax, mileage, years\_sell and mpg means are lower in cluster 2 than in the overall means. So we can say that cluster 2 is a cluster of expensive cars and low tax, low mileage, few years\_sell and low mpg.

Cluster 3: As we can see price mean is greater in cluster 3 than in the overall mean. Also, we can see that tax, mpg, mileage and years\_sell means are lower in cluster 3 than in the overall means. So we can say that cluster 3 is a cluster of expensive cars with low tax, low mpg, low mileage and few years\_sell.

Cluster 4: As we can see tax, years\_sell and mileage means are greater in cluster 4 than in the overall means. Also, we can see that mpg mean is lower in cluster 4 than in the overall mean. So we can say that cluster 4 is a cluster of cars with high tax, many years\_sell, high mileage and low mpg.

Cluster 5: As we can see mpg and years\_sell means are greater in cluster 5 than in the overall means. Also, we can see that tax and price means are lower in cluster 5 than in the overall means. So we can say that cluster 5 is a cluster of cheap cars with high mpg, many years\_sell and low tax.

Cluster 6: As we can see mpg, years\_sell and mileage means are greater in cluster 6 than in the overall means. Also, we can see that price and tax means are lower in cluster 6 than in the overall means. So we can say that cluster 6 is a cluster of cheap cars with high mpg, many years\_sell, high mileage and low tax.

Cluster 7: As we can see mileage, years\_sell and mpg means are greater in cluster 7 than in the overall means. Also, we can see that price mean is lower in cluster 7 than in the overall mean. So we can say that cluster 7 is a cluster of cheap cars with high mileage, many years\_sell and high mpg.

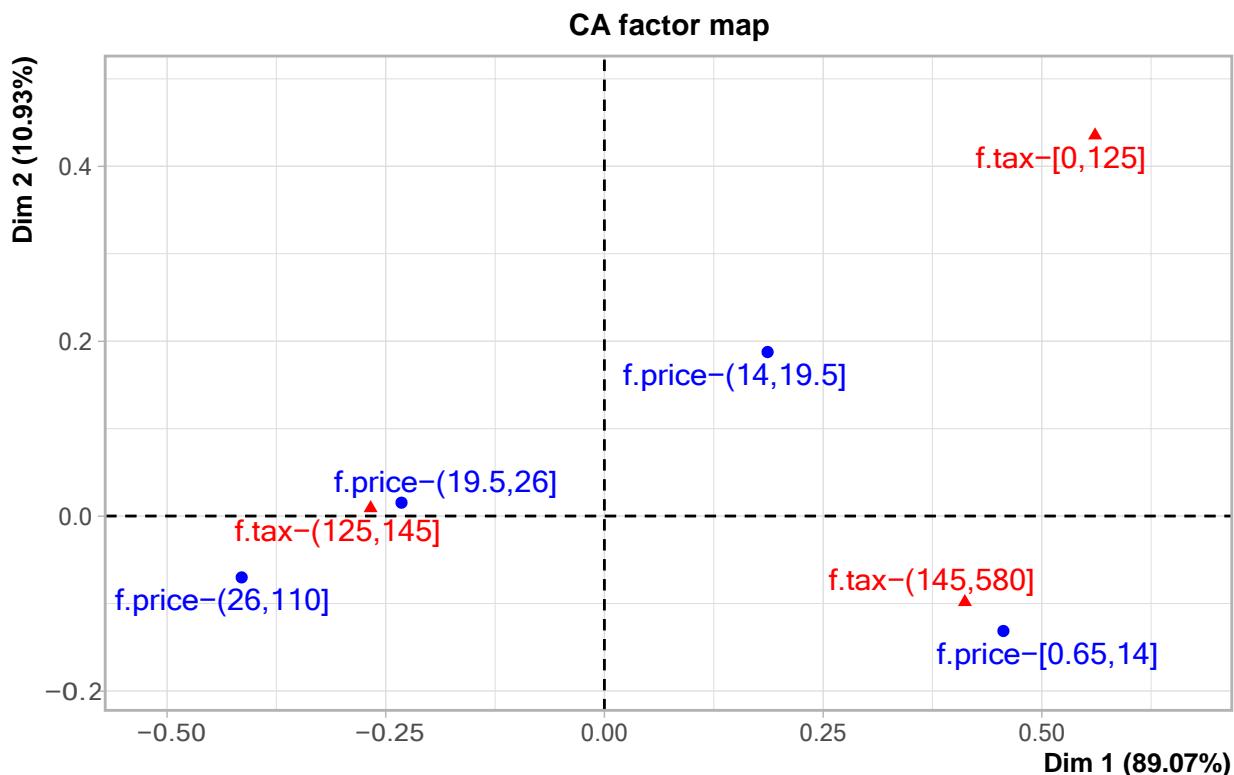
Cluster 8: Cluster of multivariate outliers.

## 5 CA

```
par(mfrow=c(1, 1))
tt<-table(df[, c("aux_price", "aux_tax")])
tt
```

```
## aux_tax
## aux_price f.tax-[0, 125] f.tax-(125, 145] f.tax-(145, 580]
## f.price-[0.65, 14] 95 495 669
## f.price-(14, 19.5] 147 671 428
## f.price-(19.5, 26] 50 912 283
## f.price-(26, 110] 5 1021 224
```

```
res.ca<-CA(tt)
```

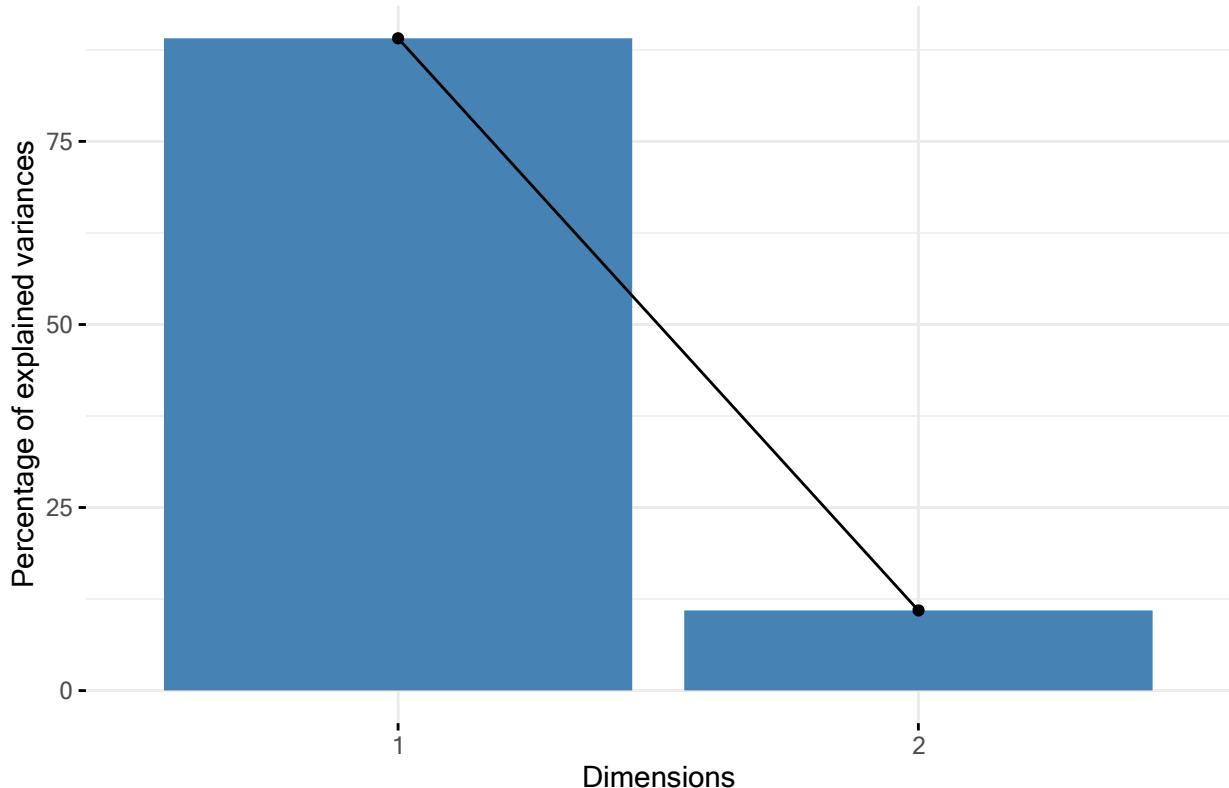


```
chisq.test(tt)
```

```
##
## Pearson' s Chi-squared test
##
## data: tt
## X-squared = 659.26, df = 6, p-value < 2.2e-16
```

```
fviz_eig(res.ca) #Same outputs as PCA
```

## Scree plot

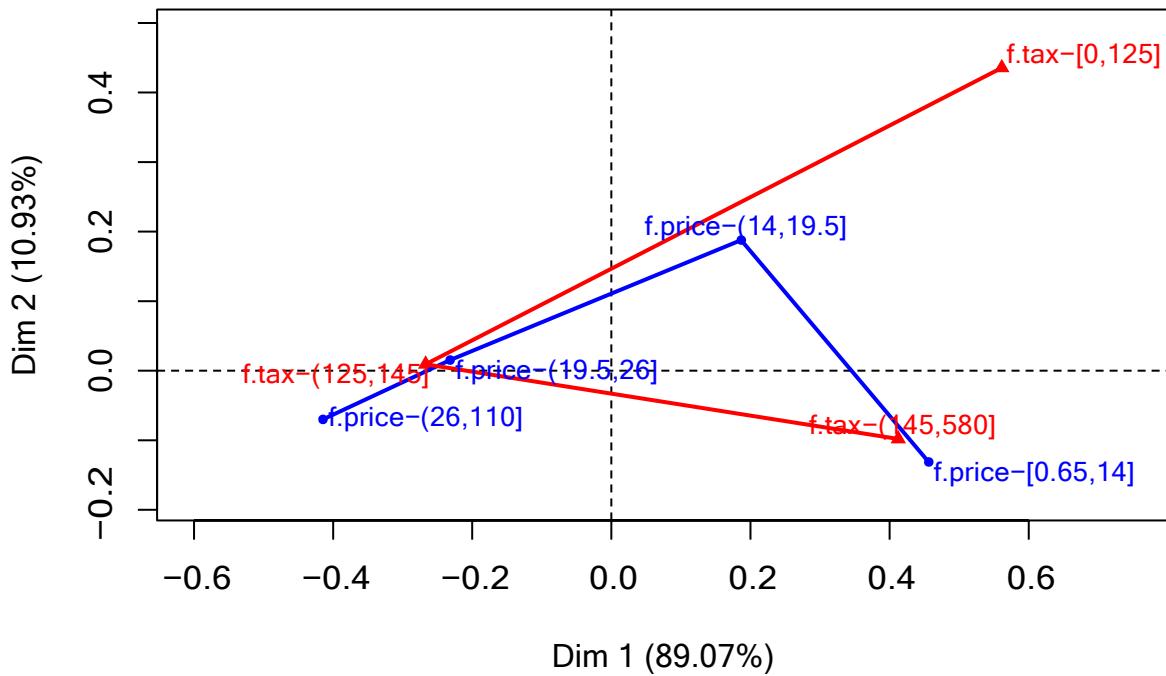


```
summary(res.ca, dig=2)
```

```
##  
## Call:  
## CA(X = tt)  
##  
## The chi square of independence between the two variables is equal to 659.2639 (p-value = 3.804822e-1)  
##  
## Eigenvalues  
##  
##          Dim.1   Dim.2  
## Variance 0.117  0.014  
## % of var. 89.071 10.929  
## Cumulative % of var. 89.071 100.000  
##  
## Rows  
##  
## f.price-[0,65] | 56.763 | 0.456 44.631 0.923 | -0.131 30.173 0.077 |  
## f.price-(65,145] | 17.452 | 0.187 7.385 0.497 | 0.188 60.917 0.503 |  
## f.price-(145,26] | 13.441 | -0.232 11.395 0.996 | 0.015 0.407 0.004 |  
## f.price-(26,110] | 44.197 | -0.415 36.589 0.972 | -0.070 8.503 0.028 |  
##  
## Columns  
##  
## f.tax-[0,125] | 29.957 | 0.561 15.920 0.624 | 0.435 78.140 0.376 |  
## f.tax-(125,145] | 44.281 | -0.267 37.661 0.999 | 0.009 0.359 0.001 |  
## f.tax-(145,580] | 57.614 | 0.412 46.419 0.946 | -0.098 21.501 0.054 |
```

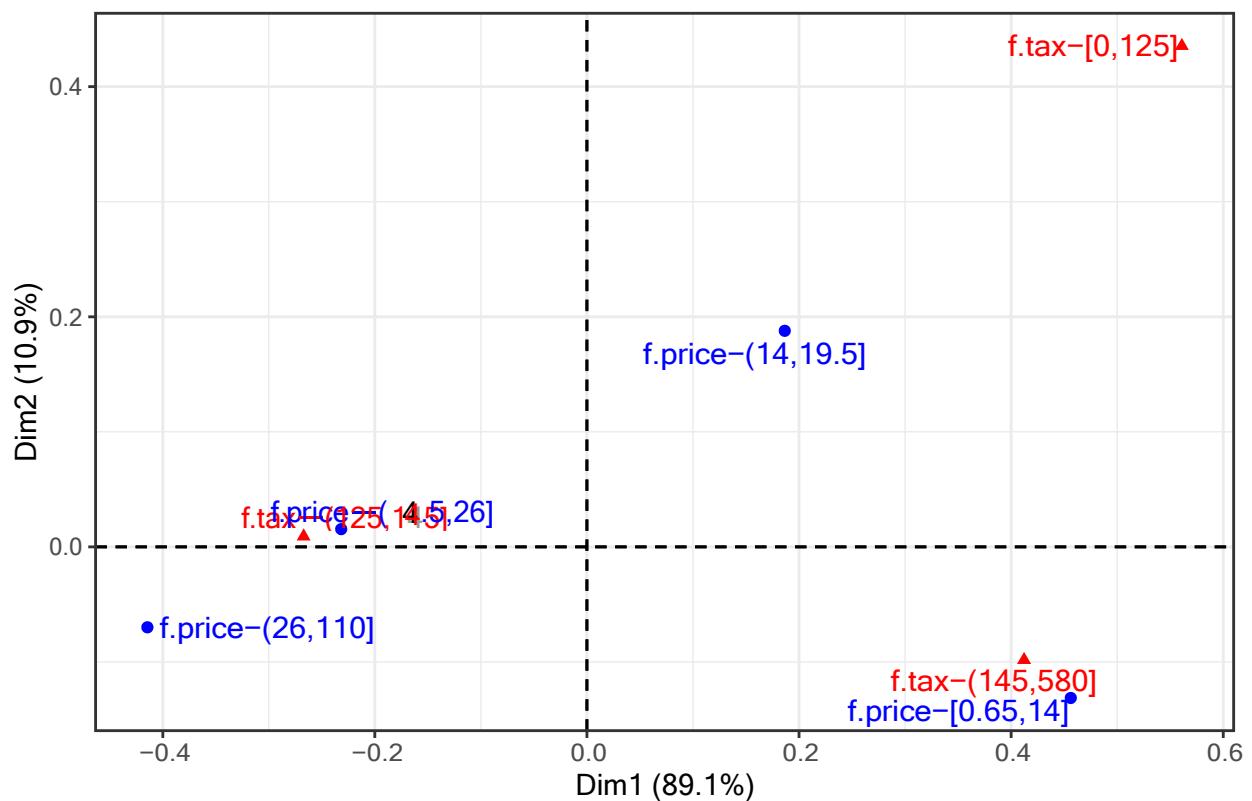
```
plot( res.ca, cex=0.8, graph.type = "classic" )  
lines( res.ca$row$coord[, 1], res.ca$row$coord[, 2], col="blue", lwd = 2 )  
lines( res.ca$col$coord[, 1], res.ca$col$coord[, 2], col="red", lwd = 2 )
```

## CA factor map



```
fviz_ca_biplot(res.ca, repel=TRUE)+theme_bw()
```

## CA – Biplot



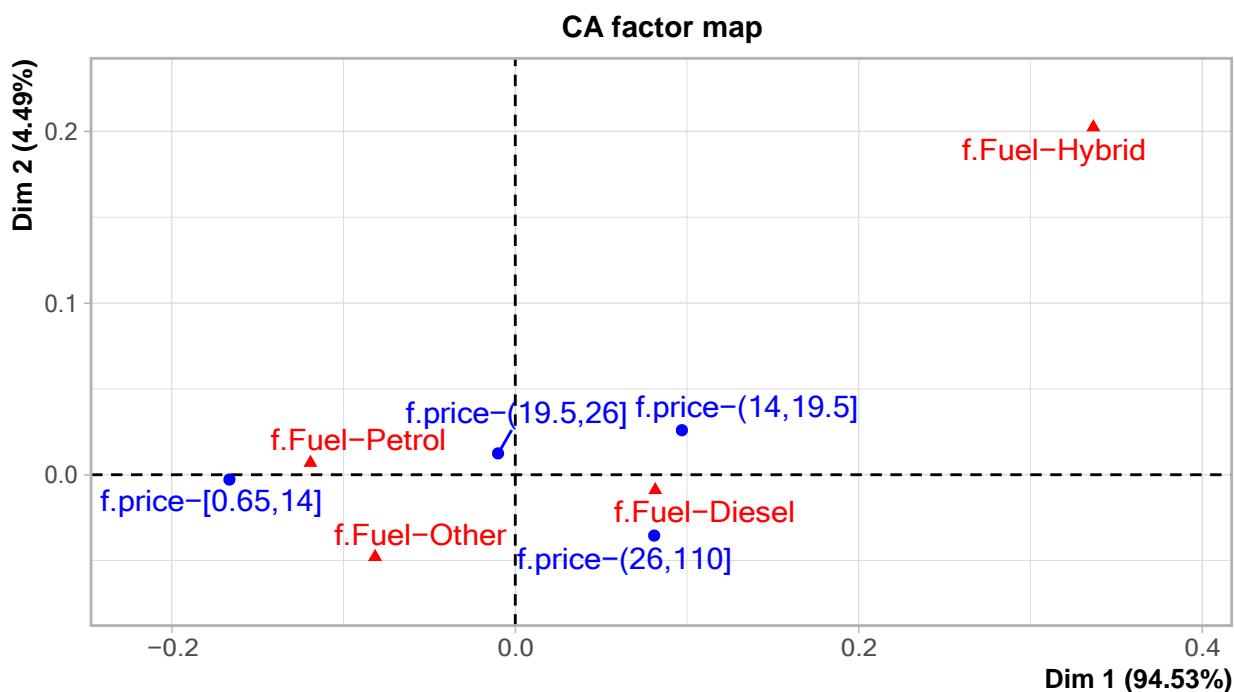
As we can see aux\_price and aux\_tax variables are not independent because the p-value is less than 0.05. Also, we can see that the cheapest cars have the highest taxes and the most expensive cars have average/medium taxes.

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

```
tt<-table(df[, c("aux_price", "fuelType")])
tt
```

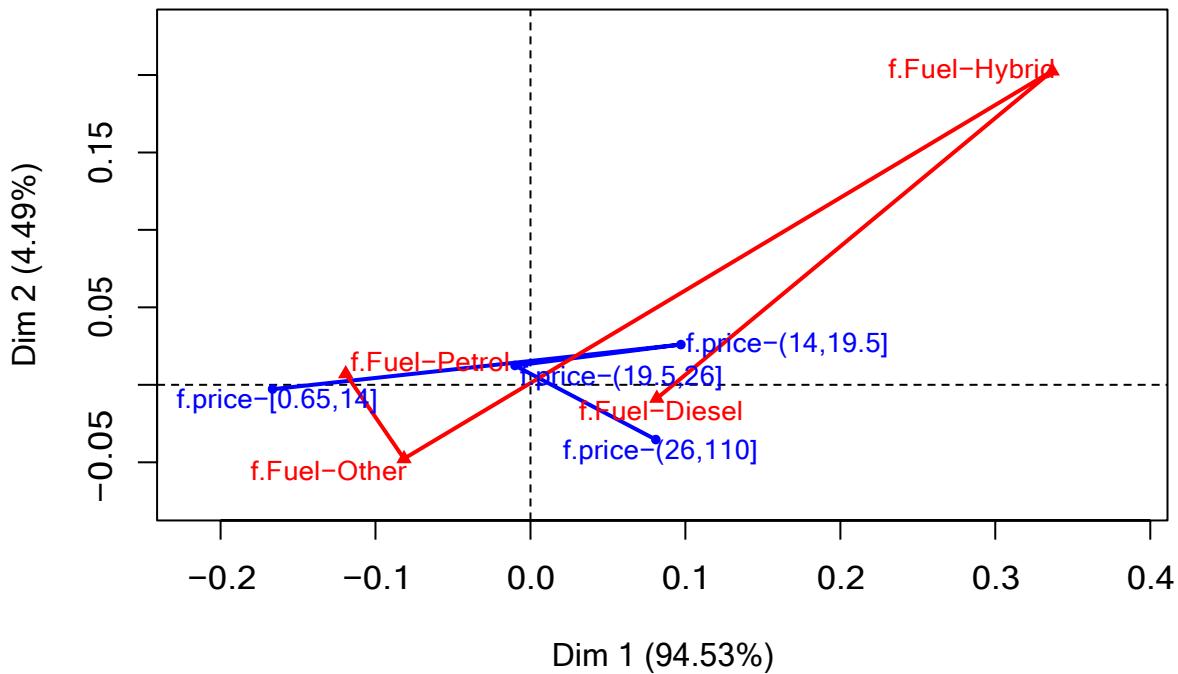
```
##          fuelType
## aux_price      f. Fuel-Diesel f. Fuel-Hybrid f. Fuel-Other f. Fuel-Petrol
##   f. price-[0.65,14]      624           6           3      626
##   f. price-(14,19.5]      755          21           2      468
##   f. price-(19.5,26]      698          15           4      528
##   f. price-(26,110]      765          13           3      469
```

```
res.ca<-CA(tt)
```



```
plot( res.ca, cex=0.8, graph.type = "classic" )
lines( res.ca$row$coord[, 1], res.ca$row$coord[, 2], col="blue", lwd = 2 )
lines( res.ca$col$coord[, 1], res.ca$col$coord[, 2], col="red", lwd = 2 )
```

## CA factor map



```
summary(res.ca)
```

```
##  
## Call:  
## CA(X = tt)  
##  
## The chi square of independence between the two variables is equal to 58.13151 (p-value = 3.066754e-09)  
##  
## Eigenvalues  
##  
##          Dim.1   Dim.2   Dim.3  
## Variance 0.011  0.001  0.000  
## % of var. 94.530  4.486  0.984  
## Cumulative % of var. 94.530 99.016 100.000  
##  
## Rows  
##  
## f.price-[0.65,14] | 6.995 | -0.167 63.516 0.998 | -0.003 0.386 0.000 |  
## f.price-(14,19.5] | 2.541 |  0.097 21.368 0.924 |  0.026 32.221 0.066 |  
## f.price-(19.5,26] | 0.140 | -0.010  0.227 0.178 |  0.012 7.348 0.273 |  
## f.price-(26,110] | 1.950 |  0.081 14.889 0.839 | -0.035 60.045 0.161 |  
##  
##          Dim.3  
## f.price-[0.65,14] | -0.007 10.918 0.002 |  
## f.price-(14,19.5] | -0.010 21.491 0.010 |  
## f.price-(19.5,26] |  0.018 67.525 0.550 |  
## f.price-(26,110] | -0.001  0.066 0.000 |  
##  
## Columns  
##  
## f.Fuel-Diesel | 3.826 | 0.082 34.405 0.988 | -0.009 8.554 0.012 |  
## f.Fuel-Hybrid | 1.699 | 0.337 11.345 0.734 |  0.202 86.477 0.265 |  
## f.Fuel-Other | 0.134 | -0.082 0.145 0.119 | -0.048 1.052 0.041 |  
## f.Fuel-Petrol | 5.967 | -0.119 54.105 0.997 |  0.007 3.917 0.003 |  
##  
##          Dim.3  
## f.Fuel-Diesel | -0.001 0.201 0.000 |  
## f.Fuel-Hybrid | 0.011 1.078 0.001 |
```

```
## f.Fuel-Other      0.217 98.563  0.840 |
## f.Fuel-Petrol    -0.001  0.158  0.000 |
```

```
chisq. test(tt)
```

```
## Warning in chisq.test(tt): Chi-squared approximation may be incorrect##
## Pearson's Chi-squared test
##
## data: tt
## X-squared = 58.132, df = 9, p-value = 3.067e-09
```

As we can see aux\_price and fuelType variables are not independent because the p-value is less than 0.05. Also, we can see that hybrid cars cause noise. We can see that diesel cars have the highest prices and petrol and other cars have the lowest prices.

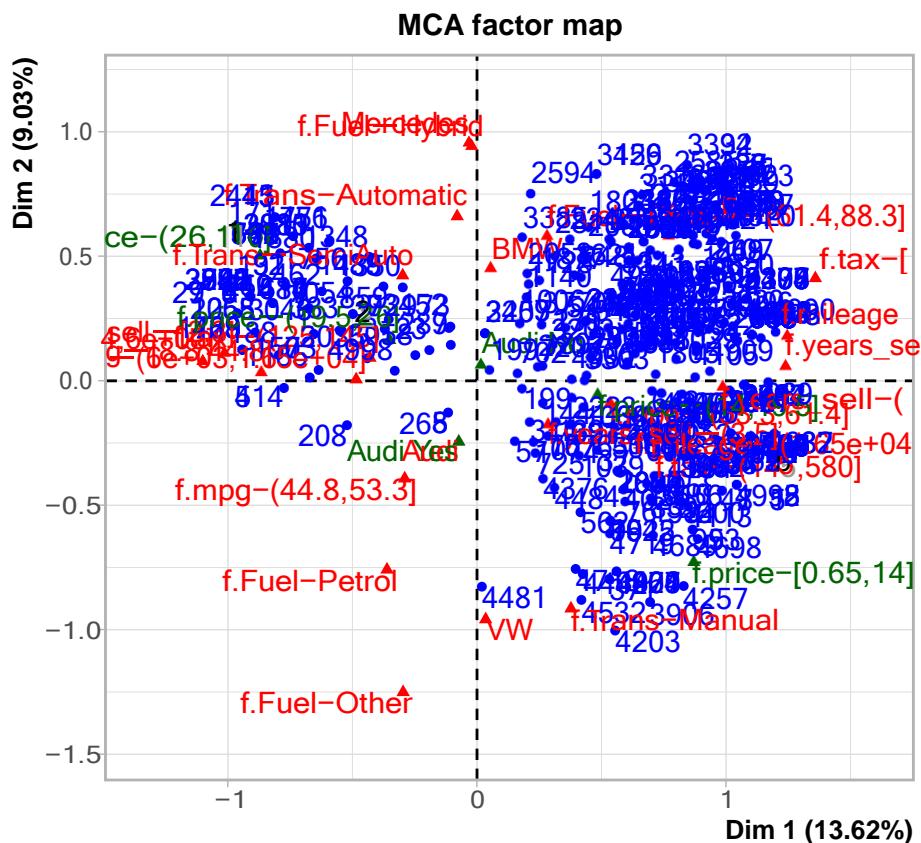
## 6 MCA

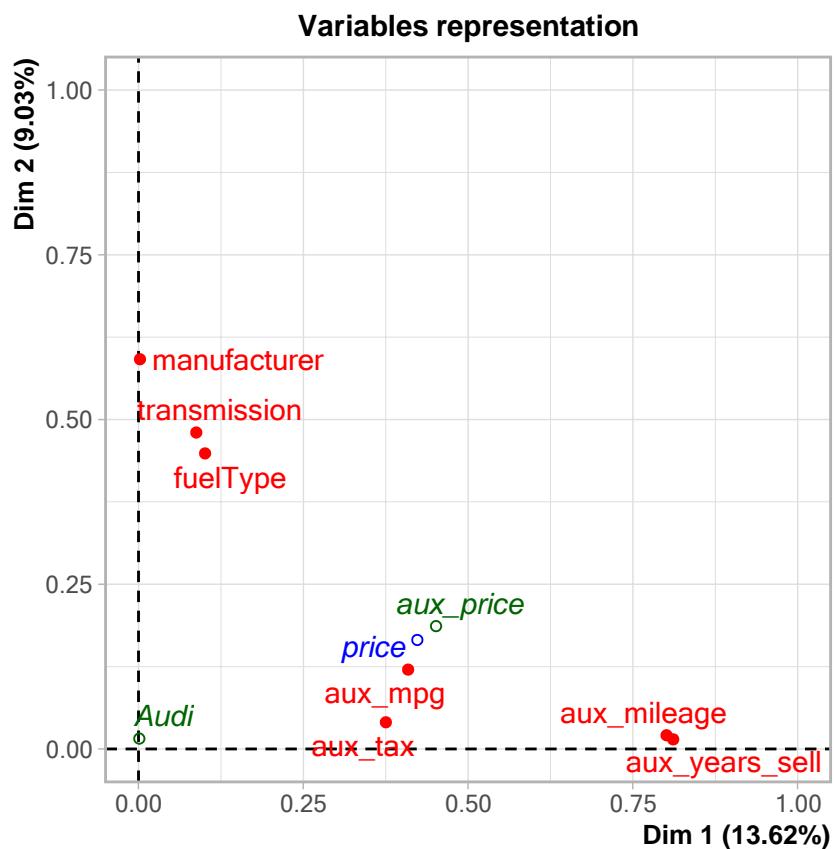
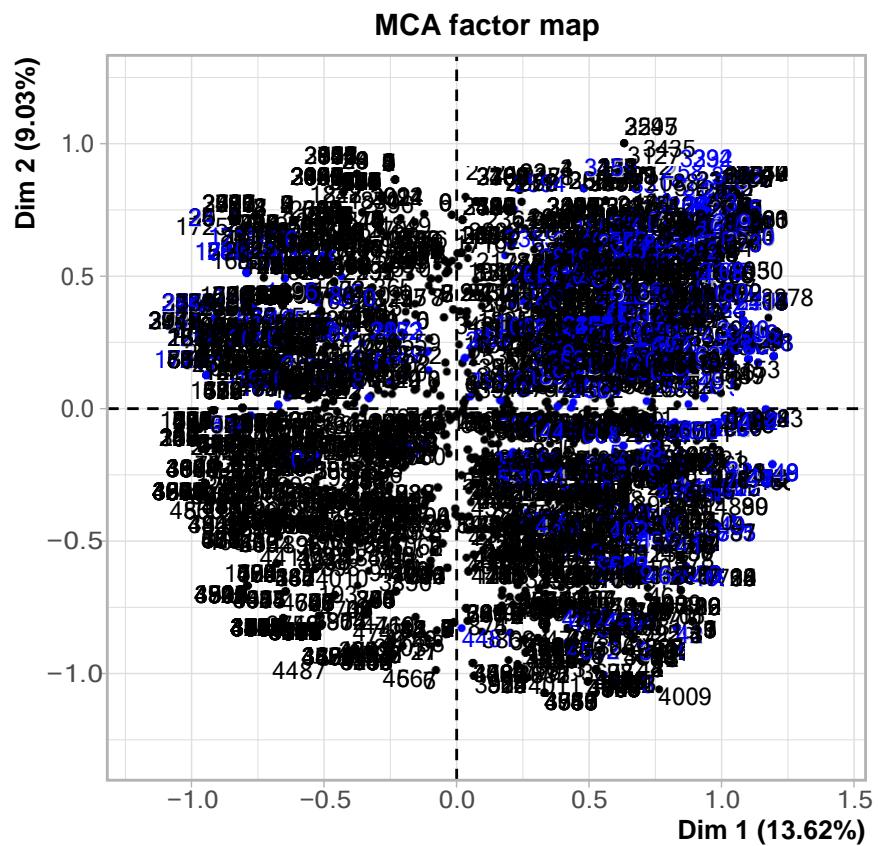
The objectives of MCA are close to PCA objectives: similarity among individuals/ similarity among variables. Also, generalization of CA objectives: relationships among the categories.

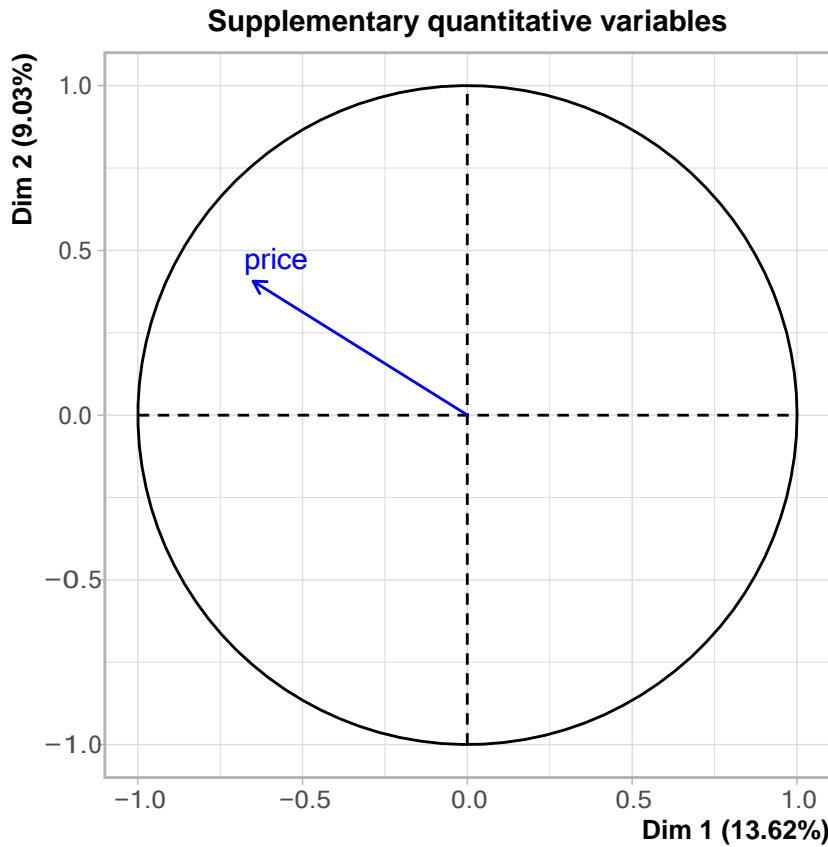
```
par(mfrow=c(1, 1))
11vout<-which(df$mout=="YesMOut");length(11vout) #Multivariate outliers

## [1] 463

res.mca<-MCA(df[, c(vars_dis[c(3:4, 6:11)], "price", "Audi")], quali.sup=c(4, 10), quanti.sup=9, ind.sup=11vout)
```







We can see that supplementary variable price is in the second quadrant (i.e. has a negative first dimension and a positive second dimension).

## 6.1 Eigenvalues and dominant axes analysis

```

mean(res.mca$eig[, 1])

## [1] 0.1428571

summary(res.mca)

## 
## Call:
## MCA(X = df[, c(vars_dis[c(3:4, 6:11)], "price", "Audi")], ind.sup = 11vout,
##       quanti.sup = 9, quali.sup = c(4, 10))
## 
## 
## Eigenvalues
##              Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance     0.370   0.245   0.221   0.184   0.166   0.160   0.154
## % of var. 13.620   9.031   8.152   6.762   6.099   5.906   5.668
## Cumulative % of var. 13.620  22.651  30.803  37.566  43.664  49.570  55.237
## 
##              Dim.8   Dim.9   Dim.10  Dim.11  Dim.12  Dim.13  Dim.14
## Variance     0.151   0.146   0.135   0.132   0.125   0.123   0.102
## % of var.  5.559   5.362   4.990   4.849   4.605   4.533   3.758
## Cumulative % of var. 60.796  66.159  71.148  75.997  80.601  85.134  88.892
## 
##              Dim.15  Dim.16  Dim.17  Dim.18  Dim.19
## Variance     0.084   0.079   0.066   0.049   0.023
## % of var.  3.088   2.915   2.441   1.816   0.847
## Cumulative % of var. 91.980  94.895  97.337  99.153 100.000
## 
## Individuals (the 10 first)
##              Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3   ctr

```

```

## 1 | 0.665 0.026 0.155 | -0.733 0.048 0.188 | -0.165 0.003
## 2 | 0.865 0.045 0.229 | -0.594 0.032 0.108 | 0.892 0.079
## 3 | 0.330 0.007 0.045 | 0.370 0.012 0.057 | -0.581 0.034
## 4 | -0.677 0.027 0.229 | -0.584 0.031 0.171 | 0.336 0.011
## 5 | -0.070 0.000 0.002 | -0.298 0.008 0.038 | -0.447 0.020
## 6 | 0.038 0.000 0.001 | -0.752 0.051 0.264 | -0.492 0.024
## 7 | 0.601 0.022 0.140 | -0.041 0.000 0.001 | -0.341 0.012
## 8 | 0.130 0.001 0.007 | -0.557 0.028 0.120 | -0.298 0.009
## 9 | -0.540 0.017 0.154 | 0.167 0.003 0.015 | -0.009 0.000
## 10 | -0.827 0.041 0.334 | -0.096 0.001 0.005 | 0.104 0.001
##
## cos2
## 1 | 0.010 |
## 2 | 0.243 |
## 3 | 0.140 |
## 4 | 0.057 |
## 5 | 0.086 |
## 6 | 0.113 |
## 7 | 0.045 |
## 8 | 0.034 |
## 9 | 0.000 |
## 10 | 0.005 |
##
## Supplementary individuals (the 10 first)
##           Dim.1   cos2   Dim.2   cos2   Dim.3   cos2
## 14 | 0.905 0.254 | 0.416 0.054 | 0.566 0.099 |
## 31 | -0.768 0.270 | 0.380 0.066 | 0.442 0.089 |
## 37 | -0.370 0.065 | 0.208 0.020 | -0.327 0.051 |
## 39 | 0.823 0.217 | -0.296 0.028 | 1.082 0.376 |
## 57 | 0.154 0.009 | -0.240 0.022 | 0.436 0.071 |
## 72 | 0.588 0.120 | 0.290 0.029 | 0.587 0.120 |
## 79 | 0.528 0.088 | 0.213 0.014 | 1.148 0.416 |
## 87 | -0.105 0.005 | 0.144 0.010 | -0.453 0.097 |
## 138 | 0.605 0.128 | 0.066 0.002 | 0.608 0.130 |
## 149 | 1.012 0.336 | -0.039 0.000 | 0.521 0.089 |
##
## Categories (the 10 first)
##           Dim.1   ctr   cos2   v. test   Dim.2   ctr   cos2
## f.Trans-Manual | 0.378 1.988 0.080 19.102 | -0.915 17.583 0.472
## f.Trans-SemiAuto | -0.298 1.342 0.057 -16.087 | 0.423 4.087 0.115
## f.Trans-Automatic | -0.079 0.060 0.002 -3.055 | 0.661 6.318 0.144
## f.Fuel-Diesel | 0.283 1.719 0.100 21.300 | 0.581 10.927 0.422
## f.Fuel-Hybrid | -0.018 0.000 0.000 -0.118 | 0.942 0.479 0.008
## f.Fuel-Other | -0.296 0.007 0.000 -0.937 | -1.251 0.201 0.003
## f.Fuel-Petrol | -0.361 2.180 0.100 -21.249 | -0.758 14.523 0.440
## Audi | -0.070 0.040 0.001 -2.434 | -0.242 0.724 0.016
## BMW | 0.058 0.026 0.001 1.947 | 0.452 2.396 0.051
## Mercedes | -0.029 0.009 0.000 -1.195 | 0.955 14.535 0.343
##           v. test   Dim.3   ctr   cos2   v. test
## f.Trans-Manual | -46.256 | -0.010 0.002 0.000 -0.518 |
## f.Trans-SemiAuto | 22.858 | -0.079 0.157 0.004 -4.255 |
## f.Trans-Automatic | 25.579 | 0.139 0.310 0.006 5.381 |
## f.Fuel-Diesel | 43.728 | -0.035 0.043 0.002 -2.620 |
## f.Fuel-Hybrid | 6.132 | -0.360 0.077 0.001 -2.342 |
## f.Fuel-Other | -3.959 | 0.199 0.006 0.000 0.631 |
## f.Fuel-Petrol | -44.663 | 0.051 0.074 0.002 3.021 |
## Audi | -8.455 | 0.311 1.318 0.026 10.835 |
## BMW | 15.279 | 0.250 0.812 0.016 8.448 |
## Mercedes | 39.461 | -0.348 2.135 0.046 -14.370 |
##
## Categorical variables (eta2)
##           Dim.1 Dim.2 Dim.3
## transmission | 0.088 0.480 0.007 |
## fuelType | 0.101 0.448 0.003 |
## manufacturer | 0.002 0.591 0.067 |

```

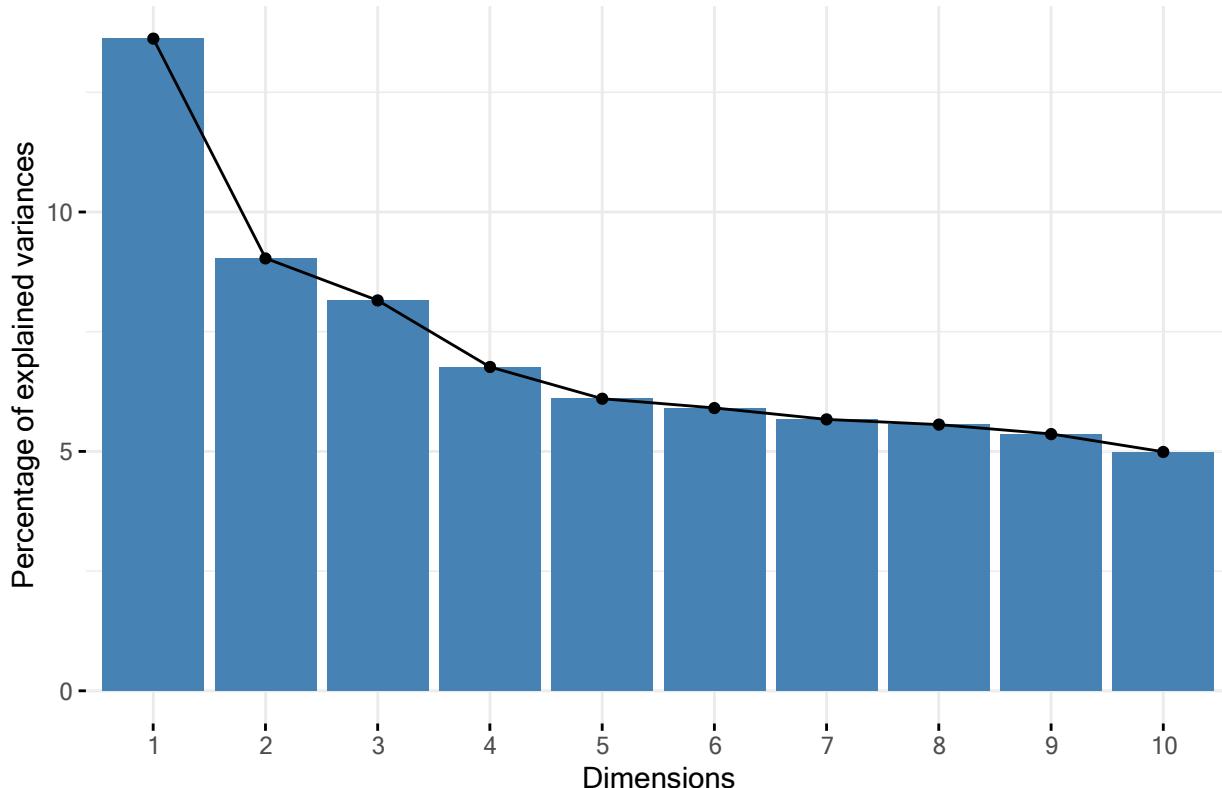
```

## aux_mileage | 0.801 0.021 0.583 |
## aux_tax | 0.375 0.040 0.143 |
## aux_mpg | 0.409 0.120 0.139 |
## aux_years_sell | 0.811 0.014 0.606 |
##
## Supplementary categories
## Dim. 1 cos2 v. test Dim. 2 cos2 v. test Dim. 3
## f.price-[0,65,14] | 0.869 0.210 30.878 | -0.725 0.146 -25.729 | 0.170
## f.price-(14,19.5] | 0.486 0.083 19.386 | -0.054 0.001 -2.147 | -0.163
## f.price-(19.5,26] | -0.351 0.045 -14.302 | 0.163 0.010 6.653 | -0.174
## f.price-(26,110] | -0.871 0.258 -34.241 | 0.503 0.086 19.776 | 0.204
## Audi No | 0.019 0.001 2.434 | 0.065 0.016 8.455 | -0.083
## Audi Yes | -0.070 0.001 -2.434 | -0.242 0.016 -8.455 | 0.311
##
## cos2 v. test
## f.price-[0,65,14] | 0.008 6.047 |
## f.price-(14,19.5] | 0.009 -6.489 |
## f.price-(19.5,26] | 0.011 -7.094 |
## f.price-(26,110] | 0.014 8.026 |
## Audi No | 0.026 -10.835 |
## Audi Yes | 0.026 10.835 |
##
## Supplementary categorical variables (eta2)
## Dim. 1 Dim. 2 Dim. 3
## aux_price | 0.452 0.186 0.032 |
## Audi | 0.001 0.016 0.026 |
##
## Supplementary continuous variable
## Dim. 1 Dim. 2 Dim. 3
## price | -0.650 | 0.407 | 0.048 |

```

```
fviz_eig(res.mca)
```

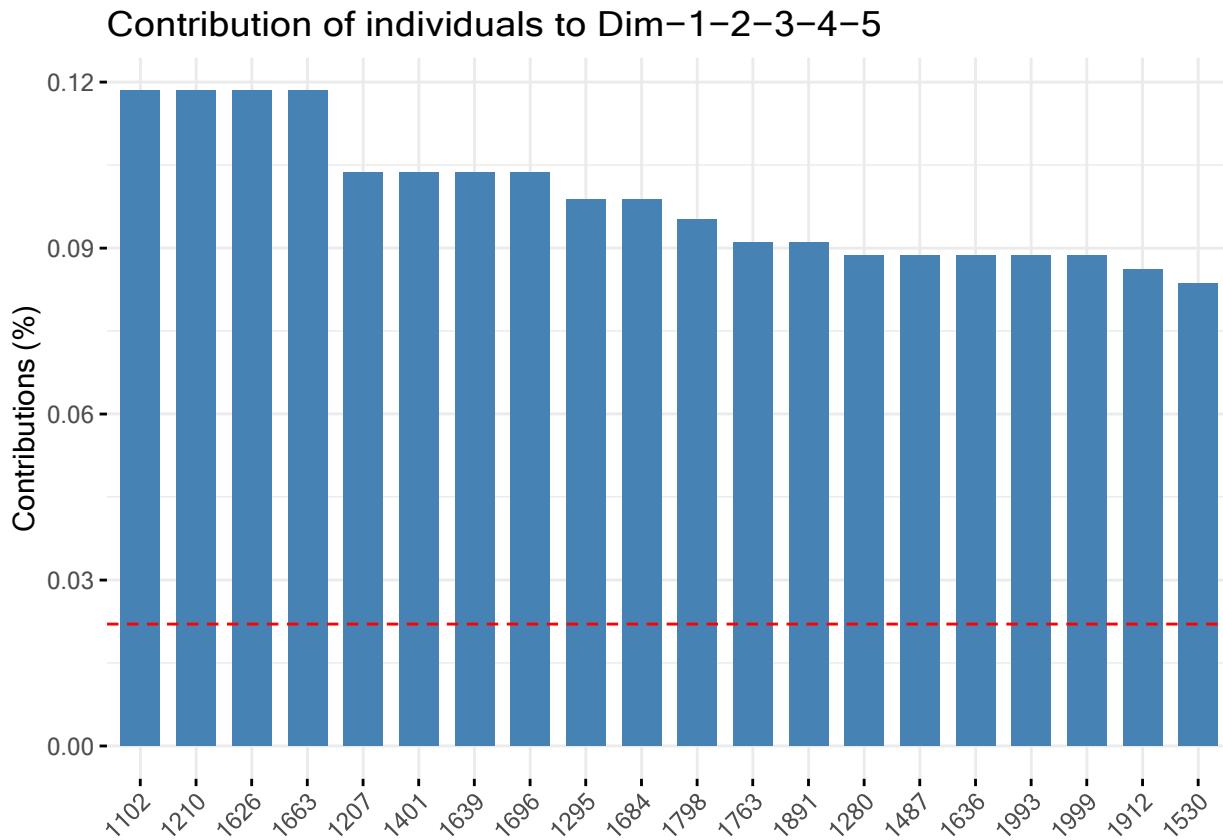
Scree plot



According to Elbow's rule we can pick up to 5 dimensions, the ones that brings more variability among planes.

## 6.2 Individuals point of view

```
fviz_contrib(res.mca, choice = "ind", axes = 1:5, top = 20)
```

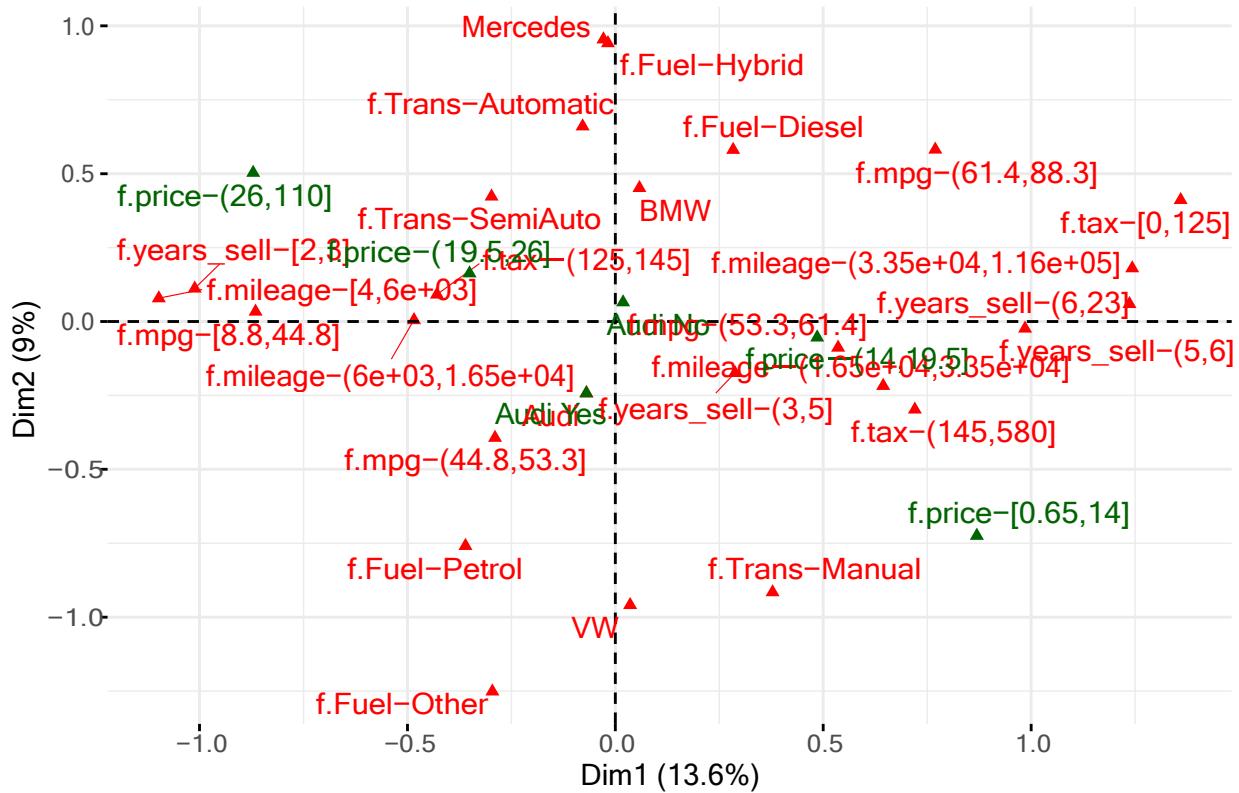


As we can see from the Contribution of individuals to Dim-1-2-3-4-5 plot, the 20 most contributive individuals in descending order are 1102, 1210, 1626, 1663, 1207, 1401, 1639, 1696, 1295, 1684, 1798, 1763, 1891, 1280, 1487, 1636, 1993, 1999, 1912 and 1530.

## 6.3 Interpreting map of categories

```
fviz_mca_var(res.mca, repel=TRUE)
```

## Variable categories – MCA



As we can see all the categories are sparse around the center of mass. In the first dimension we have as extremes f.mileage-[4,6e+03], f.years\_sell-[2,3] in the negative side and in the other one we have f.mileage-(3.35e+04,1.16e+05), f.years\_sell-(6,23) and f.tax-[0,125]. For the dimension 2 as extremes we can find f.Fuel-Hybrid and Mercedes(positive side) and f.Fuel-Other (negative side). These extremes are going to be poorly represented in that dimensions.

### 6.4 Interpreting the axes association to factor map

We can see/distinguish four clouds of variables (variables in the same cloud means they are related): First: manufacturer, transmission and fuelType variables. Second: Audi variable. Third: aux\_price, price, aux\_mpg and aux\_tax variables. Forth: aux\_mileage and aux\_years\_sell variables.

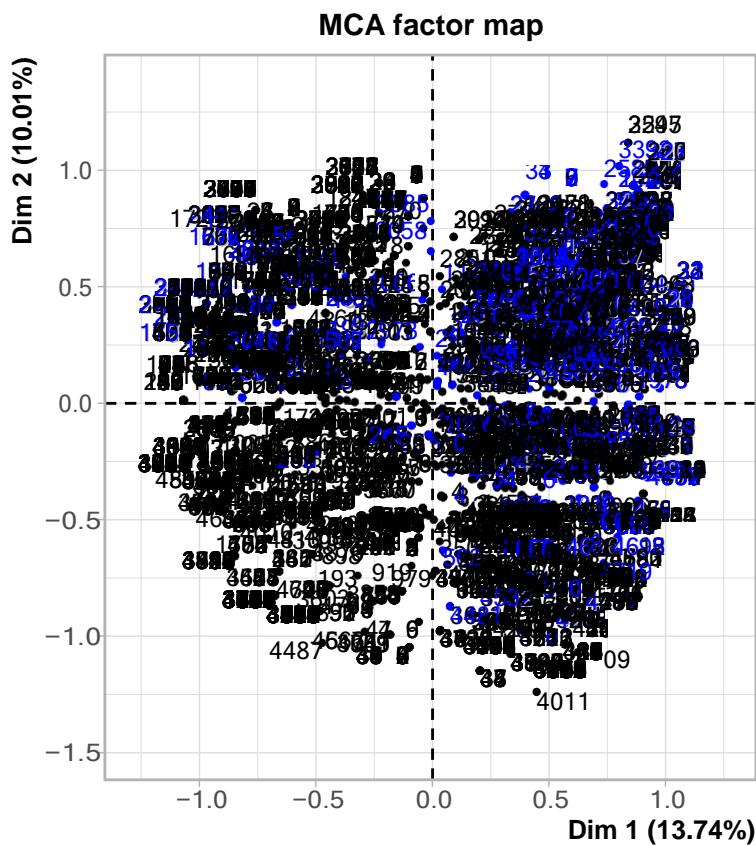
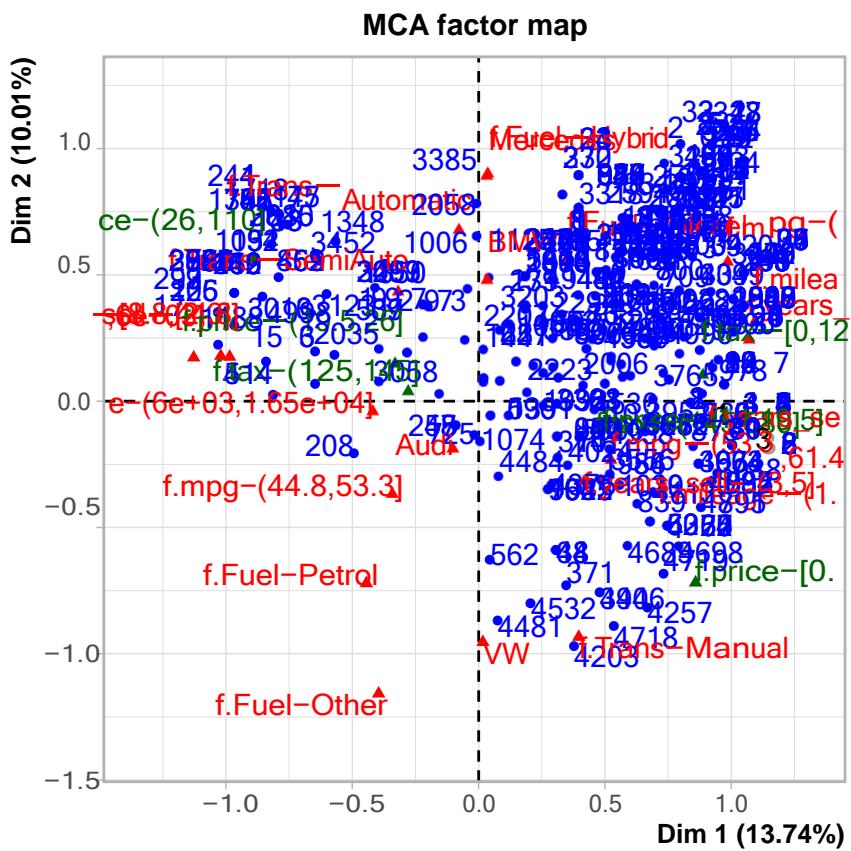
As we can see the cheapest cars are VW, have manual transmission, high taxes, medium years\_sell, medium-high mileage and medium-high mpg. As we can see the most expensive cars are Mercedes, hybrid, have automatic or semi-automatic transmission, medium taxes, few years\_sell, low mileage and low mpg.

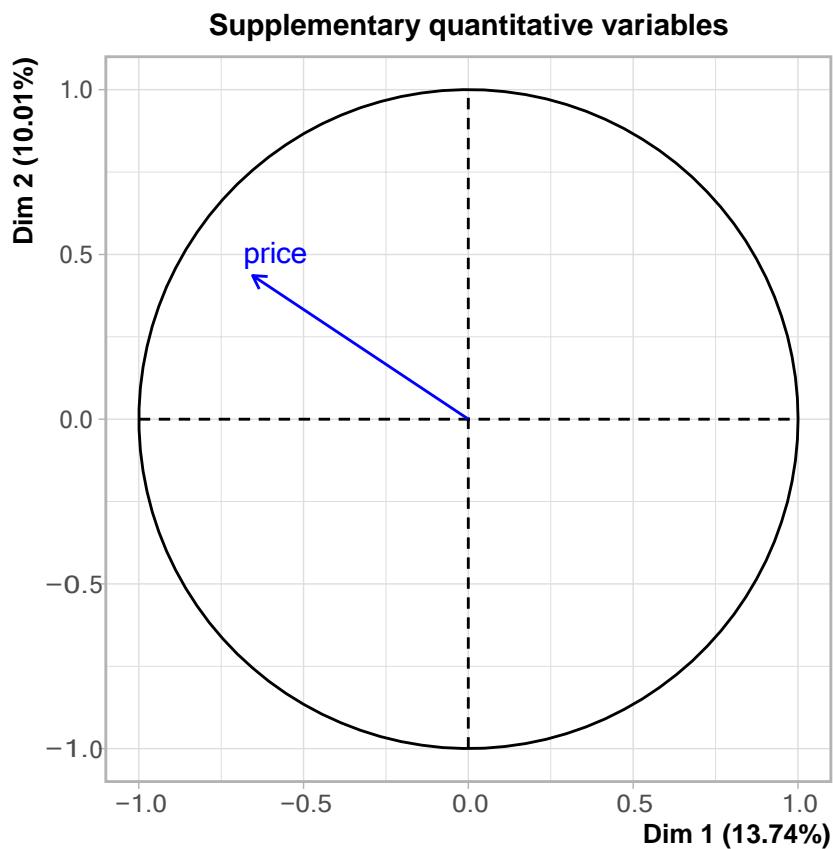
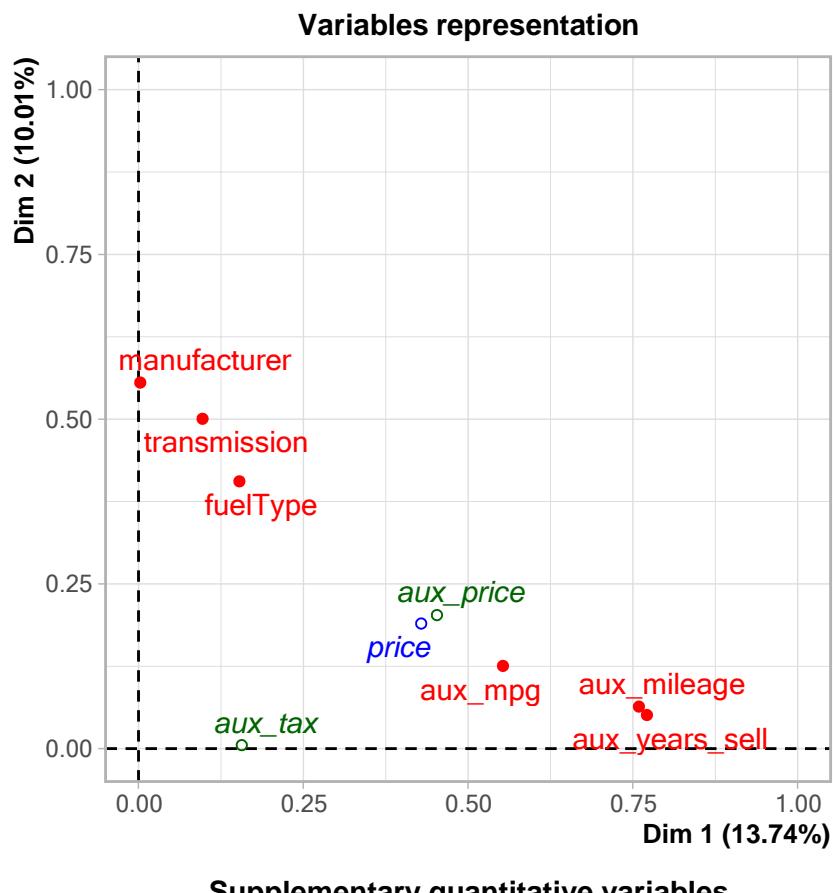
As we can see Audi cars are petrol or other and have medium-low mpg.

As we can see supplementary variable price is in the second quadrant (i.e. has a negative first dimension and a positive second dimension) which corresponds to the most expensive cars.

### 6.5 Hierarchical Clustering (from MCA)

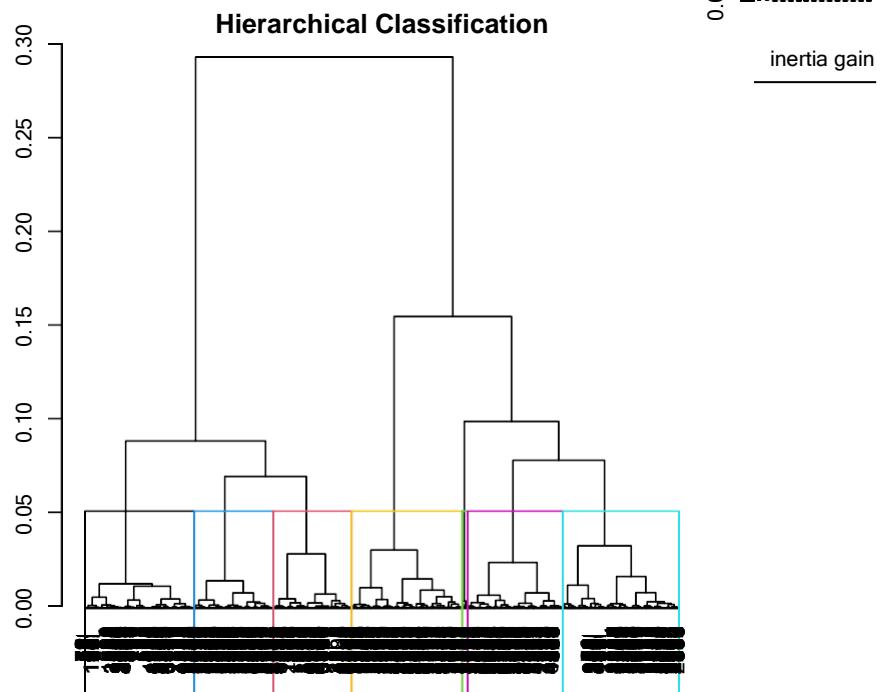
```
res.mca<-MCA(df[, c(vars_dis[c(3:4, 6:11)], "price") ], quali.sup=c(4, 6), quanti.sup=9, ind.sup=11vout, ncp
```



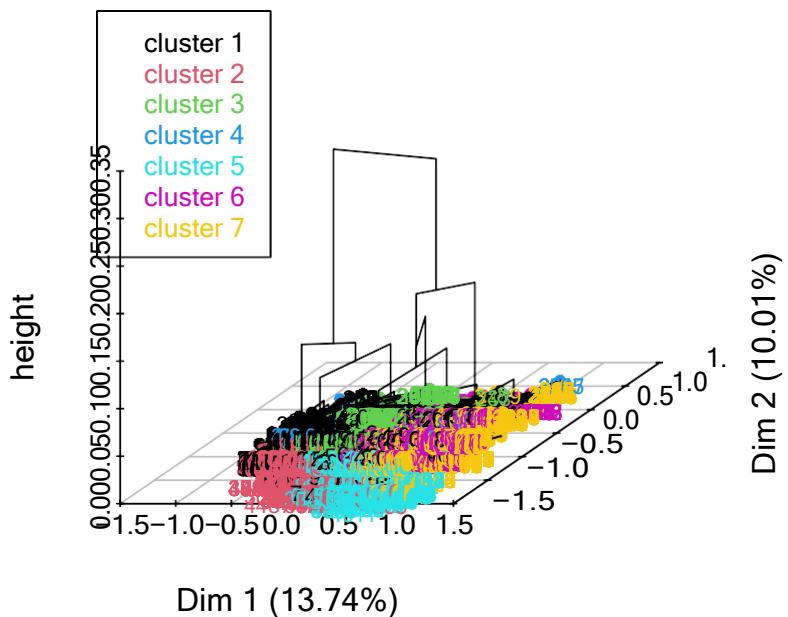


```
res.hcpc<-HCPC(res.mca, nb.clust=7, order=TRUE)
```

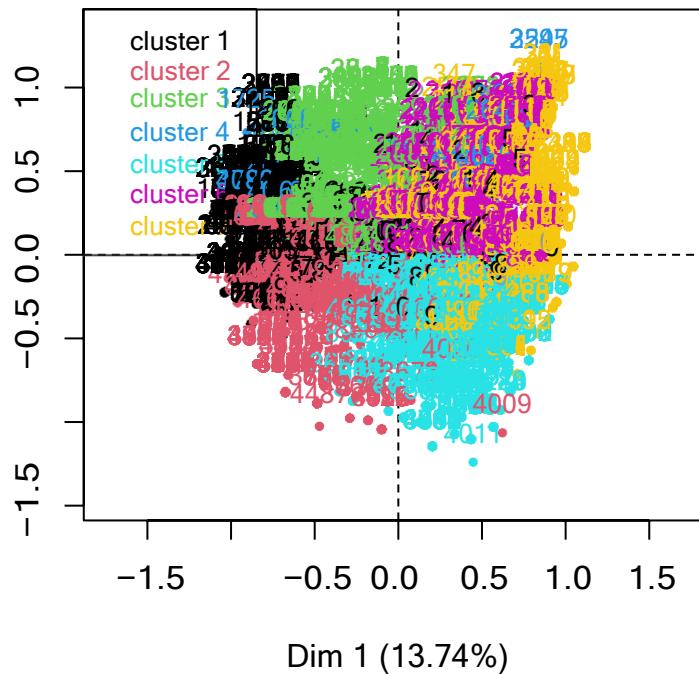
## Hierarchical Clustering



## Hierarchical clustering on the factor map



## Factor map



```
res.hcmc$desc.var
```

```
##  
## Link between the cluster variable and the categorical variables (chi-square test)##  
=====
```

	p. value	df
## transmission	0.000000e+00	12##
fuelType	0.000000e+00	18 ##
manufacturer	0.000000e+00	18 ##
aux_price	0.000000e+00	18 ##
aux_mileage	0.000000e+00	18 ##
aux_mpg	0.000000e+00	18 ##
aux_years_sell	0.000000e+00	18 ##
aux_tax	2.991422e-231	12##
## Description of each cluster by the categories		
## =====		
## \$ '1 '		
## aux_mpg=f.mpg-[8.8, 44.8]	61.4564831	87.0440252
## aux_years_sell=f.years_sell-[2, 3]	40.3659849	94.3396226
## aux_mileage=f.mileage-[4, 6e+03]	49.6253122	74.9685535
## aux_price=f.price-(26, 110]	47.6149176	69.0566038
## fuelType=f.Fuel-Petrol	28.7385554	71.0691824
## manufacturer=Audi	35.0000000	42.2641509
## aux_tax=f.tax-(125, 145]	22.6907631	85.2830189
## transmission=f.Trans-SemiAuto	26.0281690	58.1132075
## transmission=f.Trans-Automatic	23.1588287	32.8301887
## manufacturer=BMW	14.0350877	16.1006289
## aux_mileage=f.mileage-(6e+03, 1.65e+04]	14.3327842	21.8867925
## fuelType=f.Fuel-Hybrid	0.0000000	0.0000000
## aux_tax=f.tax-(145, 580]	9.0909091	14.7169811
## aux_tax=f.tax-[0, 125]	0.0000000	0.0000000
## manufacturer=VW	9.0589888	16.2264151
## aux_mpg=f.mpg-(44.8, 53.3]	7.5457317	12.4528302
		24.8181618
		40.9521710
		26.4712365
		25.4132687
		43.3325986
		21.1593564
		65.8584968
		39.1227683
		24.8402028
		20.1013886
		26.7577695
		0.9257218
		28.3667622
		5.7747410
		31.3863787
		28.9177871

```

## aux_years_sell=f.years_sell-(6, 23] 1. 3201320 1. 0062893 13. 3568437
## aux_years_sell=f.years_sell-(5, 6] 1. 3037810 1. 2578616 16. 9054441
## fuelType=f.Fuel-Diesel 9. 1306074 28. 9308176 55. 5212696
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05] 0. 8859358 1. 0062893 19. 9030196
## aux_price=f.price-(14, 19. 5] 2. 4576271 3. 6477987 26. 0083756
## transmission=f.Trans-Manual 4. 4036697 9. 0566038 36. 0370289
## aux_years_sell=f.years_sell-(3, 5] 2. 0673813 3. 3962264 28. 7855411
## aux_mileage=f.mileage-(1. 65e+04, 3. 35e+04] 1. 3945857 2. 1383648 26. 8679744
## aux_mpg=f.mpg-(61. 4, 88. 3] 0. 0000000 0. 0000000 21. 7544633
## aux_price=f.price-[0. 65, 14] 0. 0000000 0. 0000000 21. 7544633
## aux_mpg=f.mpg-(53. 3, 61. 4] 0. 3597122 0. 5031447 24. 5095878
## p. value v. test
## aux_mpg=f.mpg-[8. 8, 44. 8] 0. 000000e+00 Inf
## aux_years_sell=f.years_sell-[2, 3] 8. 947635e-273 35. 288077
## aux_mileage=f.mileage-[4, 6e+03] 1. 749928e-227 32. 200264
## aux_price=f.price-(26, 110] 5. 067290e-187 29. 167125
## fuelType=f.Fuel-Petrol 7. 441118e-68 17. 405912
## manufacturer=Audi 3. 643550e-51 15. 046433
## aux_tax=f.tax-(125, 145] 3. 064926e-41 13. 450332
## transmission=f.Trans-SemiAuto 8. 189115e-33 11. 930691
## transmission=f.Trans-Automatic 2. 099937e-08 5. 603560
## manufacturer=BMW 1. 598442e-03 -3. 156191
## aux_mileage=f.mileage-(6e+03, 1. 65e+04] 5. 289272e-04 -3. 465662
## fuelType=f.Fuel-Hybrid 2. 940636e-04 -3. 620474
## aux_tax=f.tax-(145, 580] 4. 602972e-23 -9. 889941
## aux_tax=f.tax-[0, 125] 2. 256269e-23 -9. 961068
## manufacturer=VW 2. 087998e-26 -10. 633210
## aux_mpg=f.mpg-(44. 8, 53. 3] 2. 768158e-33 -12. 020641
## aux_years_sell=f.years_sell-(6, 23] 1. 120750e-42 -13. 692819
## aux_years_sell=f.years_sell-(5, 6] 3. 382046e-55 -15. 648864
## fuelType=f.Fuel-Diesel 1. 895983e-62 -16. 677955
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05] 2. 087366e-71 -17. 868168
## aux_price=f.price-(14, 19. 5] 4. 734562e-74 -18. 204721
## transmission=f.Trans-Manual 1. 491716e-80 -19. 007027
## aux_years_sell=f.years_sell-(3, 5] 1. 509707e-89 -20. 064444
## aux_mileage=f.mileage-(1. 65e+04, 3. 35e+04] 2. 077178e-92 -20. 389390
## aux_mpg=f.mpg-(61. 4, 88. 3] 3. 830551e-95 -20. 695147
## aux_price=f.price-[0. 65, 14] 3. 830551e-95 -20. 695147
## aux_mpg=f.mpg-(53. 3, 61. 4] 1. 852422e-100 -21. 277049

##
## $ '2 '
## Cla/Mod Mod/Cla Global
## aux_mpg=f.mpg-(44. 8, 53. 3] 35. 0609756 75. 9075908 28. 9177871
## aux_years_sell=f.years_sell-[2, 3] 28. 5791173 87. 6237624 40. 9521710
## manufacturer=VW 30. 6179775 71. 9471947 31. 3863787
## transmission=f.Trans-Manual 25. 9938838 70. 1320132 36. 0370289
## aux_mileage=f.mileage-[4, 6e+03] 27. 9766861 55. 4455446 26. 4712365
## fuelType=f.Fuel-Petrol 21. 1088505 68. 4818482 43. 3325986
## aux_tax=f.tax-(125, 145] 17. 4364123 85. 9735974 65. 8584968
## aux_price=f.price-(19. 5, 26] 21. 3640099 42. 9042904 26. 8238924
## aux_mileage=f.mileage-(6e+03, 1. 65e+04] 21. 1696870 42. 4092409 26. 7577695
## fuelType=f.Fuel-Other 80. 0000000 1. 3201320 0. 2204100
## aux_price=f.price-(14, 19. 5] 16. 6949153 32. 5082508 26. 0083756
## fuelType=f.Fuel-Hybrid 0. 0000000 0. 0000000 0. 9257218
## manufacturer=BMW 10. 0877193 15. 1815182 20. 1013886
## aux_price=f.price-[0. 65, 14] 8. 6119554 14. 0264026 21. 7544633
## aux_tax=f.tax-[0, 125] 1. 1450382 0. 4950495 5. 7747410
## transmission=f.Trans-Automatic 7. 0984916 13. 2013201 24. 8402028
## manufacturer=Audi 6. 2500000 9. 9009901 21. 1593564
## aux_mpg=f.mpg-[8. 8, 44. 8] 6. 3055062 11. 7161716 24. 8181618
## aux_mpg=f.mpg-(53. 3, 61. 4] 5. 9352518 10. 8910891 24. 5095878
## aux_tax=f.tax-(145, 580] 6. 3714064 13. 5313531 28. 3667622
## aux_price=f.price-(26, 110] 5. 5507372 10. 5610561 25. 4132687
## aux_years_sell=f.years_sell-(3, 5] 5. 2067381 11. 2211221 28. 7855411

```

```

## aux_years_sell=f.years_sell-(6, 23]          0. 8250825  0. 8250825 13. 3568437
## transmission=f.Trans-SemiAuto             5. 6901408 16. 6666667 39. 1227683
## fuelType=f.Fuel-Diesel                   7. 2647876 30. 1980198 55. 5212696
## aux_years_sell=f.years_sell-(5, 6]          0. 2607562 0. 3300330 16. 9054441
## aux_mpg=f.mpg-(61. 4, 88. 3]              0. 9118541 1. 4851485 21. 7544633
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05] 0. 4429679 0. 6600660 19. 9030196
## manufacturer=Mercedes                   1. 4504432 2. 9702970 27. 3528764
## aux_mileage=f.mileage-(1. 65e+04, 3. 35e+04] 0. 7383101 1. 4851485 26. 8679744
##
## aux_mpg=f.mpg-(44. 8, 53. 3]                p. value      v. test
2. 901152e-149 26. 020527
## aux_years_sell=f.years_sell-[2, 3]          4. 020107e-145 25. 651963
## manufacturer=VW                         3. 499265e-109 22. 199160
## transmission=f.Trans-Manual             1. 779691e-75 18. 383539
## aux_mileage=f.mileage-[4, 6e+03]          3. 361859e-60 16. 365692
## fuelType=f.Fuel-Petrol                  5. 683146e-41 13. 404597
## aux_tax=f.tax-(125, 145]                 7. 321830e-33 11. 940006
## aux_price=f.price-(19. 5, 26]            3. 004438e-20 9. 218800
## aux_mileage=f.mileage-(6e+03, 1. 65e+04] 2. 578805e-19 8. 985363
## fuelType=f.Fuel-Other                   3. 525329e-06 4. 637568
## aux_price=f.price-(14, 19. 5]            1. 224549e-04 3. 841159
## fuelType=f.Fuel-Hybrid                 2. 355318e-03 -3. 041335
## manufacturer=BMW                      8. 738152e-04 -3. 328288
## aux_price=f.price-[0. 65, 14]           2. 508189e-07 -5. 157089
## aux_tax=f.tax-[0, 125]                  2. 198655e-13 -7. 336126
## transmission=f.Trans-Automatic        5. 093915e-14 -7. 529489
## manufacturer=Audi                     5. 140521e-15 -7. 823425
## aux_mpg=f.mpg-[8. 8, 44. 8]            1. 075290e-17 -8. 565584
## aux_mpg=f.mpg-(53. 3, 61. 4]          2. 474226e-19 -8. 989915
## aux_tax=f.tax-(145, 580]              2. 384043e-20 -9. 243571
## aux_price=f.price-(26, 110]            2. 287907e-22 -9. 728116
## aux_years_sell=f.years_sell-(3, 5]        1. 932095e-28 -11. 061331
## aux_years_sell=f.years_sell-(6, 23]        2. 696356e-33 -12. 022812
## transmission=f.Trans-SemiAuto        1. 631422e-37 -12. 800384
## fuelType=f.Fuel-Diesel                1. 427930e-41 -13. 506693
## aux_years_sell=f.years_sell-(5, 6]        2. 188978e-49 -14. 772956
## aux_mpg=f.mpg-(61. 4, 88. 3]          1. 586688e-55 -15. 696959
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05] 2. 496357e-56 -15. 813873
## manufacturer=Mercedes                 3. 923842e-64 -16. 908065
## aux_mileage=f.mileage-(1. 65e+04, 3. 35e+04] 5. 896455e-74 -18. 192698
##
## $ '3 '
##
## aux_mileage=f.mileage-(6e+03, 1. 65e+04]          Cla/Mod      Mod/Cla      Global
42. 9159802 67. 0527671 26. 7577695
## fuelType=f.Fuel-Diesel                      26. 6772529 86. 4864865 55. 5212696
## manufacturer=Mercedes                     35. 6970185 57. 0141570 27. 3528764
## aux_years_sell=f.years_sell-[2, 3]          29. 6555436 70. 9137709 40. 9521710
## aux_price=f.price-(26, 110]                34. 6053773 51. 3513514 25. 4132687
## transmission=f.Trans-SemiAuto            29. 2394366 66. 7953668 39. 1227683
## aux_tax=f.tax-(125, 145]                  22. 8246319 87. 7734878 65. 8584968
## aux_price=f.price-(19. 5, 26]            24. 9794577 39. 1248391 26. 8238924
## manufacturer=BMW                      24. 8903509 29. 2149292 20. 1013886
## transmission=f.Trans-Automatic        21. 7391304 31. 5315315 24. 8402028
## aux_mpg=f.mpg-(44. 8, 53. 3]            20. 4268293 34. 4916345 28. 9177871
## aux_mileage=f.mileage-[4, 6e+03]          20. 1498751 31. 1454311 26. 4712365
## aux_mpg=f.mpg-(53. 3, 61. 4]          19. 1546763 27. 4131274 24. 5095878
## fuelType=f.Fuel-Hybrid                  0. 0000000 0. 0000000 0. 9257218
## aux_mpg=f.mpg-[8. 8, 44. 8]            11. 2788632 16. 3449163 24. 8181618
## aux_tax=f.tax-[0, 125]                  1. 5267176 0. 5148005 5. 7747410
## aux_tax=f.tax-(145, 580]              7. 0707071 11. 7117117 28. 3667622
## aux_price=f.price-(14, 19. 5]          6. 1864407 9. 3951094 26. 0083756
## manufacturer=Audi                     3. 9583333 4. 8906049 21. 1593564
## aux_years_sell=f.years_sell-(6, 23]        0. 6600660 0. 5148005 13. 3568437
## manufacturer=VW                      4. 8455056 8. 8803089 31. 3863787
## aux_years_sell=f.years_sell-(5, 6]        0. 6518905 0. 6435006 16. 9054441

```

```

## aux_mileage=f.mileage-(3.35e+04, 1.16e+05] 1.1074197 1.2870013 19.9030196
## fuelType=f.Fuel-Petrol 5.3407935 13.5135135 43.3325986
## aux_price=f.price-[0.65, 14] 0.1013171 0.1287001 21.7544633
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 0.3281378 0.5148005 26.8679744
## transmission=f.Trans-Manual 0.7951070 1.6731017 36.0370289
## p. value v. test
## aux_mileage=f.mileage-(6e+03, 1.65e+04] 2.555052e-152 26.289103
## fuelType=f.Fuel-Diesel 2.075781e-90 20.162849
## manufacturer=Mercedes 1.802930e-83 19.356326
## aux_years_sell=f.years_sell-[2, 3] 4.689318e-77 18.579722
## aux_price=f.price-(26, 110] 9.446817e-67 17.259781
## transmission=f.Trans-SemiAuto 4.651756e-66 17.167481
## aux_tax=f.tax-(125, 145] 7.023825e-52 15.154979
## aux_price=f.price-(19.5, 26] 1.655429e-16 8.244723
## manufacturer=BMW 1.956959e-11 6.709199
## transmission=f.Trans-Automatic 3.416919e-06 4.644022
## aux_mpg=f.mpg-(44.8, 53.3] 2.038820e-04 3.714157
## aux_mileage=f.mileage-[4, 6e+03] 1.367217e-03 3.201484
## aux_mpg=f.mpg-(53.3, 61.4] 4.044369e-02 2.049188
## fuelType=f.Fuel-Hybrid 3.601255e-04 -3.567702
## aux_mpg=f.mpg-[8.8, 44.8] 5.036070e-10 -6.217976
## aux_tax=f.tax-[0, 125] 4.403798e-17 -8.401615
## aux_tax=f.tax-(145, 580] 9.019985e-34 -12.112942
## aux_price=f.price-(14, 19.5] 2.605585e-36 -12.583403
## manufacturer=Audi 4.043395e-43 -13.766686
## aux_years_sell=f.years_sell-(6, 23] 7.333277e-47 -14.375879
## manufacturer=VW 3.038982e-59 -16.231113
## aux_years_sell=f.years_sell-(5, 6] 1.585045e-60 -16.411401
## aux_mileage=f.mileage-(3.35e+04, 1.16e+05] 1.527820e-66 -17.231998
## fuelType=f.Fuel-Petrol 1.058377e-84 -19.501872
## aux_price=f.price-[0.65, 14] 2.597887e-90 -20.151746
## aux_mileage=f.mileage-(1.65e+04, 3.35e+04] 1.052208e-109 -22.253117
## transmission=f.Trans-Manual 2.093488e-144 -25.587653
##
## $ '4'
## Cla/Mod Mod/Cla Global p. value
## fuelType=f.Fuel-Hybrid 100.0000000 100.000000 0.9257218 4.426488e-103
## manufacturer=BMW 2.9605263 64.285714 20.1013886 5.416173e-10
## transmission=f.Trans-Automatic 2.3957409 64.285714 24.8402028 7.678985e-08
## aux_mpg=f.mpg-(53.3, 61.4] 1.9784173 52.380952 24.5095878 1.101042e-04
## aux_mpg=f.mpg-[8.8, 44.8] 0.3552398 9.523810 24.8181618 1.446723e-02
## manufacturer=VW 0.4213483 14.285714 31.3863787 1.236833e-02
## manufacturer=Audi 0.2083333 4.761905 21.1593564 3.860339e-03
## aux_mpg=f.mpg-(61.4, 88.3] 0.2026342 4.761905 21.7544633 2.976962e-03
## aux_price=f.price-[0.65, 14] 0.2026342 4.761905 21.7544633 2.976962e-03
## transmission=f.Trans-Manual 0.0000000 0.000000 36.0370289 6.341645e-09
## fuelType=f.Fuel-Petrol 0.0000000 0.000000 43.3325986 3.770847e-11
## fuelType=f.Fuel-Diesel 0.0000000 0.000000 55.5212696 1.313682e-15
## v. test
## fuelType=f.Fuel-Hybrid 21.558291
## manufacturer=BMW 6.206546
## transmission=f.Trans-Automatic 5.374513
## aux_mpg=f.mpg-(53.3, 61.4] 3.867173
## aux_mpg=f.mpg-[8.8, 44.8] -2.445448
## manufacturer=VW -2.501457
## manufacturer=Audi -2.889354
## aux_mpg=f.mpg-(61.4, 88.3] -2.970107
## aux_price=f.price-[0.65, 14] -2.970107
## transmission=f.Trans-Manual -5.807490
## fuelType=f.Fuel-Petrol -6.612813
## fuelType=f.Fuel-Diesel -7.993305
##
## $ '5'
## Cla/Mod Mod/Cla Global

```

```

## transmission=f. Trans-Manual          37. 30886850 78. 4061697 36. 0370289
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04] 42. 82198523 67. 0951157 26. 8679744
## fuelType=f. Fuel-Petrol             33. 36724313 84. 3187661 43. 3325986
## aux_years_sell=f. years_sell-(3, 5] 38. 20826953 64. 1388175 28. 7855411
## manufacturer=VW                   35. 74438202 65. 4241645 31. 3863787
## aux_price=f. price-[0. 65, 14]      41. 84397163 53. 0848329 21. 7544633
## aux_mpg=f. mpg-(53. 3, 61. 4]       34. 17266187 48. 8431877 24. 5095878
## aux_years_sell=f. years_sell-(5, 6] 29. 98696219 29. 5629820 16. 9054441
## aux_tax=f. tax-(145, 580]          24. 78632479 41. 0025707 28. 3667622
## manufacturer=Audi                  22. 08333333 27. 2493573 21. 1593564
## aux_price=f. price-(14, 19. 5]      21. 18644068 32. 1336761 26. 0083756
## aux_mileage=f. mileage-(6e+03, 1. 65e+04] 14. 08566722 21. 9794344 26. 7577695
## fuelType=f. Fuel-Hybrid            0. 00000000 0. 00000000 0. 9257218
## aux_mpg=f. mpg-(61. 4, 88. 3]       11. 85410334 15. 0385604 21. 7544633
## aux_years_sell=f. years_sell-(6, 23] 7. 92079208 6. 1696658 13. 3568437
## aux_tax=f. tax-(125, 145]          13. 88888889 53. 3419023 65. 8584968
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05] 8. 08416390 9. 3830334 19. 9030196
## aux_price=f. price-(19. 5, 26]       8. 46343468 13. 2390746 26. 8238924
## aux_mpg=f. mpg-[8. 8, 44. 8]         6. 83836590 9. 8971722 24. 8181618
## manufacturer=BMW                  3. 94736842 4. 6272494 20. 1013886
## transmission=f. Trans-Automatic   4. 17036380 6. 0411311 24. 8402028
## transmission=f. Trans-SemiAuto   6. 81690141 15. 5526992 39. 1227683
## manufacturer=Mercedes            1. 69218372 2. 6992288 27. 3528764
## aux_price=f. price-(26, 110]        1. 04076323 1. 5424165 25. 4132687
## aux_mileage=f. mileage-[4, 6e+03]   0. 99916736 1. 5424165 26. 4712365
## fuelType=f. Fuel-Diesel           4. 80349345 15. 5526992 55. 5212696
## aux_years_sell=f. years_sell-[2, 3] 0. 05382131 0. 1285347 40. 9521710
##                                     p. value   v. test
## transmission=f. Trans-Manual      3. 717752e-157 26. 708821
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04] 7. 746897e-152 26. 246937
## fuelType=f. Fuel-Petrol           5. 390946e-148 25. 908147
## aux_years_sell=f. years_sell-(3, 5] 3. 920826e-115 22. 806851
## manufacturer=VW                  4. 858213e-104 21. 660322
## aux_price=f. price-[0. 65, 14]     7. 481019e-103 21. 533988
## aux_mpg=f. mpg-(53. 3, 61. 4]      2. 747558e-60 16. 377971
## aux_years_sell=f. years_sell-(5, 6] 1. 843823e-22 9. 750047
## aux_tax=f. tax-(145, 580]          6. 778181e-17 8. 350835
## manufacturer=Audi                  8. 211719e-06 4. 459588
## aux_price=f. price-(14, 19. 5]      2. 630161e-05 4. 203331
## aux_mileage=f. mileage-(6e+03, 1. 65e+04] 7. 952001e-04 -3. 354460
## fuelType=f. Fuel-Hybrid            3. 561028e-04 -3. 570645
## aux_mpg=f. mpg-(61. 4, 88. 3]       2. 591507e-07 -5. 150964
## aux_years_sell=f. years_sell-(6, 23] 2. 854463e-12 -6. 984709
## aux_tax=f. tax-(125, 145]          1. 965129e-15 -7. 943526
## aux_mileage=f. mileage-(3. 35e+04, 1. 16e+05] 5. 432678e-18 -8. 643899
## aux_price=f. price-(19. 5, 26]       2. 671912e-23 -9. 944245
## aux_mpg=f. mpg-[8. 8, 44. 8]         3. 295473e-30 -11. 420706
## manufacturer=BMW                  7. 620494e-41 -13. 382816
## transmission=f. Trans-Automatic   3. 297226e-50 -14. 899971
## transmission=f. Trans-SemiAuto   7. 598189e-55 -15. 597263
## manufacturer=Mercedes            5. 615007e-87 -19. 768008
## aux_price=f. price-(26, 110]        1. 712382e-89 -20. 058180
## aux_mileage=f. mileage-[4, 6e+03]   1. 128337e-94 -20. 643001
## fuelType=f. Fuel-Diesel           4. 667683e-141 -25. 285026
## aux_years_sell=f. years_sell-[2, 3] 1. 112797e-199 -30. 148782
##                                     Cla/Mod  Mod/Cla  Global
## aux_mileage=f. mileage-(1. 65e+04, 3. 35e+04] 47. 4979491 74. 709677 26. 8679744
## fuelType=f. Fuel-Diesel             27. 9079000 90. 709677 55. 5212696
## aux_years_sell=f. years_sell-(3, 5] 34. 1500766 57. 548387 28. 7855411
## aux_mpg=f. mpg-(61. 4, 88. 3]       35. 3596758 45. 032258 21. 7544633
## manufacturer=Mercedes            32. 3932313 51. 870968 27. 3528764
## aux_years_sell=f. years_sell-(5, 6] 37. 4185137 37. 032258 16. 9054441

```

```

## aux_price=f.price-(14, 19. 5]
## aux_tax=f.tax-[0, 125]
## transmission=f.Trans-Automatic
## aux_mpg=f.mpg-(53. 3, 61. 4]
## transmission=f.Trans-SemiAuto
## aux_price=f.price-(19. 5, 26]
## manufacturer=BMW
## fuelType=f.Fuel-Hybrid
## manufacturer=Audi
## aux_tax=f.tax-(125, 145]
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05]
## aux_price=f.price-(26, 110]
## aux_years_sell=f.years_sell-(6, 23]
## aux_price=f.price-[0. 65, 14]
## aux_mpg=f.mpg-[8. 8, 44. 8]
## aux_mileage=f.mileage-(6e+03, 1. 65e+04]
## transmission=f.Trans-Manual
## aux_mpg=f.mpg-(44. 8, 53. 3]
## manufacturer=VW
## aux_mileage=f.mileage-[4, 6e+03]
## fuelType=f.Fuel-Petrol
## aux_years_sell=f.years_sell-[2, 3]
##
## aux_mileage=f.mileage-(1. 65e+04, 3. 35e+04]
## fuelType=f.Fuel-Diesel
## aux_years_sell=f.years_sell-(3, 5]
## aux_mpg=f.mpg-(61. 4, 88. 3]
## manufacturer=Mercedes
## aux_years_sell=f.years_sell-(5, 6]
## aux_price=f.price-(14, 19. 5]
## aux_tax=f.tax-[0, 125]
## transmission=f.Trans-Automatic
## aux_mpg=f.mpg-(53. 3, 61. 4]
## transmission=f.Trans-SemiAuto
## aux_price=f.price-(19. 5, 26]
## manufacturer=BMW
## fuelType=f.Fuel-Hybrid
## manufacturer=Audi
## aux_tax=f.tax-(125, 145]
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05]
## aux_price=f.price-(26, 110]
## aux_years_sell=f.years_sell-(6, 23]
## aux_price=f.price-[0. 65, 14]
## aux_mpg=f.mpg-[8. 8, 44. 8]
## aux_mileage=f.mileage-(6e+03, 1. 65e+04]
## transmission=f.Trans-Manual
## aux_mpg=f.mpg-(44. 8, 53. 3]
## manufacturer=VW
## aux_mileage=f.mileage-[4, 6e+03]
## fuelType=f.Fuel-Petrol
## aux_years_sell=f.years_sell-[2, 3]
##
## $ '7 '
##
## aux_years_sell=f.years_sell-(6, 23]
## aux_mileage=f.mileage-(3. 35e+04, 1. 16e+05]
## aux_price=f.price-[0. 65, 14]
## aux_tax=f.tax-(145, 580]
## aux_mpg=f.mpg-(61. 4, 88. 3]
## fuelType=f.Fuel-Diesel
## aux_years_sell=f.years_sell-(5, 6]
## transmission=f.Trans-Manual
## aux_tax=f.tax-[0, 125]
## aux_price=f.price-(14, 19. 5]
28. 2203390 42. 967742 26. 0083756
41. 9847328 14. 193548 5. 7747410
24. 3123336 35. 354839 24. 8402028
23. 2913669 33. 419355 24. 5095878
21. 1830986 48. 516129 39. 1227683
21. 7748562 34. 193548 26. 8238924
21. 1622807 24. 903226 20. 1013886
0. 0000000 0. 0000000 0. 9257218
12. 6041667 15. 612903 21. 1593564
15. 3279786 59. 096774 65. 8584968
12. 2923588 14. 322581 19. 9030196
9. 1066782 13. 548387 25. 4132687
5. 6105611 4. 387097 13. 3568437
7. 2948328 9. 290323 21. 7544633
7. 8152753 11. 354839 24. 8181618
6. 6721582 10. 451613 26. 7577695
7. 6452599 16. 129032 36. 0370289
6. 0213415 10. 193548 28. 9177871
4. 1432584 7. 612903 31. 3863787
0. 3330558 0. 516129 26. 4712365
3. 6622584 9. 290323 43. 3325986
0. 4305705 1. 032258 40. 9521710
p. value v. test
3. 337919e-213 31. 163415
4. 459588e-120 23. 299746
6. 518196e-77 18. 562040
1. 287292e-58 16. 142265
6. 228426e-58 16. 044675
1. 197608e-51 15. 119880
7. 913032e-30 11. 344330
1. 000515e-22 9. 811915
5. 445100e-13 7. 213702
7. 222057e-10 6. 161146
5. 505570e-09 5. 831123
6. 207012e-07 4. 984662
3. 332463e-04 3. 587983
3. 683043e-04 -3. 561812
2. 028692e-05 -4. 261709
1. 622508e-05 -4. 311364
1. 086008e-05 -4. 399300
1. 676024e-18 -8. 777195
3. 169007e-19 -8. 962673
2. 496693e-23 -9. 950997
2. 668963e-24 -10. 171093
6. 809665e-34 -12. 135972
1. 424694e-40 -13. 336238
1. 711519e-42 -13. 662027
6. 368853e-67 -17. 282533
2. 462196e-107 -22. 007105
1. 324903e-113 -22. 652275
2. 849808e-183 -28. 869916

## Cla/Mod Mod/Cla Global
83. 0033003 65. 8376963 13. 3568437
76. 4119601 90. 3141361 19. 9030196
41. 9452888 54. 1884817 21. 7544633
35. 6643357 60. 0785340 28. 3667622
34. 5491388 44. 6335079 21. 7544633
24. 2159587 79. 8429319 55. 5212696
29. 3350717 29. 4502618 16. 9054441
23. 8532110 51. 0471204 36. 0370289
38. 5496183 13. 2198953 5. 7747410
23. 8983051 36. 9109948 26. 0083756

```

```

## manufacturer=BMW
## manufacturer=Audi
## manufacturer=VW
## aux_mpg=f.mpg-(44.8,53.3]
## fuelType=f.Fuel-Hybrid
## manufacturer=Mercedes
## transmission=f.Trans-SemiAuto
## aux_mpg=f.mpg-[8.8,44.8]
## aux_mileage=f.mileage-(1.65e+04,3.35e+04]
## fuelType=f.Fuel-Petrol
## aux_price=f.price-(19.5,26]
## aux_years_sell=f.years_sell-(3,5]
## aux_price=f.price-(26,110]
## aux_mileage=f.mileage-(6e+03,1.65e+04]
## aux_mileage=f.mileage-[4,6e+03]
## aux_tax=f.tax-(125,145]
## aux_years_sell=f.years_sell-[2,3]
##
## aux_years_sell=f.years_sell-(6,23]
## aux_mileage=f.mileage-(3.35e+04,1.16e+05]
## aux_price=f.price-[0.65,14]
## aux_tax=f.tax-(145,580]
## aux_mpg=f.mpg-(61.4,88.3]
## fuelType=f.Fuel-Diesel
## aux_years_sell=f.years_sell-(5,6]
## transmission=f.Trans-Manual
## aux_tax=f.tax-[0,125]
## aux_price=f.price-(14,19.5]
## manufacturer=BMW
## manufacturer=Audi
## manufacturer=VW
## aux_mpg=f.mpg-(44.8,53.3]
## fuelType=f.Fuel-Hybrid
## manufacturer=Mercedes
## transmission=f.Trans-SemiAuto
## aux_mpg=f.mpg-[8.8,44.8]
## aux_mileage=f.mileage-(1.65e+04,3.35e+04]
## fuelType=f.Fuel-Petrol
## aux_price=f.price-(19.5,26]
## aux_years_sell=f.years_sell-(3,5]
## aux_price=f.price-(26,110]
## aux_mileage=f.mileage-(6e+03,1.65e+04]
## aux_mileage=f.mileage-[4,6e+03]
## aux_tax=f.tax-(125,145]
## aux_years_sell=f.years_sell-[2,3]
##
## Link between the cluster variable and the quantitative variables
## =====
##          Eta2 P-value
## price 0.5187428      0
##
## Description of each cluster by quantitative variables
## =====
## $ '1 '
##          v.test Mean in category Overall mean sd in category Overall sd
## price 34.0206      31311.47     21432.75     8552.083     9014.2
##          p.value
## price 1.105008e-253
##
## $ '2 '
##          v.test Mean in category Overall mean sd in category Overall sd
## price -4.072999     20044.33     21432.75     5446.497     9014.2
##          p.value

```

```

## price 4.641166e-05
##
## $ '3 '
##      v. test Mean in category Overall mean sd in category Overall sd
## price 21.4862      27758.81      21432.75      7101.876      9014.2
##      p.value
## price 2.095881e-102
##
## $ '4 '
## NULL
##
## $ '5 '
##      v. test Mean in category Overall mean sd in category Overall sd
## price -23.2499      14592.72      21432.75      5035.004      9014.2
##      p.value
## price 1.426001e-119
##
## $ '6 '
##      v. test Mean in category Overall mean sd in category Overall sd
## price -3.888949      20285.96      21432.75      5683.481      9014.2
##      p.value
## price 0.0001006793
##
## $ '7 '
##      v. test Mean in category Overall mean sd in category Overall sd
## price -25.47222      13856.5      21432.75      4417.114      9014.2
##      p.value
## price 4.005977e-143

```

We start with the description of the categorical variables that characterize the clusters:

Cluster 1: Cluster one has cars with low mpg, few years\_sell, high price, low mileage and medium taxes. Also, cars with semiAuto transmission and cars with petrol as fuel predominate.

Cluster 2: Cluster two has cars with few years\_sell, low mileage, medium-low mpg and medium price. Also, cars with Manual transmission, VW cars and cars with petrol as fuel predominate.

Cluster 3: Cluster three has cars with few years\_sell, high price, medium taxes and low mileage. Also, cars with diesel as fuel, Mercedes cars and cars with semiAuto transmission predominate.

Cluster 4: Cluster four has cars medium-high mpg. Also, BMW cars, cars with Hybrid as fuel and cars with Automatic transmission predominate.

Cluster 5: Cluster five has cars with medium years\_sell, low price and medium-high mileage. Also, cars with petrol as fuel, VW cars and cars with Manual transmission predominate.

Cluster 6: Cluster six has cars with medium years\_sell, medium taxes and medium-high mileage. Also, cars with diesel as fuel and Mercedes cars predominate.

Cluster 7: Cluster seven has cars with many years\_sell, high mileage, low price and high taxes. Also, cars with diesel as fuel and cars with Manual transmission predominate.

Cluster 8: Cluster of multivariate outliers.

Regarding quantitative variables:

Cluster 1: As we can see price mean is greater in cluster 1 than in the overall mean. So we can say that cluster 1 is a cluster of expensive cars.

Cluster 2: As we can see price mean is lower in cluster 2 than in the overall mean. So we can say that cluster 2 is a cluster of cheap cars.

Cluster 3: As we can see price mean is greater in cluster 3 than in the overall mean. So we can say that cluster 3 is a cluster of expensive cars.

Cluster 4:

Cluster 5: As we can see price mean is lower in cluster 5 than in the overall mean. So we can say that cluster 5 is a cluster of cheap cars.

Cluster 6: As we can see price mean is lower in cluster 6 than in the overall mean. So we can say that cluster 6 is a cluster of cheap cars.

Cluster 7: As we can see price mean is lower in cluster 7 than in the overall mean. So we can say that cluster 7 is a cluster of cheap cars.

Cluster 8: Cluster of multivariate outliers.

```
res.hcmc$desc.ind$para
```

```
## Cluster: 1
##      160      426      432      471      501
## 0.2160579 0.2160579 0.2160579 0.2160579 0.2160579
##
## -----
## Cluster: 2
##      3498      3508      3516      3517      3518
## 0.3057169 0.3057169 0.3057169 0.3057169 0.3057169
##
## -----
## Cluster: 3
##      2532      2148      2348      2388      2403
## 0.1690883 0.3052584 0.3052584 0.3052584 0.3052584
##
## -----
## Cluster: 4
##      1798      1566      1910      1912      1196
## 0.5046765 0.5308641 0.6060830 0.6225019 0.8208252
##
## -----
## Cluster: 5
##      3321      3374      3643      3814      36
## 0.2480597 0.2480597 0.2502881 0.2502881 0.3025934
##
## -----
## Cluster: 6
##      2167      2239      2265      2650      2686
## 0.1562898 0.1562898 0.1562898 0.1562898 0.1562898
##
## -----
## Cluster: 7
##      3558      4083      4988      3278      4094
## 0.1830092 0.1830092 0.1830092 0.3034436 0.3458925
```

```
res.hcmc$desc.ind$dist
```

```
## Cluster: 1
##      35      40      46      69      91
## 1.513956 1.513956 1.513956 1.513956 1.513956
##
## -----
## Cluster: 2
##      4487      4010      4663      4009      4026
## 2.615673 2.385078 2.294474 1.981549 1.928640
##
## -----
## Cluster: 3
##      2266      2274      2285      2287      2384
## 1.24873 1.24873 1.24873 1.24873 1.24873
##
## -----
## Cluster: 4
##      1530      1989      2023      1910      1102
## 3.762245 3.762245 3.762245 3.681187 3.672630
##
## -----
## Cluster: 5
##      4011      174      317      1      73
## 1.953666 1.582628 1.582628 1.423799 1.423799
##
## -----
## Cluster: 6
```

```

##      1092      1279      1529      1544      1866
## 1.474618 1.474618 1.474618 1.474618 1.474618
##
## -----
## Cluster: 7
##      3552      3575      3590      3618      3647
## 1.804655 1.804655 1.804655 1.804655 1.804655

```

The parangons of cluster 1 are 160, 426, 432, 471 and 501. The parangons of cluster 2 are 3498, 3508, 3516, 3517 and 3518. The parangons of cluster 3 are 2532, 2148, 2348, 2388 and 2403. The parangons of cluster 4 are 1798, 1566, 1910, 1912 and 1196. The parangons of cluster 5 are 3321, 3374, 3643, 3814 and 36. The parangons of cluster 6 are 2167, 2239, 2265, 2650 and 2686. The parangons of cluster 7 are 3558, 4083, 4988, 3278 and 4094.

The distant ones of cluster 1 are 35, 40, 46, 69 and 91. The distant ones of cluster 2 are 4487, 4010, 4663, 4009 and 4026. The distant ones of cluster 3 are 2266, 2274, 2285, 2287 and 2384. The distant ones of cluster 4 are 1530, 1989, 2023, 1910 and 1102. The distant ones of cluster 5 are 4011, 174, 317, 1 and 73. The distant ones of cluster 6 are 1092, 1279, 1529, 1544 and 1866. The distant ones of cluster 7 are 3552, 3575, 3590, 3618 and 3647.

We have created a new variable indicating the cluster to which the individual belongs after the Hierarchical clustering.

```

df$clamHC<-8
df[ row.names(res.hcmc$data.clust), "clamHC" ]<-res.hcmc$data.clust$clust
df$clamHC<-factor(df$clamHC)
levels( df$clamHC ) <- paste0("claHCMC-", levels( df$clamHC ) )

```

## 6.6 Comparisons

```

tt<-table(df$clamHC, df$claHC)
100*sum(diag(tt)/sum(tt))

```

```

## [1] 41.72

```

We have a concordance of the 41.72% so we can say that they are different, if we had a greater concordance, this would mean that they would be more similar.

```

ta<-table(df$clamHC, df$claKM)
100*sum(diag(ta)/sum(ta))

```

```

## [1] 39.58

```

We have a concordance of the 39.58% so we can say that they are different, if we had a greater concordance, this would mean that they would be more similar.

# USED CAR PRICES CASE STUDY

Deliverable 3: General and Binary/Logistic Regression Models

Miquel Parra i Xavier Alaman

May 29, 2022

## Contents

<b>1 R libraries imports, useful functions and data loading</b>	<b>1</b>
1.1 Load Required Packages . . . . .	2
1.2 Sample load . . . . .	2
1.3 Preparing the data . . . . .	2
<b>2 Linear Models</b>	<b>3</b>
2.1 Only Numeric variables . . . . .	3
2.1.1 Target transformation . . . . .	5
2.1.2 Explanatory variables transformation . . . . .	8
2.2 Including factors . . . . .	10
2.2.1 Significant factors . . . . .	10
2.3 Interactions . . . . .	16
2.3.1 Factors interaction . . . . .	16
2.3.2 Factor and covariate interaction . . . . .	19
2.4 Best model selection . . . . .	22
2.5 Diagnostics . . . . .	24
<b>3 Binary/Logistic Regression Models</b>	<b>28</b>
3.1 Dividing/Splitting the sample . . . . .	28
3.2 Only numeric variables . . . . .	28
3.2.1 Transformations . . . . .	31
3.3 Including factors . . . . .	32
3.4 Interactions . . . . .	35
3.4.1 Factors interaction . . . . .	35
3.4.2 Factor and covariate interaction . . . . .	37
3.5 Best model selection . . . . .	39
3.6 Diagnostics . . . . .	40
3.7 Prediction . . . . .	45
3.8 Confusion matrix . . . . .	47

## 1 R libraries imports, useful functions and data loading

In this first section we will load all required packages and libraries, and load our data.

## 1.1 Load Required Packages

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

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

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

## 1.2 Sample load

This sample has a new variable called engineSize\_num that has three categories: small\_engine (engineSize<=2), medium\_engine (2<engineSize<=3) and large\_engine (engineSize>3).

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

# Clean workspace
rm(list=ls())

# Users file path
miquel_fp <- "C:/Users/Miquel/Documents/GitHub/ADEI/"
xavi_fp <- "~/Documents/FIB/ADEI/ADEI/"
filepath <- xavi_fp
filepath <- miquel_fp
# Set working directory
setwd(filepath)

# Load data from file
load(paste0(filepath, "MyOldCars-5000Clean2.RData"))
# Index reset
row.names(df) <- NULL
```

## 1.3 Preparing the data

We cannot have values equal to zero in the variables because in case we apply a logarithmic transformation to them, this would give an error since the logarithm of zero is undefined.

```
names(df)

##  [1] "model"          "year"           "price"          "transmission"
##  [5] "mileage"        "fuelType"        "tax"            "mpg"
##  [9] "engineSize"      "manufacturer"    "years_sell"     "engineSize_num"
## [13] "totalMOE"        "aux_price"       "aux_mileage"    "aux_tax"
## [17] "aux_mpg"         "aux_years_sell"  "Audi"          "mout"

vars_con<-names(df)[c(5,7,8,11)]
vars_res<-names(df)[c(3,19)]
vars_dis<-names(df)[c(1,2,4,6,9,10,12,14,15,16,17,18)]
11<-which(df$years_sell==0);11
```

```

## integer(0)

df$years_sell[11] <- 0.5
11 <- which(df$tax==0); 11

## integer(0)

df$tax[11] <- 0.5
11 <- which(df$mileage==0); 11

## integer(0)

df$mileage[11] <- 0.5
11 <- which(df$mpg==0); 11

## integer(0)

df$mpg[11] <- 0.5

```

## 2 Linear Models

We do linear regression in order to predict the value of price variable based on/according to the values of other variables.

### 2.1 Only Numeric variables

```

m0<-lm(price~1,data=df)
m1<-lm(price~mileage+tax+mpg+years_sell,data=df)
anova(m0,m1)

## Analysis of Variance Table
##
## Model 1: price ~ 1
## Model 2: price ~ mileage + tax + mpg + years_sell
##   Res.Df   RSS Df  Sum of Sq   F   Pr(>F)
## 1    4999 5.8816e+11
## 2    4995 2.7995e+11  4 3.0822e+11 1374.9 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m1)

##
## Call:
## lm(formula = price ~ mileage + tax + mpg + years_sell, data = df)
##
## Residuals:
##     Min      1Q      Median      3Q      Max
## -17976   -5039    -312     3478    73806
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.831e+04  1.634e+03 35.691 < 2e-16 ***

```

```

## mileage      -2.498e-02  7.745e-03  -3.225  0.00127  **
## tax          -2.057e+01  9.807e+00  -2.098  0.03597  *
## mpg          -4.509e+02  1.028e+01  -43.854  < 2e-16 *** 
## years_sell  -1.977e+03  7.503e+01  -26.349  < 2e-16 *** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7486 on 4995 degrees of freedom
## Multiple R-squared:  0.524, Adjusted R-squared:  0.5237 
## F-statistic: 1375 on 4 and 4995 DF, p-value: < 2.2e-16

```

```
vif(m1)
```

```

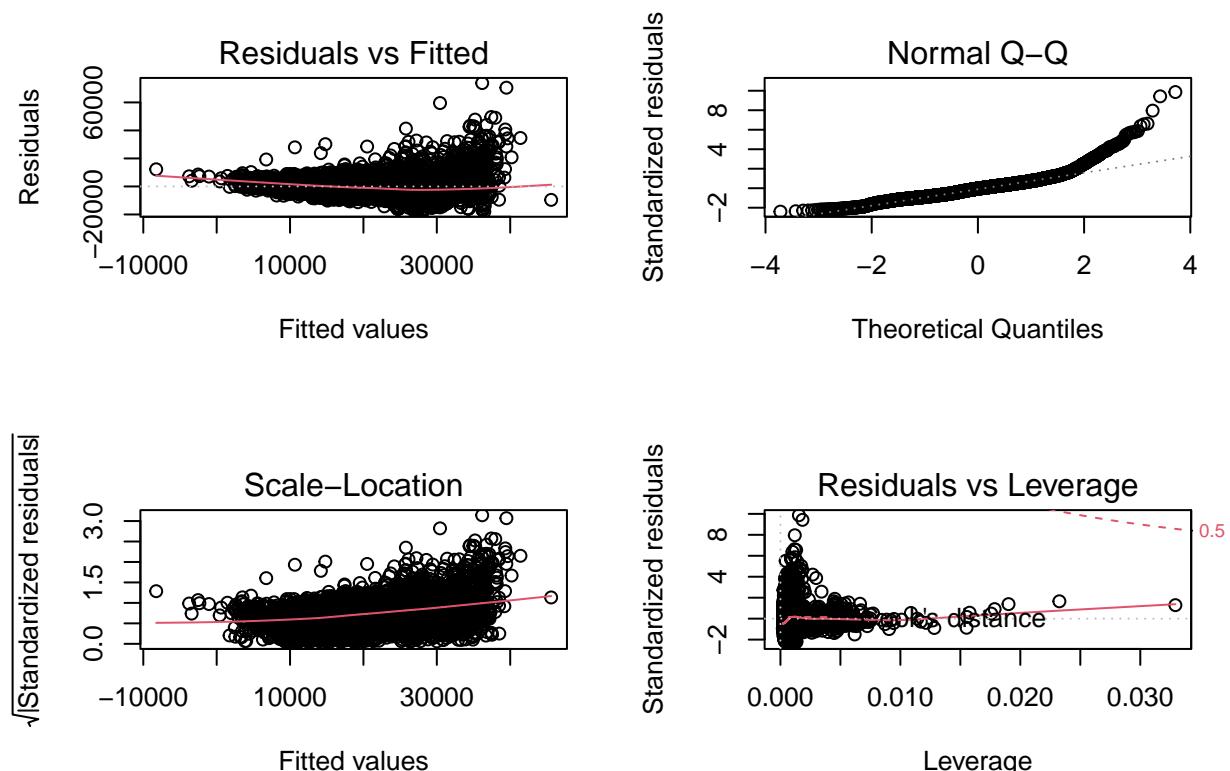
##      mileage          tax          mpg years_sell
## 2.462933  1.151081  1.245932  2.300784

```

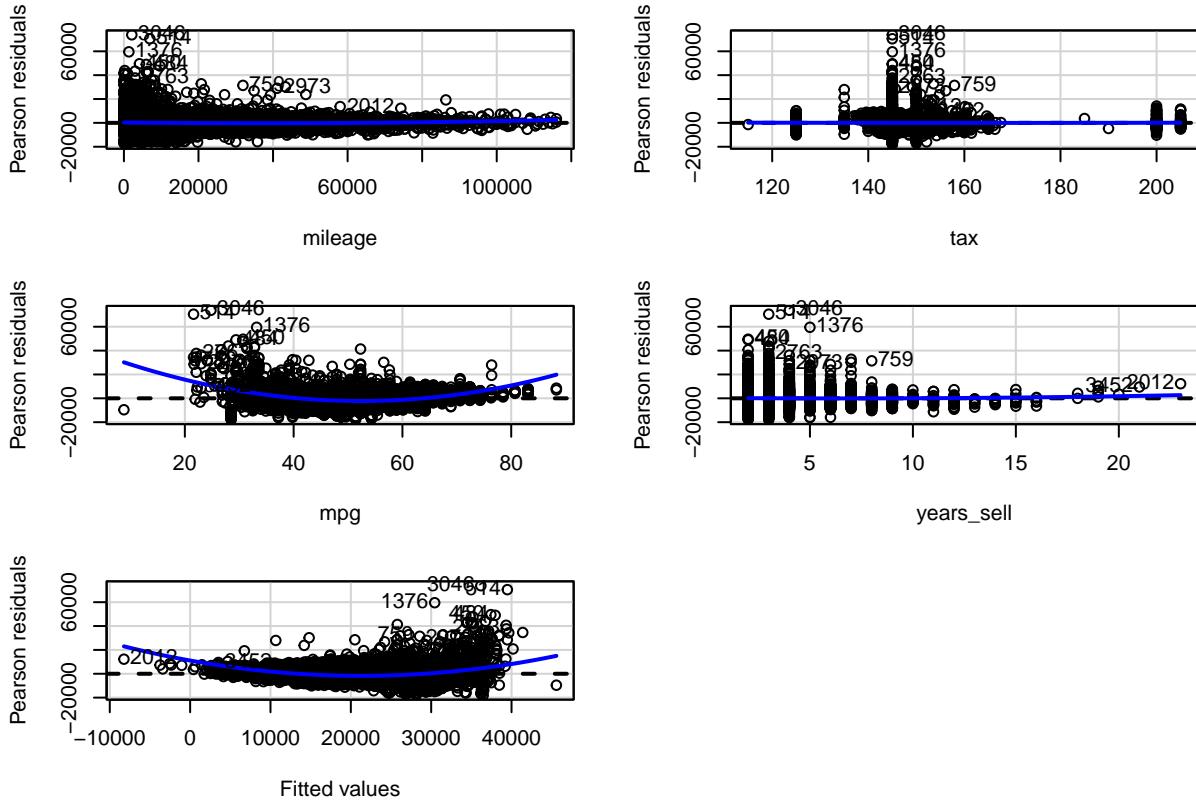
```

par(mfrow=c(2,2))
plot(m1,id.n=0)

```



```
residualPlots(m1,id=list(method=cooks.distance(m1),n=10))
```



```
##          Test stat Pr(>|Test stat|) 
## mileage      2.9460    0.003234 ** 
## tax          0.3137    0.753756  
## mpg         28.6974   < 2.2e-16 *** 
## years_sell   1.1526    0.249143  
## Tukey test   20.7127   < 2.2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see some or all of the regressors are useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

As we can see all the chosen variables have coefficients different from zero, i.e. they are useful, because their p-values are less than 0.05. These explanatory variables explain a 52.4 % of the target's variability. Also, we can see that there are no collinear variables, i.e., highly correlated variables.

However, we can see that residuals are neither homoscedastic nor normal. So, we have to apply transformations. Also, the clearest nonlinear variable is mpg.

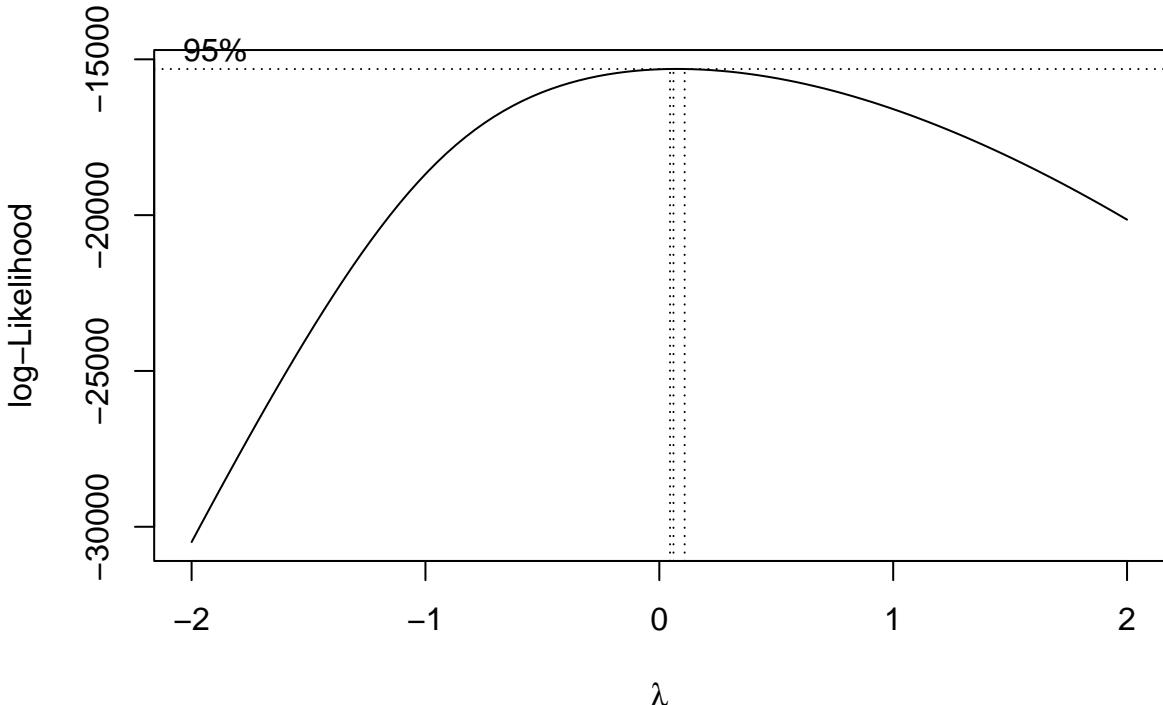
### 2.1.1 Target transformation

```
library(MASS)

## 
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
## 
##     select

# Target variable transformation?
par(mfrow=c(1,1))
boxcox(price~mileage+tax+mpg+years_sell,data=df)
```



```

# Lambda=0 - log transformation is needed

# New model:
m2<-lm(log(price)~mileage+tax+mpg+years_sell,data=df)
summary(m2)

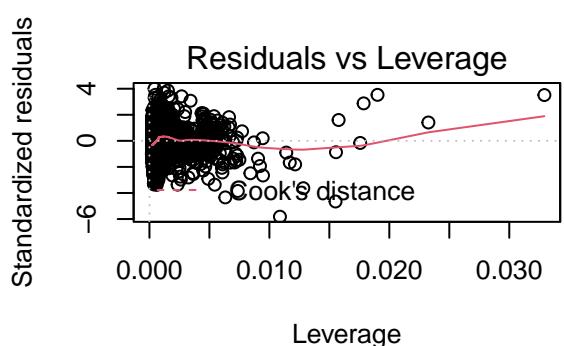
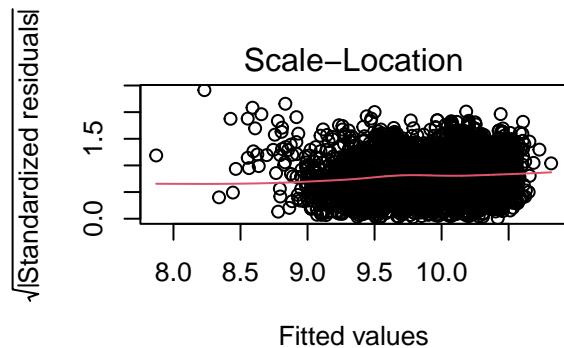
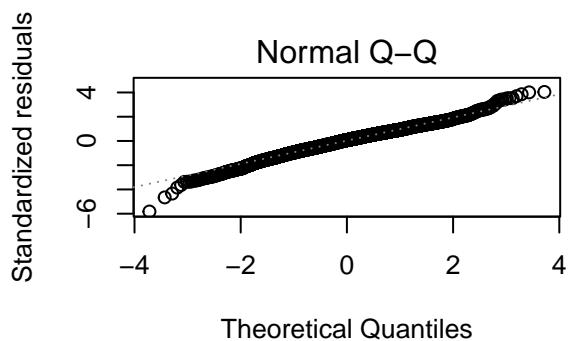
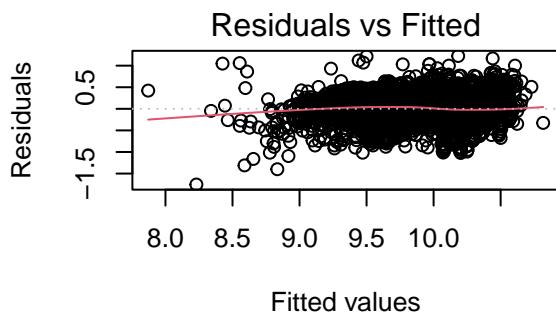
## 
## Call:
## lm(formula = log(price) ~ mileage + tax + mpg + years_sell, data = df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.75369 -0.18786  0.02125  0.20282  1.22401 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.124e+01  6.611e-02 170.067 < 2e-16 ***
## mileage     -1.464e-06  3.134e-07 -4.673 3.04e-06 ***
## tax          6.297e-04  3.968e-04   1.587  0.113    
## mpg          -1.635e-02  4.161e-04 -39.297 < 2e-16 ***
## years_sell  -1.212e-01  3.036e-03 -39.914 < 2e-16 *** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.3029 on 4995 degrees of freedom
## Multiple R-squared:  0.6118, Adjusted R-squared:  0.6115 
## F-statistic: 1968 on 4 and 4995 DF,  p-value: < 2.2e-16

vif(m2)

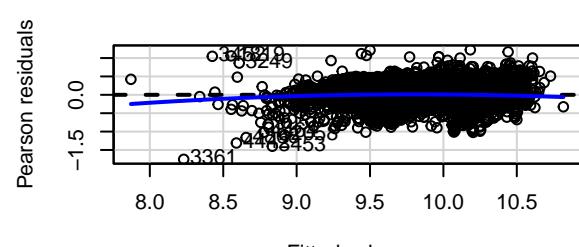
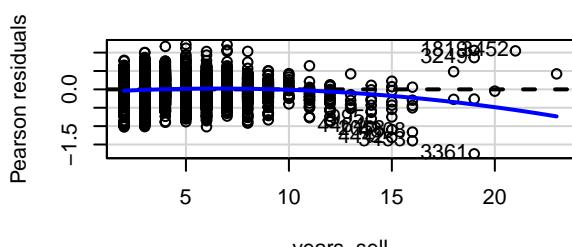
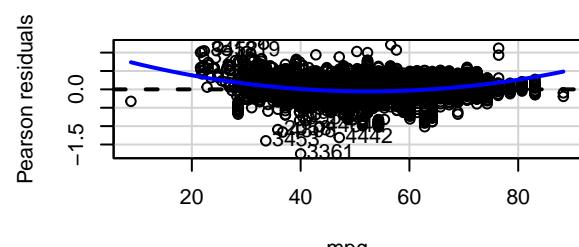
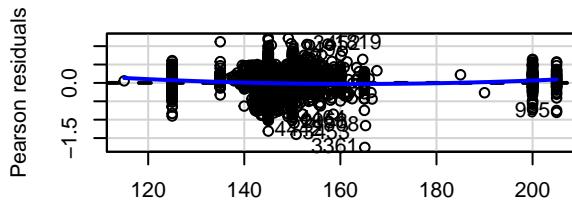
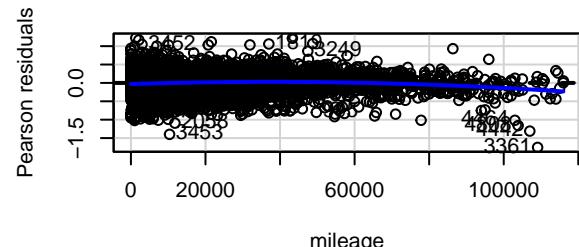
##      mileage          tax          mpg      years_sell
## 2.462933  1.151081  1.245932  2.300784

```

```
par(mfrow=c(2,2))
plot(m2, id.n=0)
```



```
residualPlots(m2, id=list(method=cooks.distance(m2), n=10))
```



```
##          Test stat  Pr(>|Test stat|)
```

```

## mileage      -6.8930      6.142e-12 ***
## tax          5.7864      7.628e-09 ***
## mpg          16.4715      < 2.2e-16 ***
## years_sell  -7.8926      3.611e-15 ***
## Tukey test   -3.3891      0.0007012 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

This model is better than the previous one.

As we can see the transformation needed is a logarithmic transformation, because lambda is almost equal to zero.

Now, we can see that tax variable is not useful, because its p-value is greater than 0.05, so we should remove it but we won't do it because applying a transformation to it we will make it useful. However, the explanatory variables explain a 61.18 % of the target's variability. There are still no collinear variables because they have not changed.

However, the residuals continue to be neither homoscedastic nor normal. So, more transformations are needed. Also, the clearest nonlinear variables are mpg and years\_sell.

### 2.1.2 Explanatory variables transformation

```
boxTidwell(log(price)~mileage+tax+mpg, data=df[!df$mout=="YesMOut",])
```

```

##           MLE of lambda Score Statistic (z)  Pr(>|z|)
## mileage      0.73871      4.8992 9.623e-07 ***
## tax          25.87560      3.9858 6.726e-05 ***
## mpg          -0.33882      13.0103 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## iterations = 13

```

We need to cube the tax variable, take the square root of the mileage variable and take the square root and raise to minus one the mpg variable.

```
m3<-lm(log(price)~sqrt(mileage)+poly(tax,3)+ I(mpg^(-1/2))+years_sell,data=df[!df$mout=="YesMOut",])
```

```
summary(m3)
```

```

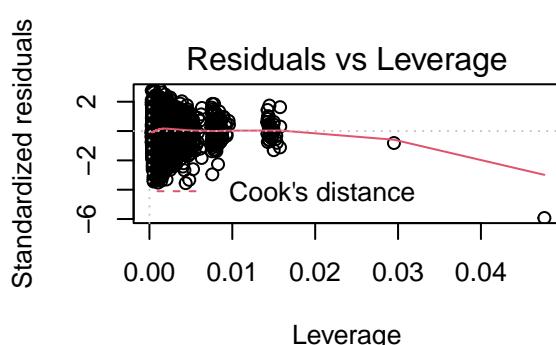
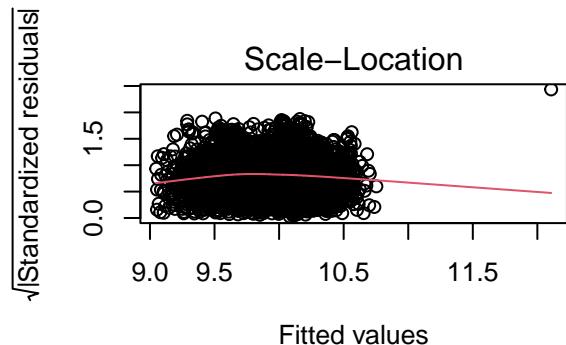
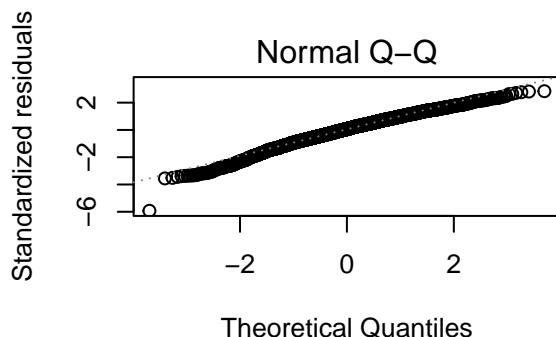
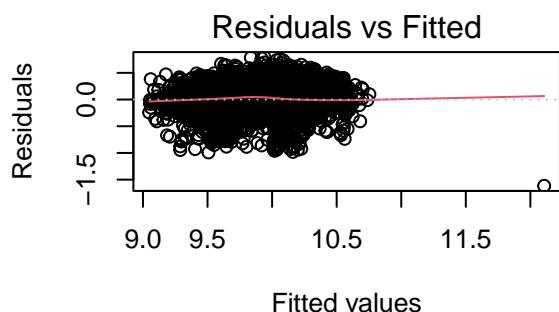
## 
## Call:
## lm(formula = log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) +
##     years_sell, data = df[!df$mout == "YesMOut", ])
## 
## Residuals:
##      Min      1Q  Median      3Q      Max
## -1.6155 -0.1705  0.0188  0.1962  0.7924
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 9.0369252  0.0511414 176.705 < 2e-16 ***
## sqrt(mileage) -0.0005373  0.0001247 -4.310 1.67e-05 ***
## poly(tax, 3)1  1.1805230  0.2926125  4.034 5.56e-05 ***
## poly(tax, 3)2  2.1910644  0.2920661  7.502 7.52e-14 ***
## poly(tax, 3)3 -1.0113561  0.3043684 -3.323 0.000898 ***
## I(mpg^(-1/2)) 10.1768838  0.3084791 32.991 < 2e-16 ***
## years_sell    -0.1125324  0.0045400 -24.787 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2793 on 4529 degrees of freedom
## Multiple R-squared:  0.5651, Adjusted R-squared:  0.5645 
## F-statistic: 980.6 on 6 and 4529 DF,  p-value: < 2.2e-16

```

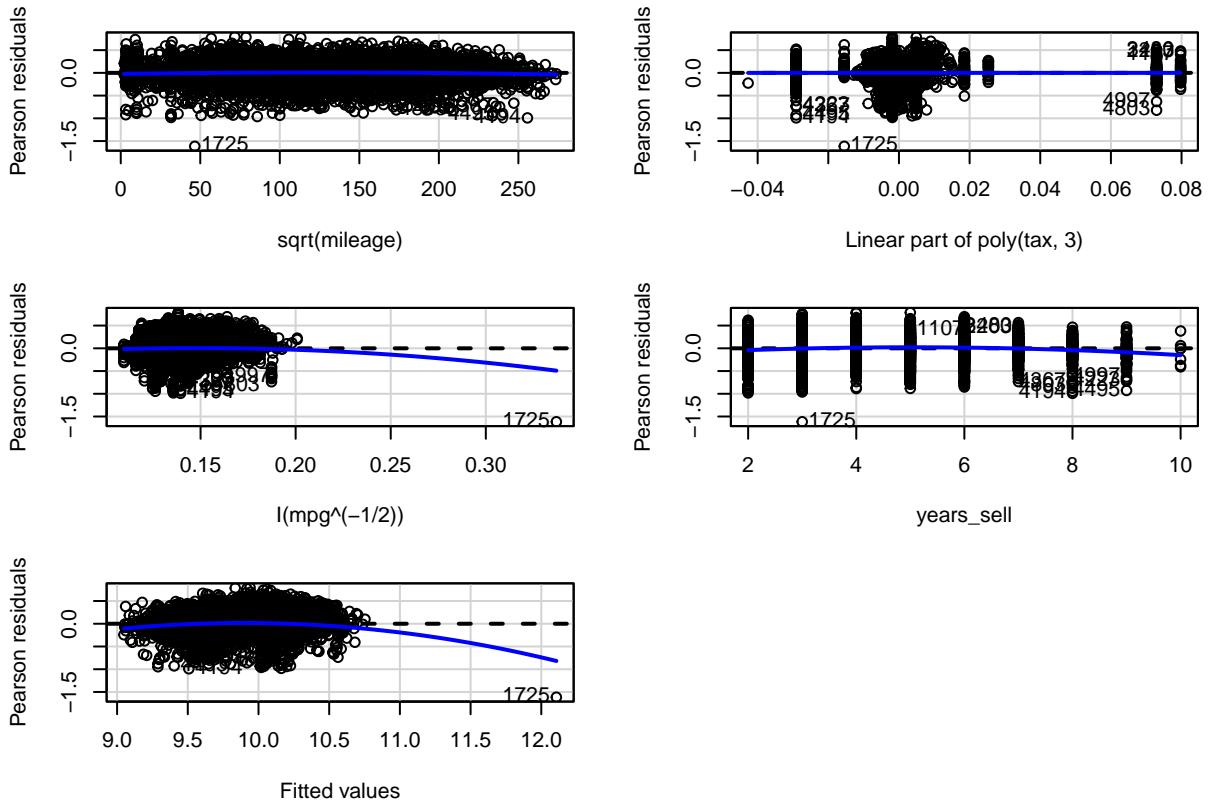
```
vif(m3)
```

```
##          GVIF Df GVIF^(1/(2*Df))
## sqrt(mileage) 3.474396  1      1.863973
## poly(tax, 3)  1.392537  3      1.056739
## I(mpg^(-1/2)) 1.395796  1      1.181438
## years_sell     3.585436  1      1.893525
```

```
par(mfrow=c(2,2))
plot(m3, id.n=0)
```



```
residualPlots(m3, id=list(method=cooks.distance(m3), n=10))
```



```
##                Test stat Pr(>|Test stat|)  
##  sqrt(mileage)   -1.9739      0.04846  *  
##  poly(tax, 3)  
##  I(mpg^(-1/2))  -2.1485      0.03173  *  
##  years_sell     -5.8106      6.652e-09 ***  
##  Tukey test      -6.6060      3.949e-11 ***  
##  ---  
##  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
```

This model is better than the previous one.

As we can see all the chosen variables have coefficients different from zero, i.e. they are useful, because their p-values are less than 0.05.

These explanatory variables explain a 56.51 % of the target's variability. We can say that the mileage and years\_sell variables are collinear, i.e., they are highly correlated. However, this is because we have adapted them.

Also, we can see that residuals are now homoscedastic and normal. However, we can see some influential observation. Also, there are no excessively non-linear variables.

## 2.2 Including factors

### 2.2.1 Significant factors

```
m4 <- update(m3, ~.+transmission+fuelType+manufacturer+engineSize_num, data=df[!df$mout=="YesMOut",])
m4pet<-update(m3, ~.+fuelType+manufacturer+engineSize_num, data=df[!df$mout=="YesMOut",])
anova(m4pet,m4)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           fuelType + manufacturer + engineSize_num
## Model 2: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
```

```

##      transmission + fuelType + manufacturer + engineSize_num
##  Res.Df    RSS Df Sum of Sq   F   Pr(>F)
## 1    4521 150.22
## 2    4519 131.03  2    19.195 331 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

As we can see transmission variable is useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

```

m4 <- update(m3, ~.+transmission+fuelType+manufacturer+engineSize_num,data=df[!df$mout=="YesMOut",])
m4pet<-update(m3, ~.+transmission+manufacturer+engineSize_num,data=df[!df$mout=="YesMOut",])
anova(m4pet,m4)

```

```

## Analysis of Variance Table
##
## Model 1: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + manufacturer + engineSize_num
## Model 2: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + fuelType + manufacturer + engineSize_num
##  Res.Df    RSS Df Sum of Sq   F   Pr(>F)
## 1    4522 171.63
## 2    4519 131.03  3    40.599 466.73 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

As we can see fuelType variable is useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

```

m4 <- update(m3, ~.+transmission+fuelType+manufacturer+engineSize_num,data=df[!df$mout=="YesMOut",])
m4pet<-update(m3, ~.+transmission+fuelType+engineSize_num,data=df[!df$mout=="YesMOut",])
anova(m4pet,m4)

```

```

## Analysis of Variance Table
##
## Model 1: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + fuelType + engineSize_num
## Model 2: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + fuelType + manufacturer + engineSize_num
##  Res.Df    RSS Df Sum of Sq   F   Pr(>F)
## 1    4522 161.72
## 2    4519 131.03  3    30.692 352.84 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

As we can see manufacturer variable is useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

```

m4 <- update(m3, ~.+transmission+fuelType+manufacturer+engineSize_num,data=df[!df$mout=="YesMOut",])
m4pet<-update(m3, ~.+transmission+fuelType+manufacturer,data=df[!df$mout=="YesMOut",])
anova(m4pet,m4)

```

```

## Analysis of Variance Table
##
## Model 1: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + fuelType + manufacturer
## Model 2: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + fuelType + manufacturer + engineSize_num
##  Res.Df    RSS Df Sum of Sq   F   Pr(>F)
## 1    4521 136.33
## 2    4519 131.03  2    5.2963 91.331 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

As we can see engineSize\_num variable is useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

```
m4 <- update(m3, ~.+transmission+fuelType+manufacturer+engineSize_num, data=df[!df$mout=="YesMOut",])
anova(m3, m4)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell
## Model 2: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + fuelType + manufacturer + engineSize_num
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1   4529 353.19
## 2   4519 131.03 10    222.16 766.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m4)
```

```
##
## Call:
## lm(formula = log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) +
##     years_sell + transmission + fuelType + manufacturer + engineSize_num,
##     data = df[!df$mout == "YesMOut", ])
##
## Residuals:
##   Min     1Q     Median     3Q     Max
## -1.94903 -0.10563  0.00462  0.11013  0.57122
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                9.205e+00  6.778e-02 135.800 < 2e-16 ***
## sqrt(mileage)             -9.844e-04  7.645e-05 -12.875 < 2e-16 ***
## poly(tax, 3)1              9.227e-01  1.798e-01   5.132 2.99e-07 ***
## poly(tax, 3)2              7.230e-01  1.796e-01   4.026 5.77e-05 ***
## poly(tax, 3)3              -1.600e-01  1.873e-01  -0.854 0.393064
## I(mpg^(-1/2))              1.122e+01  2.284e-01   49.105 < 2e-16 ***
## years_sell                 -9.331e-02  2.832e-03  -32.945 < 2e-16 ***
## transmissionf.Trans-SemiAuto 1.742e-01  6.905e-03  25.224 < 2e-16 ***
## transmissionf.Trans-Automatic 1.470e-01  7.657e-03  19.192 < 2e-16 ***
## fuelTypef.Fuel-Hybrid       -5.057e-02  2.700e-02  -1.873 0.061110 .
## fuelTypef.Fuel-Other         -8.630e-02  5.428e-02  -1.590 0.111956
## fuelTypef.Fuel-Petrol        -2.359e-01  6.309e-03  -37.385 < 2e-16 ***
## manufacturerBMW             -4.467e-02  8.092e-03  -5.521 3.57e-08 ***
## manufacturerMercedes        2.941e-03  8.143e-03   0.361 0.718036
## manufacturerVW              -2.058e-01  7.289e-03  -28.239 < 2e-16 ***
## engineSize_nummedium_engine -1.914e-01  5.215e-02  -3.671 0.000245 ***
## engineSize_numsmall_engine  -2.936e-01  5.235e-02  -5.609 2.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1703 on 4519 degrees of freedom
## Multiple R-squared:  0.8386, Adjusted R-squared:  0.8381
## F-statistic: 1468 on 16 and 4519 DF,  p-value: < 2.2e-16
```

```
vif(m4)
```

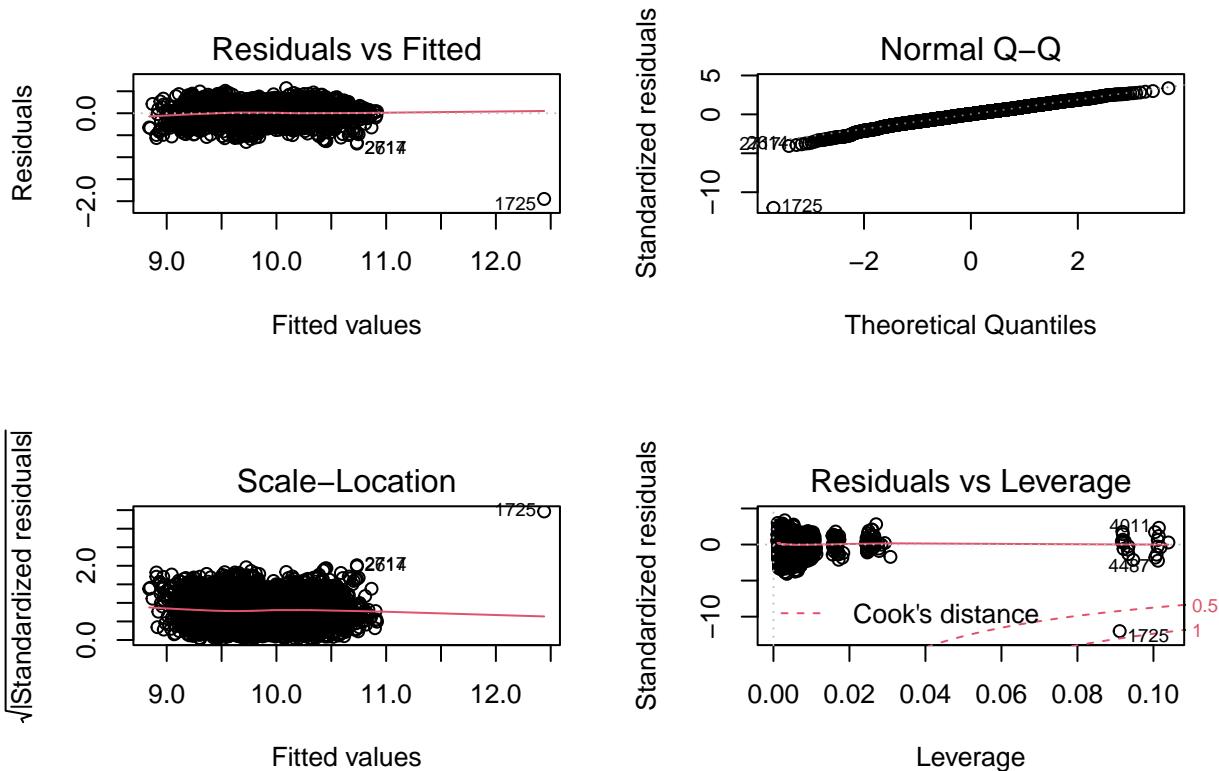
```
##          GVIF Df GVIF^(1/(2*Df))
## sqrt(mileage) 3.514030 1      1.874575
## poly(tax, 3)  1.462268 3      1.065380
## I(mpg^(-1/2)) 2.057817 1      1.434509
## years_sell    3.753243 1      1.937329
```

```

## transmission 1.533167 2      1.112749
## fuelType      1.588838 3      1.080223
## manufacturer 1.577555 3      1.078940
## engineSize_num 1.582246 2      1.121550

par(mfrow=c(2,2))
plot(m4)

```



```
marginalModelPlots(m4)
```

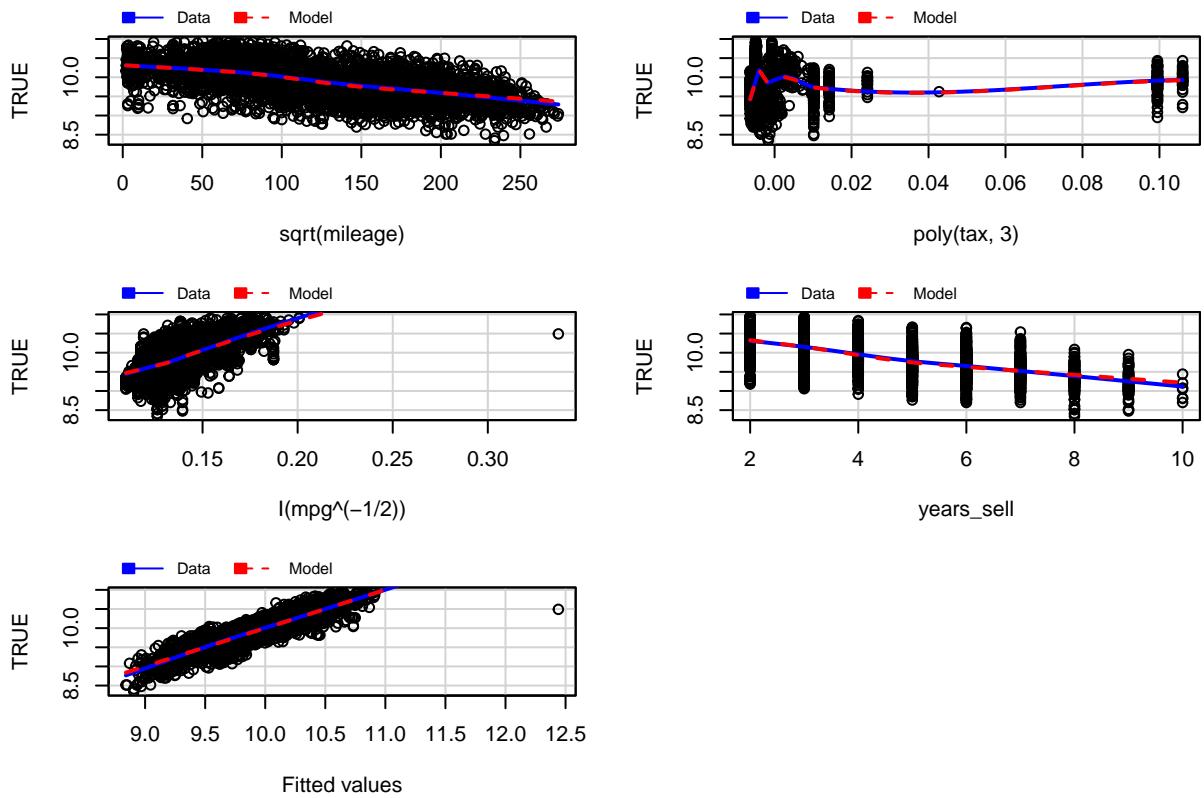
```

## Warning in mmmps(...): Splines and/or polynomials replaced by a fitted linear
## combination

## Warning in mmmps(...): Interactions and/or factors skipped

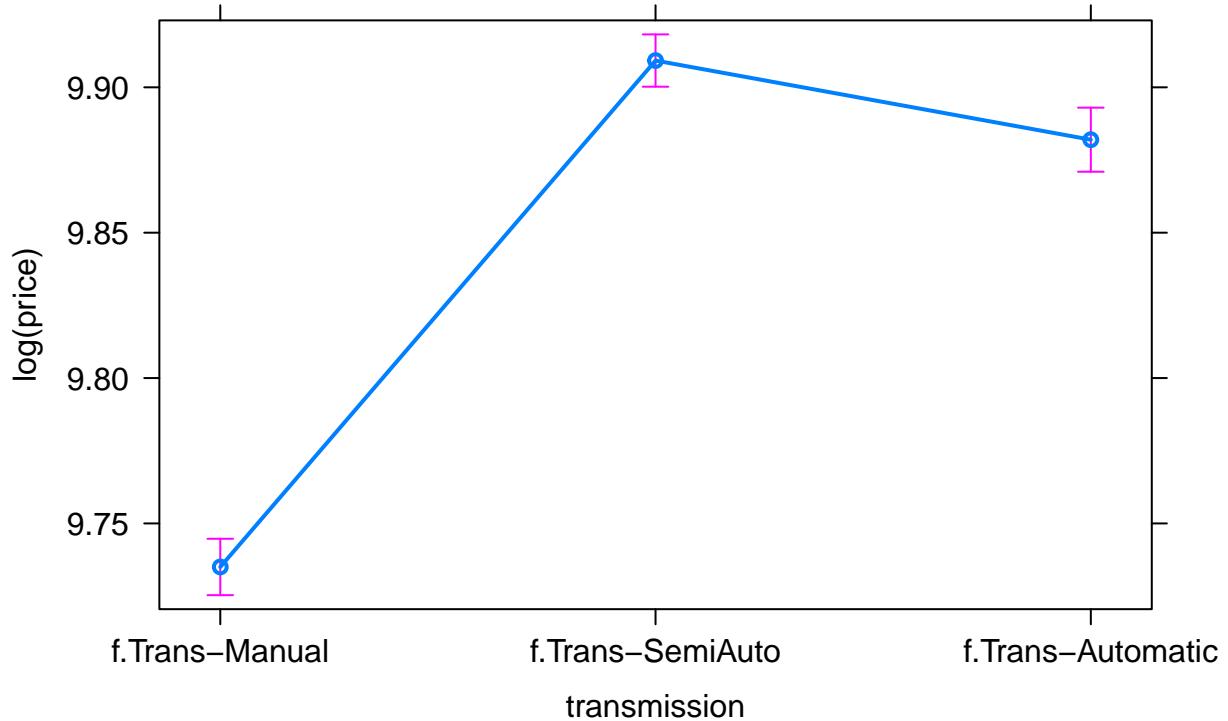
```

## Marginal Model Plots



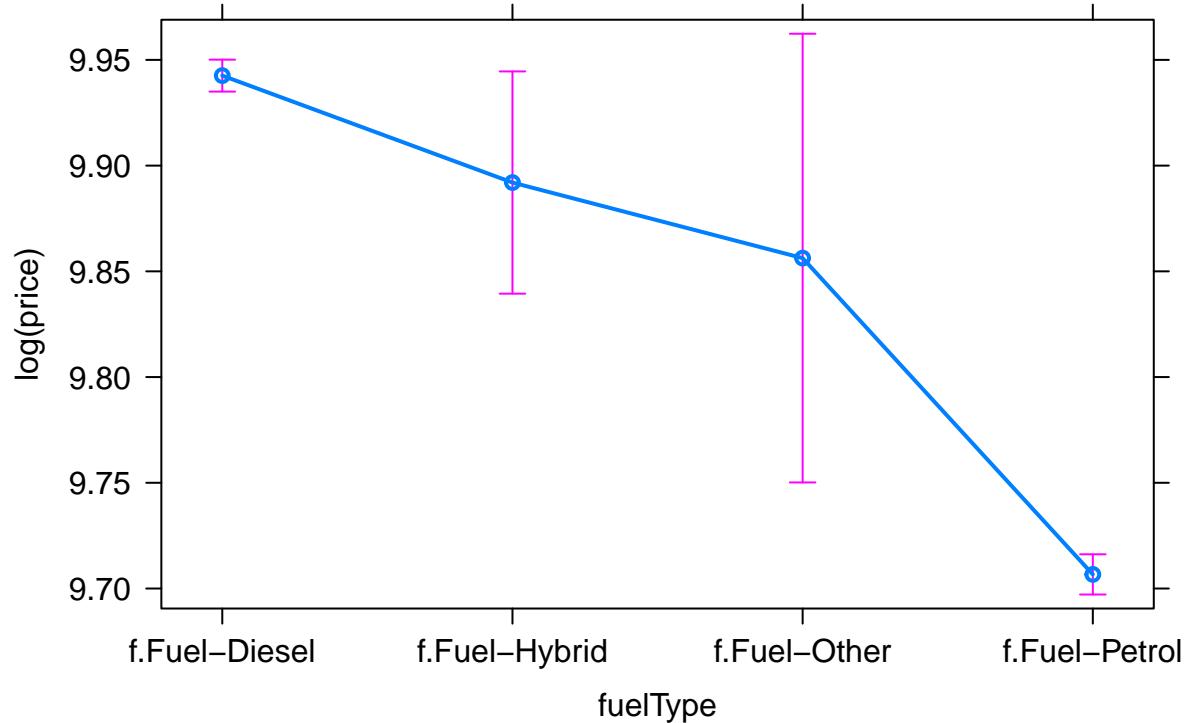
```
#residualPlots(m4, id=list(method=cooks.distance(m4), n=10))
plot(allEffects(m4), selection = 5)
```

## transmission effect plot



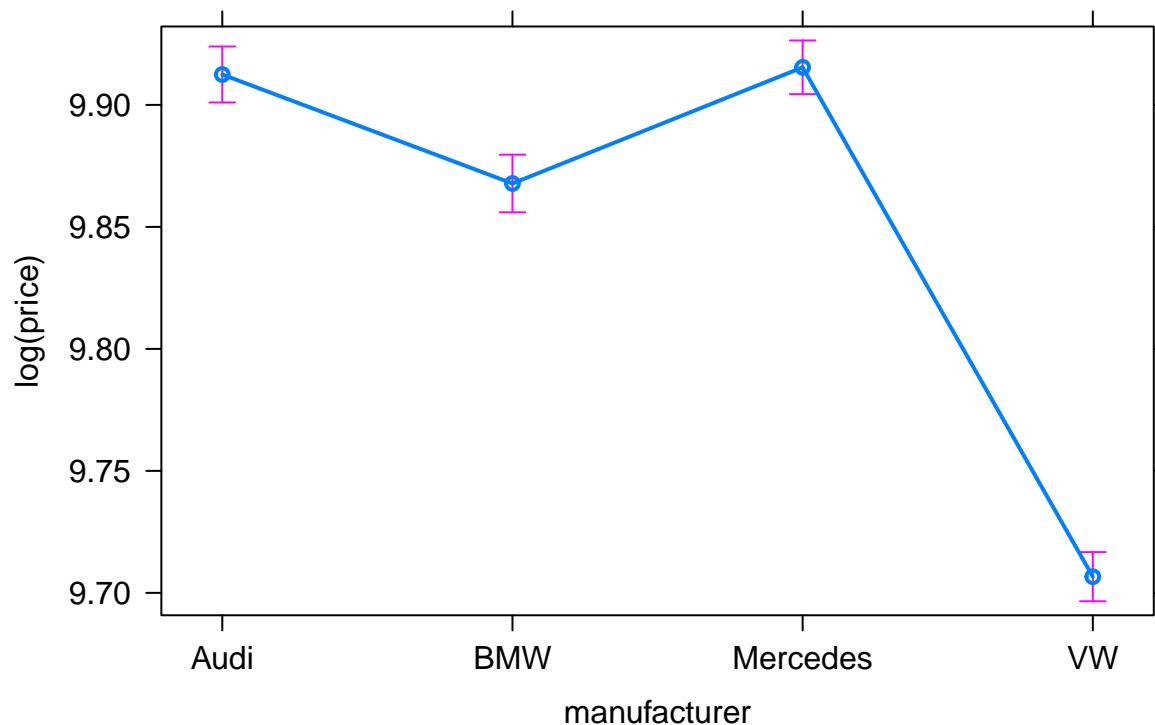
```
plot(allEffects(m4), selection = 6)
```

### **fuelType effect plot**



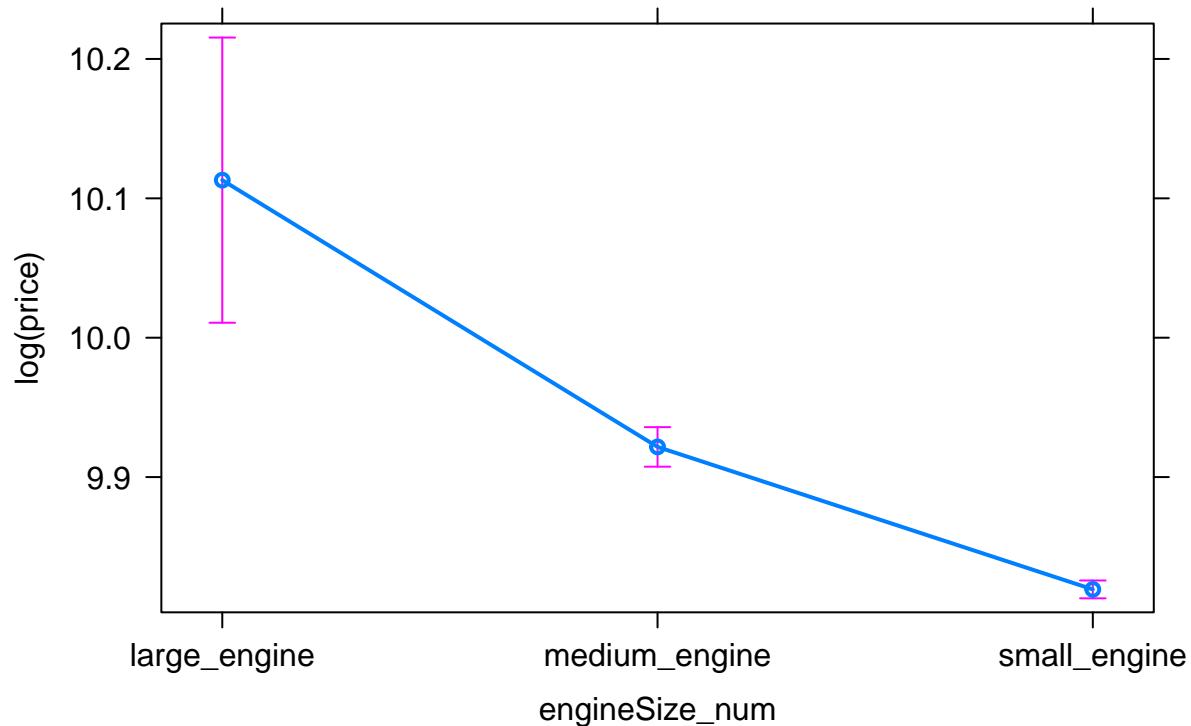
```
plot(allEffects(m4), selection = 7)
```

### **manufacturer effect plot**



```
plot(allEffects(m4), selection = 8)
```

### engineSize\_num effect plot



As we can see some or all of the new regressors are useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

These explanatory variables explain an 83.86 % of the target's variability.

We can say that the mileage and years\_sell variables are still collinear, i.e., they are highly correlated. However, this is because we have adapted them.

Also, we can see that residuals are still homoscedastic and normal. However, we can see some influential observation.

Also, we can see that the model captures the data well.

We can see: The fact that a car is semi-automatic makes the logarithm of the price increase by 0.17 units. The fact that a car is automatic makes the logarithm of the price increase by 0.15 units. The effect of being a hybrid car, an other car or a diesel car (baseline) is the same, because their p-values are greater than 0.05. The fact that a car is petrol makes the logarithm of the price decrease by 0.24 units. The fact that a car is BWM makes the logarithm of the price decrease by 0.045 units. The effect of being a Mercedes car and an Audi car (baseline) is the same, because its p-value is greater than 0.05. The effect that a car is VW makes the logarithm of the price decrease by 0.21 units. The effect that a car have medium\_engine makes the logarithm of the price decrease by 0.19 units. The effect that a car have small\_engine makes the logarithm of the price decrease by 0.29 units.

## 2.3 Interactions

### 2.3.1 Factors interaction

We are going to see if price variable (Y response) is related to fuelType (factor A) and aux\_tax (factor B) variables.

```
m5<-lm(log(price)~sqrt(mileage)+fuelType*aux_tax+poly(tax,3)+I(mpg^(-1/2))+years_sell+
  transmission+manufacturer+engineSize_num,data=df[!df$mout=="YesMOut",])
anova(m4,m5)
```

```
## Analysis of Variance Table
```

```

## 
## Model 1: log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) + years_sell +
##           transmission + fuelType + manufacturer + engineSize_num
## Model 2: log(price) ~ sqrt(mileage) + fuelType * aux_tax + poly(tax, 3) +
##           I(mpg^(-1/2)) + years_sell + transmission + manufacturer +
##           engineSize_num
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   4519 131.03
## 2   4513 130.16  6   0.86818 5.017 3.905e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
AIC(m4,m5)
```

```

##   df      AIC
## m4 18 -3168.681
## m5 24 -3186.836

```

```
summary(m5)
```

```

## 
## Call:
## lm(formula = log(price) ~ sqrt(mileage) + fuelType * aux_tax +
##     poly(tax, 3) + I(mpg^(-1/2)) + years_sell + transmission +
##     manufacturer + engineSize_num, data = df[!df$mout == "YesMOut",
##     ])
## 
## Residuals:
##   Min     1Q     Median     3Q     Max 
## -1.93233 -0.10450  0.00669  0.10866  0.57846 
## 
## Coefficients: (2 not defined because of singularities)
##              Estimate Std. Error t value
## (Intercept) 9.397e+00 1.075e-01 87.426
## sqrt(mileage) -9.599e-04 7.742e-05 -12.399
## fuelTypeef.Fuel-Hybrid -3.733e-02 4.972e-02 -0.751
## fuelTypeef.Fuel-Other -1.156e-01 8.541e-02 -1.353
## fuelTypeef.Fuel-Petrol -2.423e-01 2.357e-02 -10.282
## aux_taxf.tax-(125,145] -1.938e-01 8.221e-02 -2.358
## aux_taxf.tax-(145,580] -2.519e-01 8.809e-02 -2.859
## poly(tax, 3)1 2.977e+00 7.513e-01 3.962
## poly(tax, 3)2 -2.020e+00 1.066e+00 -1.896
## poly(tax, 3)3 4.233e-01 4.217e-01 1.004
## I(mpg^(-1/2)) 1.117e+01 2.322e-01 48.090
## years_sell -9.149e-02 2.869e-03 -31.889
## transmissionf.Trans-SemiAuto 1.725e-01 6.922e-03 24.923
## transmissionf.Trans-Automatic 1.453e-01 7.674e-03 18.939
## manufacturerBMW -4.417e-02 8.079e-03 -5.467
## manufacturerMercedes 2.001e-03 8.143e-03 0.246
## manufacturerVW -2.053e-01 7.282e-03 -28.187
## engineSize_nummedium_engine -1.876e-01 5.202e-02 -3.607
## engineSize_numsmall_engine -2.918e-01 5.222e-02 -5.587
## fuelTypeef.Fuel-Hybrid:aux_taxf.tax-(125,145] 4.948e-02 6.309e-02 0.784
## fuelTypeef.Fuel-Other:aux_taxf.tax-(125,145] 7.087e-02 1.100e-01 0.644
## fuelTypeef.Fuel-Petrol:aux_taxf.tax-(125,145] -7.115e-03 2.398e-02 -0.297
## fuelTypeef.Fuel-Hybrid:aux_taxf.tax-(145,580] NA NA NA
## fuelTypeef.Fuel-Other:aux_taxf.tax-(145,580] NA NA NA
## fuelTypeef.Fuel-Petrol:aux_taxf.tax-(145,580] 4.115e-02 2.503e-02 1.644
## 
##              Pr(>|t|) 
## (Intercept) < 2e-16 ***
## sqrt(mileage) < 2e-16 ***
## fuelTypeef.Fuel-Hybrid 0.452790

```

```

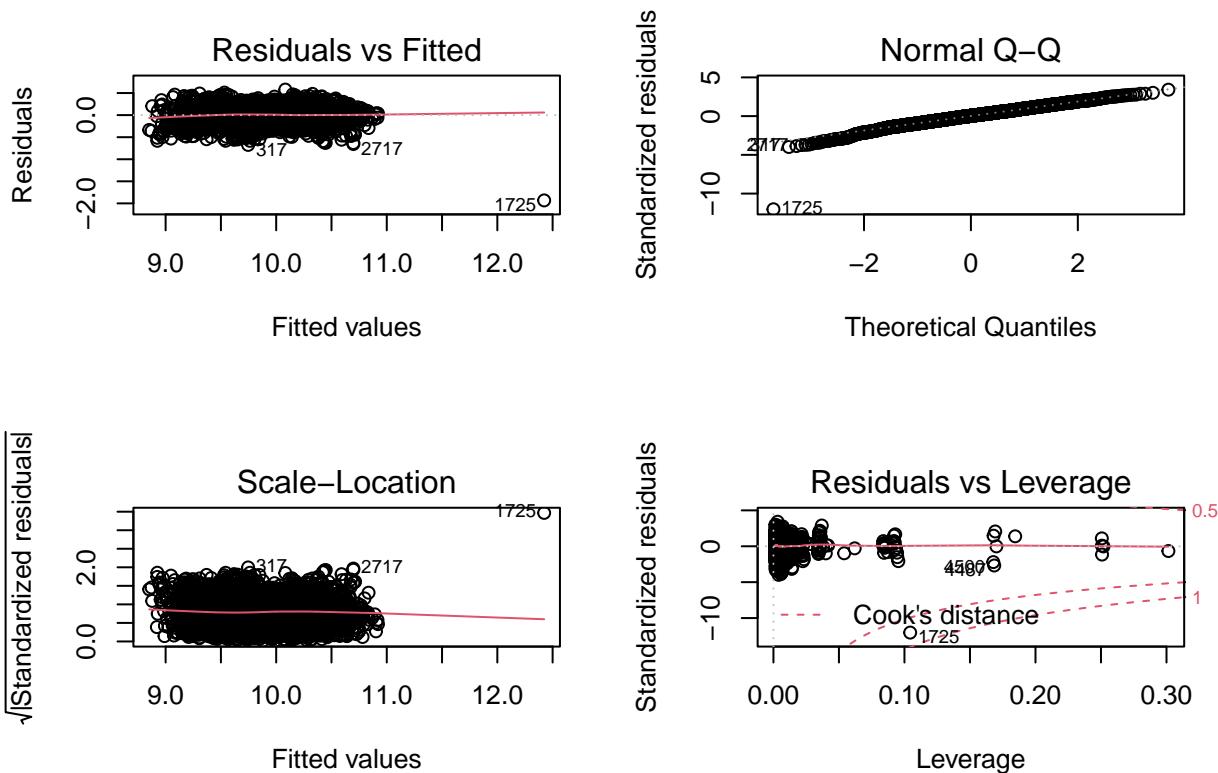
## fuelTypef.Fuel-Other          0.176035
## fuelTypef.Fuel-Petrol        < 2e-16 ***
## aux_taxf.tax-(125,145]       0.018422 *
## aux_taxf.tax-(145,580]       0.004263 **
## poly(tax, 3)1                 7.54e-05 ***
## poly(tax, 3)2                 0.058053 .
## poly(tax, 3)3                 0.315537
## I(mpg^(-1/2))                < 2e-16 ***
## years_sell                     < 2e-16 ***
## transmissionf.Trans-SemiAuto < 2e-16 ***
## transmissionf.Trans-Automatic < 2e-16 ***
## manufacturerBMW              4.81e-08 ***
## manufacturerMercedes          0.805905
## manufacturerVW                < 2e-16 ***
## engineSize_nummedium_engine   0.000313 ***
## engineSize_numsmall_engine    2.44e-08 ***
## fuelTypef.Fuel-Hybrid:aux_taxf.tax-(125,145] 0.432954
## fuelTypef.Fuel-Other:aux_taxf.tax-(125,145]  0.519592
## fuelTypef.Fuel-Petrol:aux_taxf.tax-(125,145]  0.766718
## fuelTypef.Fuel-Hybrid:aux_taxf.tax-(145,580]      NA
## fuelTypef.Fuel-Other:aux_taxf.tax-(145,580]      NA
## fuelTypef.Fuel-Petrol:aux_taxf.tax-(145,580]  0.100187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1698 on 4513 degrees of freedom
## Multiple R-squared:  0.8397, Adjusted R-squared:  0.8389
## F-statistic: 1075 on 22 and 4513 DF,  p-value: < 2.2e-16

```

```

par(mfrow=c(2,2))
plot(m5)

```

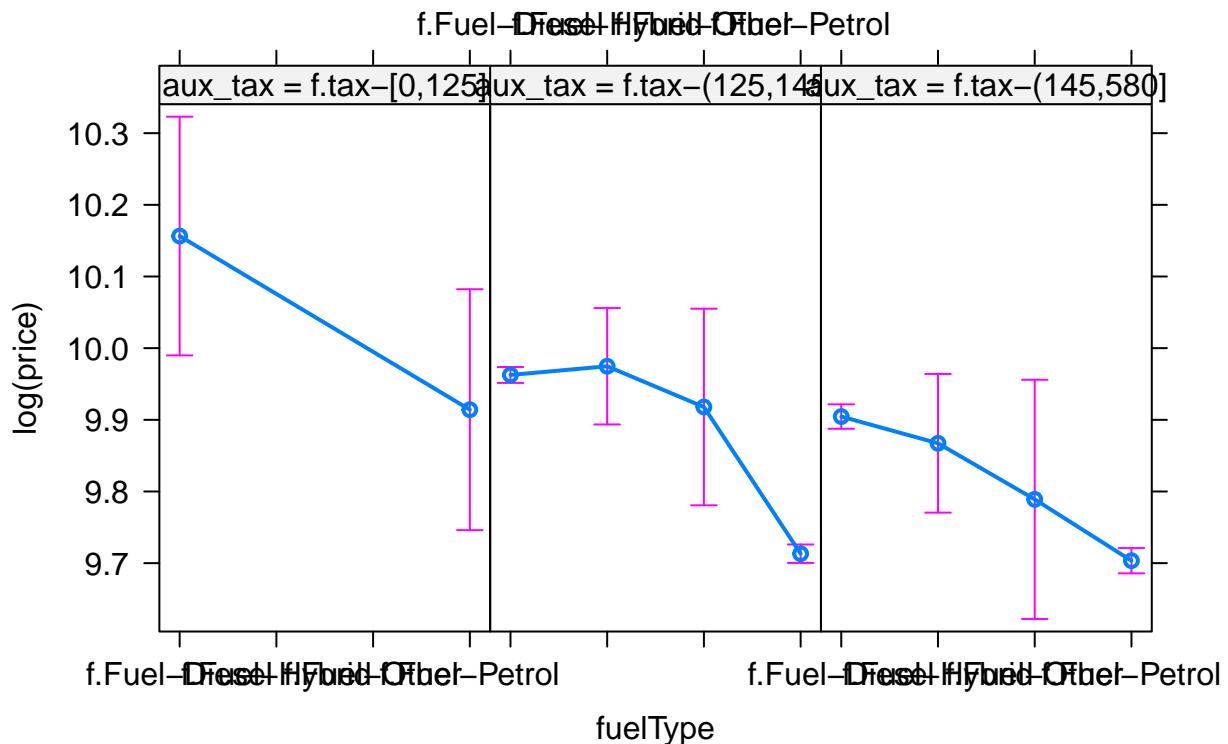


```

plot(allEffects(m5), selection = 8)

```

### fuelType\*aux\_tax effect plot



As we can see price variable is related to fuelType and aux\_tax variables, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

These explanatory variables explain a 83.97 % of the target's variability. Also, we can see that residuals are still homoscedastic and normal. However, we can see some influential observation.

We can see that neither interaction is significant, because the p-values are greater than 0.05.

#### 2.3.2 Factor and covariate interaction

We are going to see if price variable (Y response) is related to fuelType (factor A) and mpg (numeric) variables.

```
m6<-lm(log(price)~sqrt(mileage)+fuelType*aux_tax+fuelType*I(mpg^(-1/2))+poly(tax,3)+years_sell+
  transmission+manufacturer+engineSize_num,data=df[!df$mout=="YesMOut",])
anova(m5,m6)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ sqrt(mileage) + fuelType * aux_tax + poly(tax, 3) +
##   I(mpg^(-1/2)) + years_sell + transmission + manufacturer +
##   engineSize_num
## Model 2: log(price) ~ sqrt(mileage) + fuelType * aux_tax + fuelType *
##   I(mpg^(-1/2)) + poly(tax, 3) + years_sell + transmission +
##   manufacturer + engineSize_num
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1   4513 130.16
## 2   4510 125.67  3     4.4935 53.754 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m6)
```

```
##
## Call:
## lm(formula = log(price) ~ sqrt(mileage) + fuelType * aux_tax +
```

```

##      fuelType * I(mpg^(-1/2)) + poly(tax, 3) + years_sell + transmission +
##      manufacturer + engineSize_num, data = df[!df$mout == "YesMOut",
##      ])
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -0.69746 -0.10487  0.00618  0.10768  0.59387
##
## Coefficients: (2 not defined because of singularities)
##                                         Estimate Std. Error t value
## (Intercept)                      9.317e+00  1.124e-01 82.873
## sqrt(mileage)                  -9.737e-04  7.659e-05 -12.713
## fuelTypef.Fuel-Hybrid           1.244e+00  1.294e-01  9.615
## fuelTypef.Fuel-Other             4.685e-01  3.895e-01  1.203
## fuelTypef.Fuel-Petrol            -4.606e-01  5.547e-02 -8.304
## aux_taxf.tax-(125,145]          -1.498e-01  8.180e-02 -1.831
## aux_taxf.tax-(145,580]          -2.030e-01  8.766e-02 -2.316
## I(mpg^(-1/2))                  1.113e+01  3.111e-01 35.797
## poly(tax, 3)1                  2.461e+00  7.498e-01  3.282
## poly(tax, 3)2                  -1.477e+00  1.061e+00 -1.392
## poly(tax, 3)3                  3.197e-01  4.161e-01  0.768
## years_sell                      -8.985e-02  2.823e-03 -31.826
## transmissionf.Trans-SemiAuto   1.675e-01  6.816e-03 24.580
## transmissionf.Trans-Automatic  1.386e-01  7.566e-03 18.312
## manufacturerBMW                -4.435e-02  8.020e-03 -5.530
## manufacturerMercedes            2.832e-03  8.117e-03  0.349
## manufacturerVW                 -2.006e-01  7.200e-03 -27.855
## engineSize_nummedium_engine     -1.525e-01  5.139e-02 -2.968
## engineSize_numsmall_engine       -2.518e-01  5.152e-02 -4.887
## fuelTypef.Fuel-Hybrid:aux_taxf.tax-(125,145]  1.107e-01  6.221e-02  1.780
## fuelTypef.Fuel-Other:aux_taxf.tax-(125,145]  1.253e-01  1.136e-01  1.103
## fuelTypef.Fuel-Petrol:aux_taxf.tax-(125,145] -1.765e-02  2.376e-02 -0.743
## fuelTypef.Fuel-Hybrid:aux_taxf.tax-(145,580]      NA        NA        NA
## fuelTypef.Fuel-Other:aux_taxf.tax-(145,580]      NA        NA        NA
## fuelTypef.Fuel-Petrol:aux_taxf.tax-(145,580]  3.599e-02  2.471e-02  1.456
## fuelTypef.Fuel-Hybrid:I(mpg^(-1/2))            -9.446e+00  8.875e-01 -10.644
## fuelTypef.Fuel-Other:I(mpg^(-1/2))            -4.253e+00  2.748e+00 -1.548
## fuelTypef.Fuel-Petrol:I(mpg^(-1/2))            1.532e+00  3.748e-01  4.088
## Pr(>|t|)
## (Intercept)                      < 2e-16 ***
## sqrt(mileage)                  < 2e-16 ***
## fuelTypef.Fuel-Hybrid           < 2e-16 ***
## fuelTypef.Fuel-Other             0.22902
## fuelTypef.Fuel-Petrol            < 2e-16 ***
## aux_taxf.tax-(125,145]          0.06712 .
## aux_taxf.tax-(145,580]          0.02060 *
## I(mpg^(-1/2))                  < 2e-16 ***
## poly(tax, 3)1                  0.00104 **
## poly(tax, 3)2                  0.16388
## poly(tax, 3)3                  0.44235
## years_sell                      < 2e-16 ***
## transmissionf.Trans-SemiAuto   < 2e-16 ***
## transmissionf.Trans-Automatic  < 2e-16 ***
## manufacturerBMW                3.38e-08 ***
## manufacturerMercedes            0.72723
## manufacturerVW                 < 2e-16 ***
## engineSize_nummedium_engine     0.00301 **
## engineSize_numsmall_engine      1.06e-06 ***
## fuelTypef.Fuel-Hybrid:aux_taxf.tax-(125,145]  0.07521 .
## fuelTypef.Fuel-Other:aux_taxf.tax-(125,145]  0.27018
## fuelTypef.Fuel-Petrol:aux_taxf.tax-(125,145]  0.45762
## fuelTypef.Fuel-Hybrid:aux_taxf.tax-(145,580]      NA
## fuelTypef.Fuel-Other:aux_taxf.tax-(145,580]      NA
## fuelTypef.Fuel-Petrol:aux_taxf.tax-(145,580]  0.14534

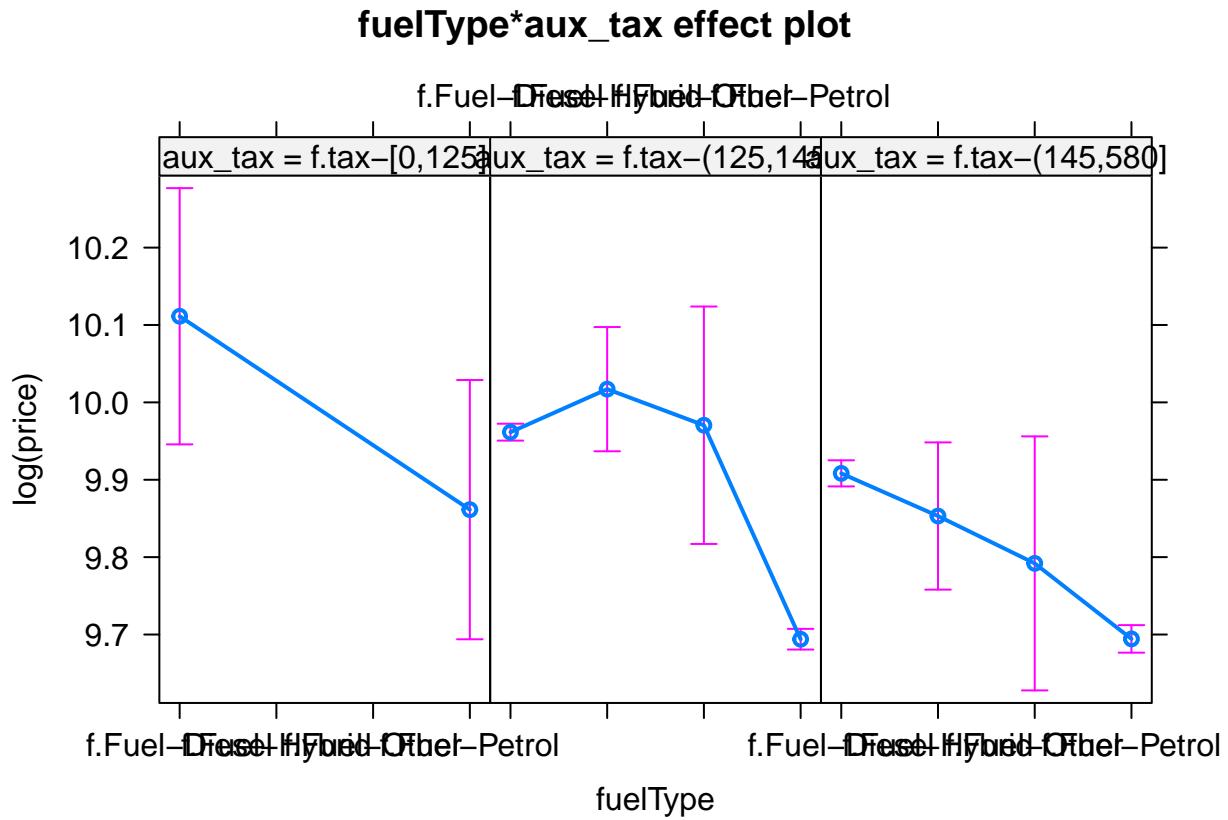
```

```

## fuelTypeef.Fuel-Hybrid:I(mpg^(-1/2)) < 2e-16 ***
## fuelTypeef.Fuel-Other:I(mpg^(-1/2)) 0.12179
## fuelTypeef.Fuel-Petrol:I(mpg^(-1/2)) 4.43e-05 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1669 on 4510 degrees of freedom
## Multiple R-squared: 0.8452, Adjusted R-squared: 0.8444
## F-statistic: 985.3 on 25 and 4510 DF, p-value: < 2.2e-16

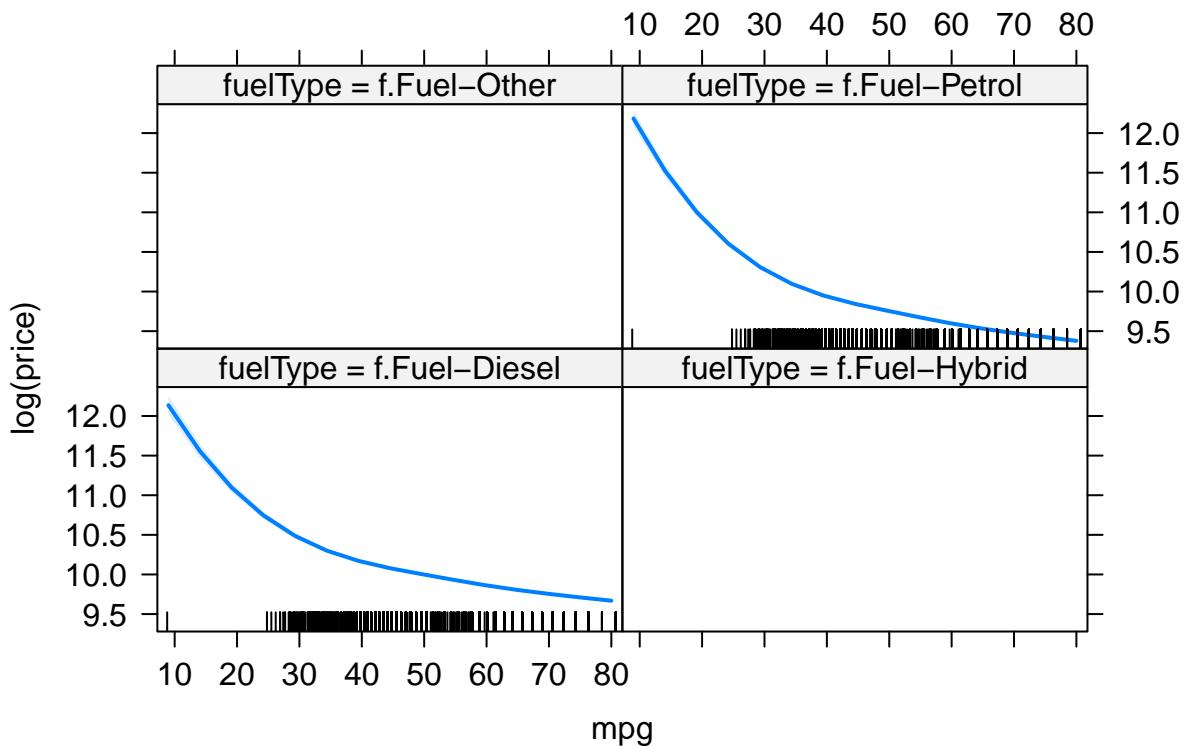
```

```
plot(allEffects(m6), selection = 7)
```



```
plot(allEffects(m6), selection = 8)
```

## fuelType\*mpg effect plot



As we can see price variable is related to fuelType and mpg variables, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

These explanatory variables explain a 84.52 % of the target's variability.

We can see that the only significant interactions are:

That within/in the hybrid category, the mpg variable causes the logarithm of the price decrease by -9.45 units, because the p-value is less than 0.05.

That within/in the petrol category, the mpg variable causes the logarithm of the price increase by 1.53 units, because the p-value is less than 0.05.

## 2.4 Best model selection

```

m7 <- step( m6, k=log(nrow(df[!df$mout=="YesMOut",])))
## Start:  AIC=-16047.89
## log(price) ~ sqrt(mileage) + fuelType * aux_tax + fuelType *
##   I(mpg^(-1/2)) + poly(tax, 3) + years_sell + transmission +
##   manufacturer + engineSize_num
##
##                                Df  Sum of Sq    RSS    AIC
## - fuelType:aux_tax        4    0.7897 126.46 -16053
## <none>                      125.67 -16048
## - poly(tax, 3)            3    1.1608 126.83 -16031
## - fuelType:I(mpg^(-1/2))  3    4.4935 130.16 -15914
## - engineSize_num          2    4.6877 130.36 -15899
## - sqrt(mileage)           1    4.5033 130.17 -15897
## - transmission            2   17.3640 143.03 -15478
## - manufacturer            3   27.4917 153.16 -15176
## - years_sell               1   28.2231 153.89 -15137
##
## Step:  AIC=-16053.16
## log(price) ~ sqrt(mileage) + fuelType + aux_tax + I(mpg^(-1/2)) +
##   poly(tax, 3) + years_sell + transmission + manufacturer +

```

```

##      engineSize_num + fuelType:I(mpg^(-1/2))
##
##                               Df  Sum of Sq    RSS    AIC
## - aux_tax                      2    0.1544 126.61 -16064
## <none>                         126.46 -16053
## - poly(tax, 3)                  3    1.1615 127.62 -16037
## - fuelType:I(mpg^(-1/2))       3    4.2646 130.72 -15928
## - engineSize_num                2    4.6556 131.11 -15906
## - sqrt(mileage)                1    4.6062 131.06 -15899
## - transmission                 2    17.5858 144.04 -15479
## - manufacturer                 3    27.8595 154.32 -15175
## - years_sell                   1    28.6802 155.14 -15134
##
## Step:  AIC=-16064.46
## log(price) ~ sqrt(mileage) + fuelType + I(mpg^(-1/2)) + poly(tax,
##      3) + years_sell + transmission + manufacturer + engineSize_num +
##      fuelType:I(mpg^(-1/2))
##
##                               Df  Sum of Sq    RSS    AIC
## <none>                         126.61 -16064
## - poly(tax, 3)                  3    1.0239 127.64 -16053
## - fuelType:I(mpg^(-1/2))       3    4.4176 131.03 -15934
## - engineSize_num                2    4.5798 131.19 -15920
## - sqrt(mileage)                1    4.8772 131.49 -15901
## - transmission                 2    17.8100 144.42 -15484
## - manufacturer                 3    28.2647 154.88 -15176
## - years_sell                   1    30.0499 156.66 -15107

```

```
summary(m7)
```

```

##
## Call:
## lm(formula = log(price) ~ sqrt(mileage) + fuelType + I(mpg^(-1/2)) +
##      poly(tax, 3) + years_sell + transmission + manufacturer +
##      engineSize_num + fuelType:I(mpg^(-1/2)), data = df[!df$mout ==
##      "YesMOut", ])
##
## Residuals:
##      Min      1Q      Median      3Q      Max
## -0.68681 -0.10461  0.00487  0.10837  0.58394
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                9.160e+00  7.135e-02 128.377 < 2e-16 ***
## sqrt(mileage)              -9.964e-04  7.555e-05 -13.189 < 2e-16 ***
## fuelTypef.Fuel-Hybrid      1.294e+00  1.258e-01 10.288 < 2e-16 ***
## fuelTypef.Fuel-Other       3.432e-01  3.866e-01  0.888 0.374682  
## fuelTypef.Fuel-Petrol      -4.382e-01  5.261e-02 -8.330 < 2e-16 ***
## I(mpg^(-1/2))              1.124e+01  3.025e-01 37.163 < 2e-16 ***
## poly(tax, 3)1              8.162e-01  1.774e-01  4.602 4.30e-06 ***
## poly(tax, 3)2              7.318e-01  1.766e-01  4.143 3.48e-05 ***
## poly(tax, 3)3              -1.146e-01 1.858e-01 -0.617 0.537340  
## years_sell                 -9.133e-02 2.790e-03 -32.739 < 2e-16 ***
## transmissionf.Trans-SemiAuto 1.688e-01  6.805e-03 24.806 < 2e-16 ***
## transmissionf.Trans-Automatic 1.395e-01  7.556e-03 18.470 < 2e-16 ***
## manufacturerBMW            -4.450e-02 8.030e-03 -5.542 3.16e-08 ***
## manufacturerMercedes       4.114e-03  8.105e-03  0.508 0.611758  
## manufacturerVW              -2.013e-01 7.206e-03 -27.941 < 2e-16 ***
## engineSize_nummedium_engine -1.587e-01  5.153e-02 -3.081 0.002079 ** 
## engineSize_numsmall_engine  -2.557e-01  5.167e-02 -4.948 7.75e-07 ***
## fuelTypef.Fuel-Hybrid:I(mpg^(-1/2)) -9.522e+00 8.779e-01 -10.847 < 2e-16 ***
## fuelTypef.Fuel-Other:I(mpg^(-1/2))  -2.959e+00 2.624e+00 -1.128 0.259583  
## fuelTypef.Fuel-Petrol:I(mpg^(-1/2)) 1.360e+00  3.693e-01  3.684 0.000233 ***

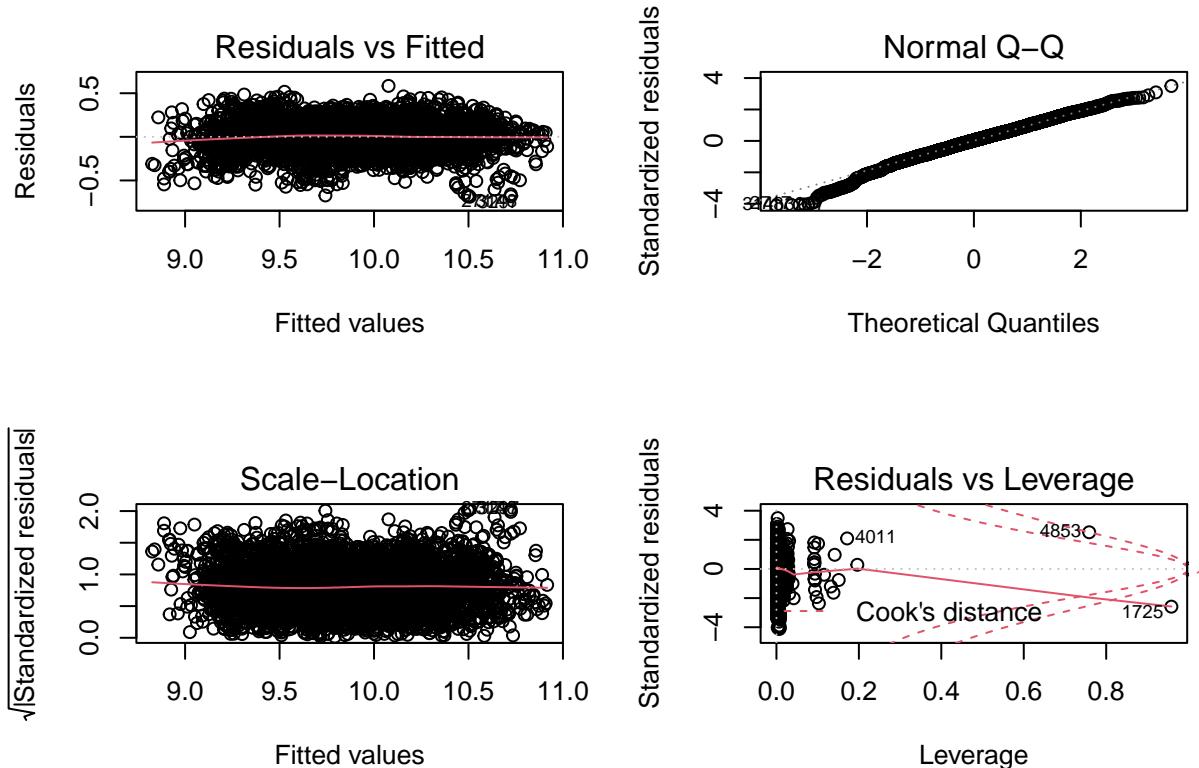
```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1674 on 4516 degrees of freedom
## Multiple R-squared: 0.8441, Adjusted R-squared: 0.8434
## F-statistic: 1287 on 19 and 4516 DF, p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(m7)

```



Our best model (model that explains the maximum possible variability of the target with/using the least number of possible variables) explains a 84.41 % of the target's variability.

Also, we can see that residuals are still homoscedastic and normal. However, we can see some influential observation.

## 2.5 Diagnostics

A good model should be consistent with theoretical properties in residual analysis. Neither influential nor unusual data should be included.

```

dfwork <- df[!df$mout=="YesMOut",]
# Define initial parameters:
p <- length(m7$coefficients)
n <- length(m7$fitted.values)
h_param <- 3

# Residual analysis:
llres <- which(abs(rstudent(m7))>3); llres

## 317 846 1329 2084 2113 2188 2225 2231 2343 2414 2548 2576 2614 2717 2875 2885
## 286 766 1184 1827 1852 1921 1954 1960 2064 2134 2262 2290 2326 2428 2580 2589
## 3148 3170 3297 3492 4194 4694 4696 4709 4711 4734 4745 4749 4767 4775
## 2835 2854 2958 3119 3773 4254 4256 4268 4270 4291 4302 4306 4324 4332

```

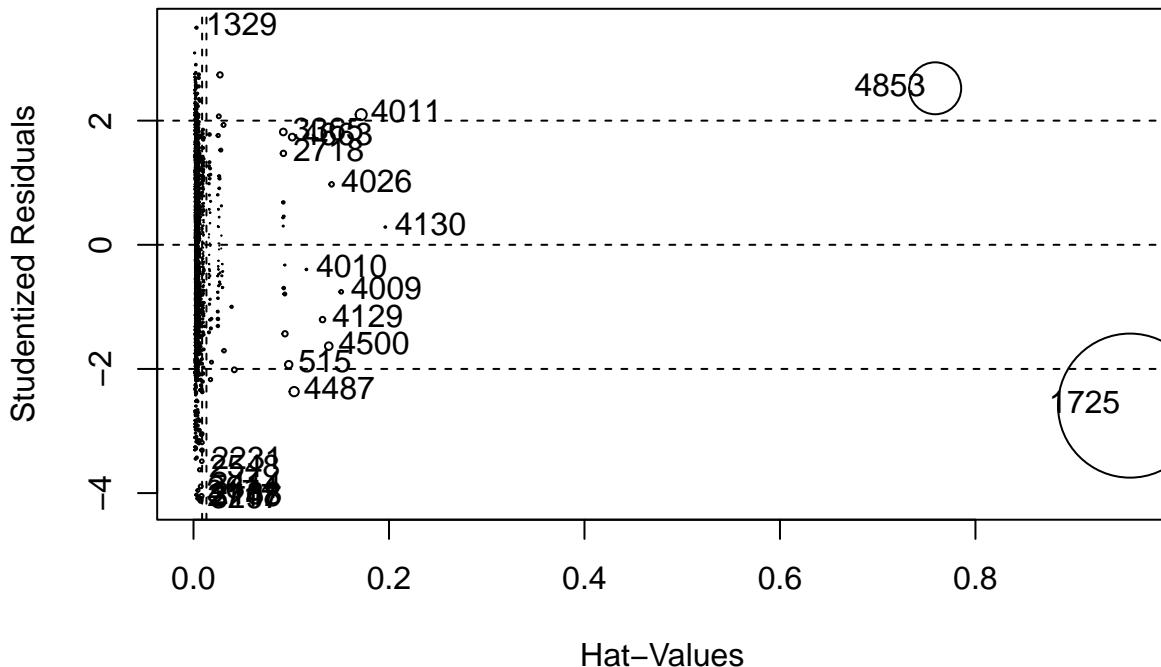
```

length(llres)

## [1] 30

par(mfrow=c(1,1))
influencePlot(m7, id=list(n=10))

```

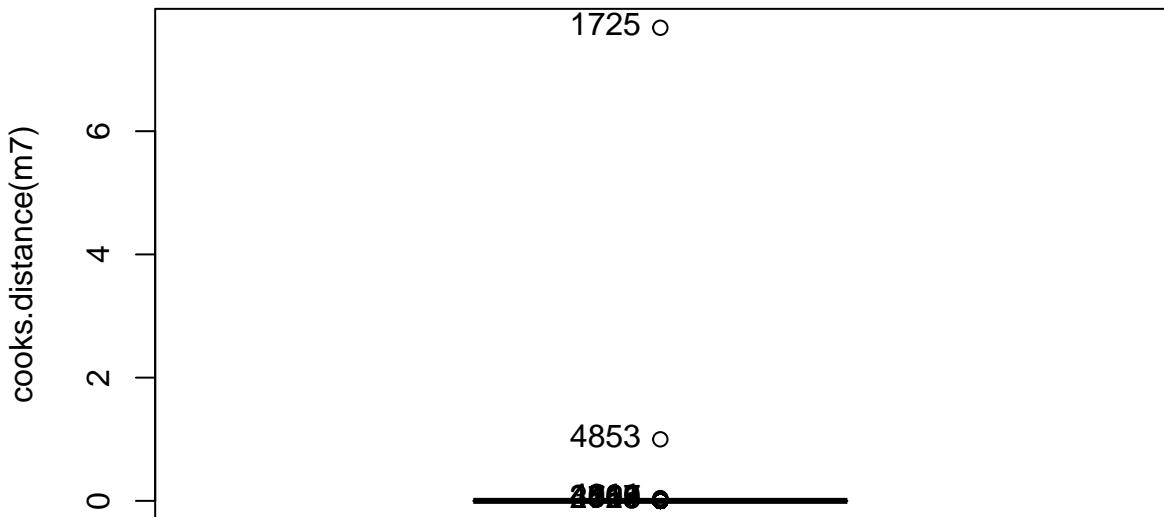


```

##           StudRes      Hat      CookD
## 317    -4.0279974 0.003151680 0.0025562288
## 515    -1.9301950 0.097427441 0.0200959943
## 1329   3.4971738 0.003090173 0.0018908311
## 1725   -2.5907973 0.958173913 7.6786521919
## 2188   -4.0480721 0.008143926 0.0067046241
## 2231   -3.4864599 0.008505395 0.0052008225
## 2414   -3.8912232 0.007416609 0.0056392688
## 2548   -3.6241929 0.006271932 0.0041339101
## 2614   -3.9558674 0.004565950 0.0035773850
## 2717   -4.0853080 0.004565950 0.0038144514
## 2718   1.4724636 0.091970385 0.0109772820
## 3148   -4.0921725 0.007261760 0.0061034220
## 3297   -4.1240130 0.007225702 0.0061674111
## 3365   1.8175147 0.091874195 0.0167013642
## 4009   -0.7574870 0.151120453 0.0051078542
## 4010   -0.3971596 0.115551784 0.0010305888
## 4011   2.0972403 0.171532745 0.0455000011
## 4026   0.9736037 0.141303389 0.0077992398
## 4129   -1.2053057 0.132023852 0.0110475370
## 4130   0.2859693 0.196278161 0.0009987649
## 4487   -2.3653409 0.102962077 0.0320761559
## 4500   -1.6327958 0.138398975 0.0214042610
## 4663   1.7349146 0.100989455 0.0168983462
## 4853   2.5210982 0.758815641 0.9986697310

```

```
Boxplot(cooks.distance(m7), id=list(labels=row.names(dfwork)))
```



```
## [1] "1725" "4853" "4011" "4487" "4500" "515" "4663" "3365" "4129" "2718"
```

*# A priori influential observation*

```
ll_priori_influential <- which(abs(hatvalues(m7))>h_param*(p/n))
length(ll_priori_influential)
```

```
## [1] 97
```

*# A posteriori influential observation:*

```
ll_posteriori_influential <- which(abs(cooks.distance(m7))>(4/(n-p)));
length(ll_posteriori_influential)
```

```
## [1] 161
```

```
ll_unique_influential <- unique(c(ll_priori_influential,ll_posteriori_influential));
length(ll_unique_influential)
```

```
## [1] 209
```

```
m7 <- update(m7,data=dfwork[-ll_posteriori_influential,])
summary(m7)
```

```
##  
## Call:  
## lm(formula = log(price) ~ sqrt(mileage) + fuelType + I(mpg^(-1/2)) +  
## poly(tax, 3) + years_sell + transmission + manufacturer +  
## engineSize_num + fuelType:I(mpg^(-1/2)), data = dfwork[-ll_posteriori_influential,  
## ])  
##
```

```

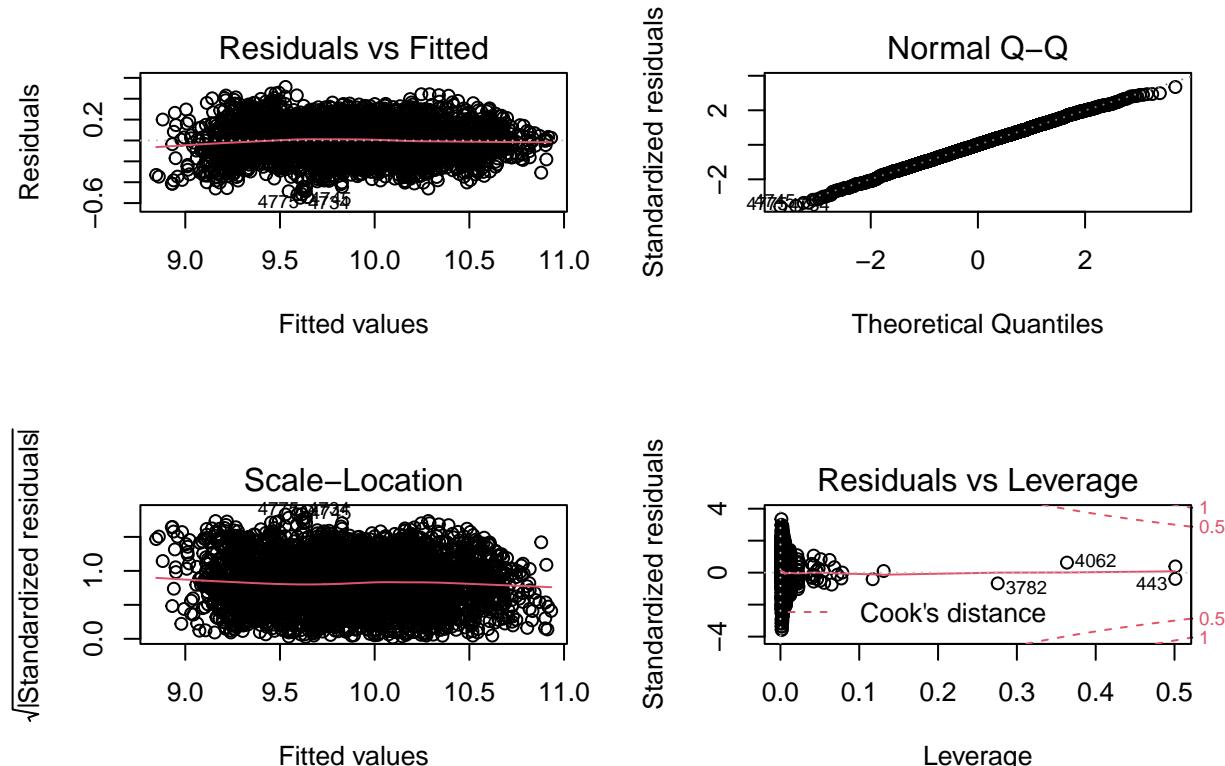
## Residuals:
##      Min      1Q  Median      3Q      Max
## -0.54893 -0.10249  0.00149  0.10290  0.51323
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 8.988e+00  1.184e-01  75.908 < 2e-16 ***
## sqrt(mileage)              -1.047e-03 7.114e-05 -14.712 < 2e-16 ***
## fuelTypef.Fuel-Hybrid      7.144e-01  8.047e-01   0.888 0.374706    
## fuelTypef.Fuel-Petrol      -4.424e-01 4.959e-02  -8.922 < 2e-16 ***
## I(mpg^(-1/2))              1.211e+01  2.859e-01  42.359 < 2e-16 ***
## poly(tax, 3)1              6.755e-01  1.602e-01   4.218 2.52e-05 ***
## poly(tax, 3)2              4.669e-01  1.620e-01   2.882 0.003966 **  
## poly(tax, 3)3              -3.097e-02 1.701e-01  -0.182 0.855516    
## years_sell                  -8.504e-02 2.644e-03 -32.163 < 2e-16 ***
## transmissionf.Trans-SemiAuto 1.548e-01  6.371e-03  24.304 < 2e-16 ***
## transmissionf.Trans-Automatic 1.331e-01  7.064e-03  18.846 < 2e-16 ***
## manufacturerBMW            -3.560e-02 7.467e-03  -4.769 1.92e-06 ***
## manufacturerMercedes        2.616e-02 7.631e-03   3.428 0.000614 ***  
## manufacturerVW              -1.969e-01 6.697e-03 -29.397 < 2e-16 ***
## engineSize_nummedium_engine -1.357e-01 1.090e-01  -1.245 0.213376    
## engineSize_numsmall_engine  -2.184e-01 1.090e-01  -2.003 0.045212 *  
## fuelTypef.Fuel-Hybrid:I(mpg^(-1/2)) -5.524e+00 5.881e+00  -0.939 0.347644    
## fuelTypef.Fuel-Petrol:I(mpg^(-1/2))  1.336e+00 3.485e-01   3.834 0.000128 ***  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1534 on 4357 degrees of freedom
## Multiple R-squared:  0.8643, Adjusted R-squared:  0.8637
## F-statistic:  1632 on 17 and 4357 DF,  p-value: < 2.2e-16

```

```

par(mfrow=c(2,2))
plot(m7)

```



We have 30 outliers on the regression. We have 97 a priori influential observations. We have 161 a posteriori influential observations. As we can see in the Boxplot the 1725 and 4853 observations are the most significant

ones. Between the a priori influential observations and the a posteriori influential observations we have 209 unique observations. That means that we have 49 observations that were influential a priori and then became influential a posteriori

To achieve the best model we remove the a posteriori influential observations from the model.

These explanatory variables explain a 86.43 % of the target's variability. Also, we can see that residuals are still homoscedastic and normal.

### 3 Binary/Logistic Regression Models

#### 3.1 Dividing/Splitting the sample

```
set.seed(1234)
llwork <- sample(1:nrow(df), round(0.70*nrow(df), 0))

df_train <- df[llwork,]
df_test <- df[-llwork,]
```

#### 3.2 Only numeric variables

```
m0<-glm(Audi~1,family="binomial",data=df_train[!df_train$mout=="YesMOut",])
m1<-glm(Audi~mileage+tax+mpg+years_sell,family="binomial",data=df_train[!df_train$mout=="YesMOut",])
anova( m0, m1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Audi ~ 1
## Model 2: Audi ~ mileage + tax + mpg + years_sell
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3169      3288.2
## 2      3165      3242.1  4    46.112 2.334e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

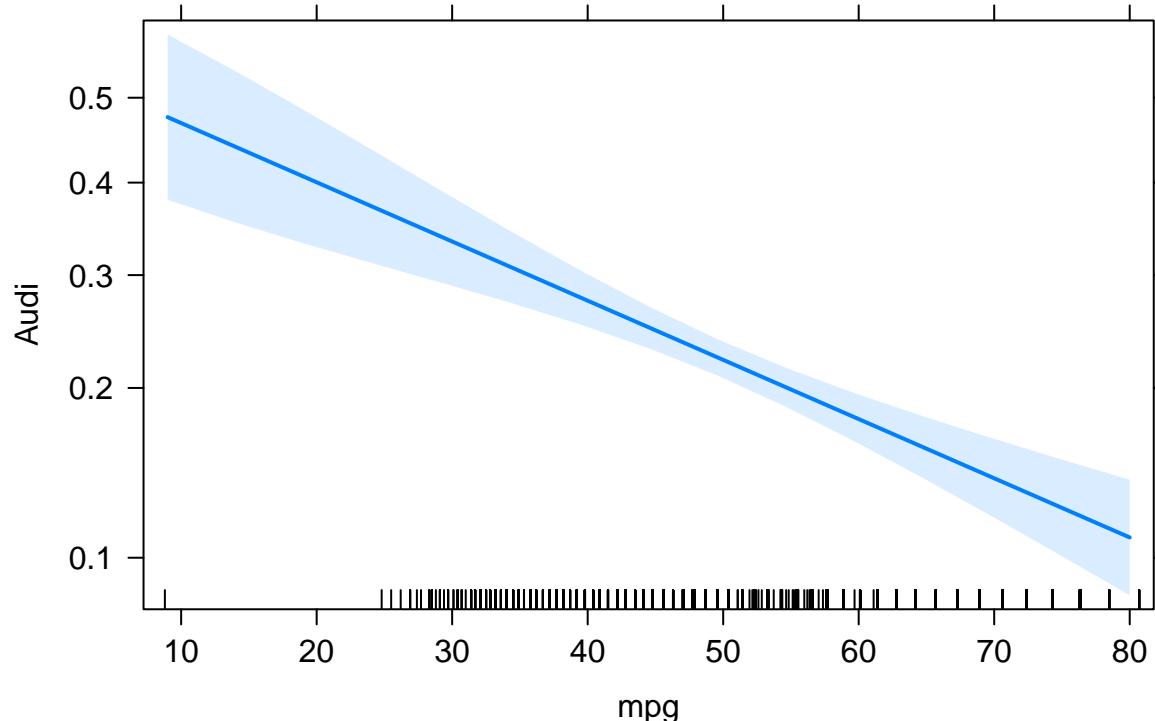
summary(m1)
```

```
##
## Call:
## glm(formula = Audi ~ mileage + tax + mpg + years_sell, family = "binomial",
##      data = df_train[!df_train$mout == "YesMOut", ])
##
## Deviance Residuals:
##      Min        1Q        Median        3Q        Max
## -1.0640  -0.7150  -0.6463  -0.5441   2.0986
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.903e-01  6.453e-01  -1.070   0.285
## mileage      2.032e-06  4.982e-06   0.408   0.683
## tax          3.227e-03  3.830e-03   0.843   0.399
## mpg          -2.828e-02  4.655e-03  -6.075 1.24e-09 ***
## years_sell   7.546e-02  4.686e-02   1.610   0.107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##  
## Null deviance: 3288.2 on 3169 degrees of freedom  
## Residual deviance: 3242.1 on 3165 degrees of freedom  
## AIC: 3252.1  
##  
## Number of Fisher Scoring iterations: 4
```

```
plot(allEffects(m1), selection = 3)
```

mpg effect plot

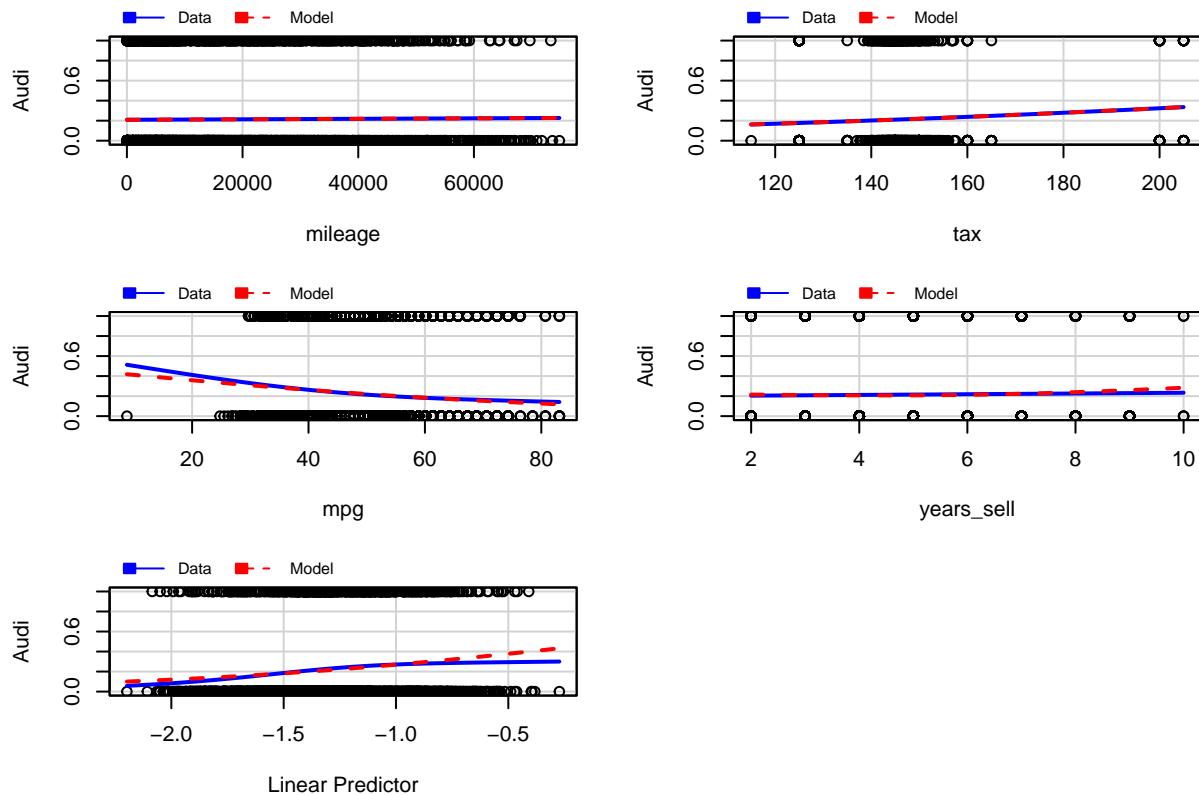


```
vif(m1)
```

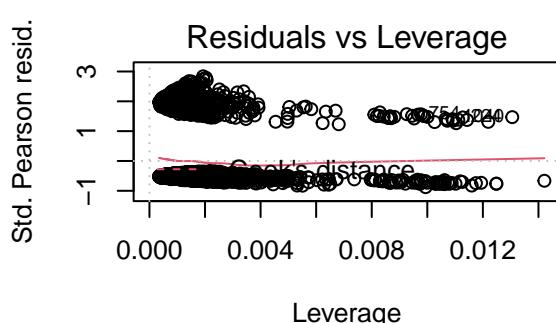
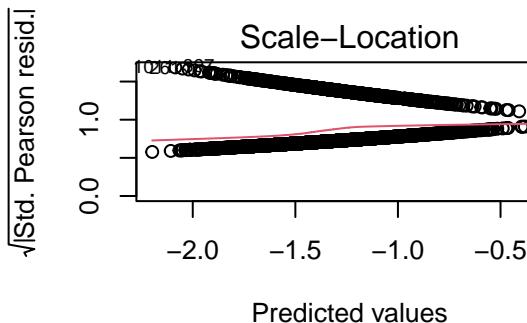
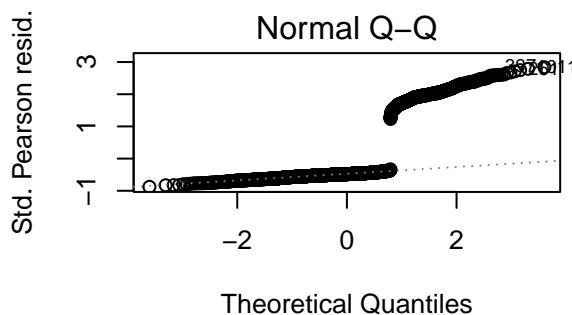
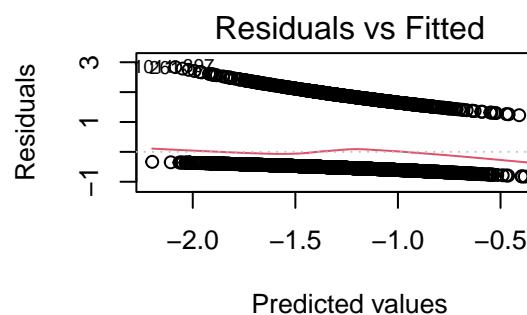
```
## mileage          tax          mpg years_sell  
## 3.514555  1.131628  1.347377  3.611162
```

```
marginalModelPlots(m1)
```

## Marginal Model Plots



```
par(mfrow=c(2,2))
plot(m1)
```



As we can see the decrease in deviance is significant, because the p-value of anova test is less than 0.05.

As we can see the only useful variable is the mpg variable, because its p-value is less than 0.05. We should remove the other variables but we won't do it because applying transformations to them or adding the factors they may become significant as well.

We can say that against more mpg less probability of being Audi. More specifically, increasing by 1 unit mpg then  $\exp(-0.0281) = 0.97 \rightarrow 100(1-0.97) = 3\%$ , the probability of being Audi decreases by 3 %.

Regarding correlations between variables we don't have to worry (because there aren't  $\geq 4$ ). Also, we can see that the model captures the data well.

Also, we can see that the model captures the data well and patterns in the residuals.

### 3.2.1 Transformations

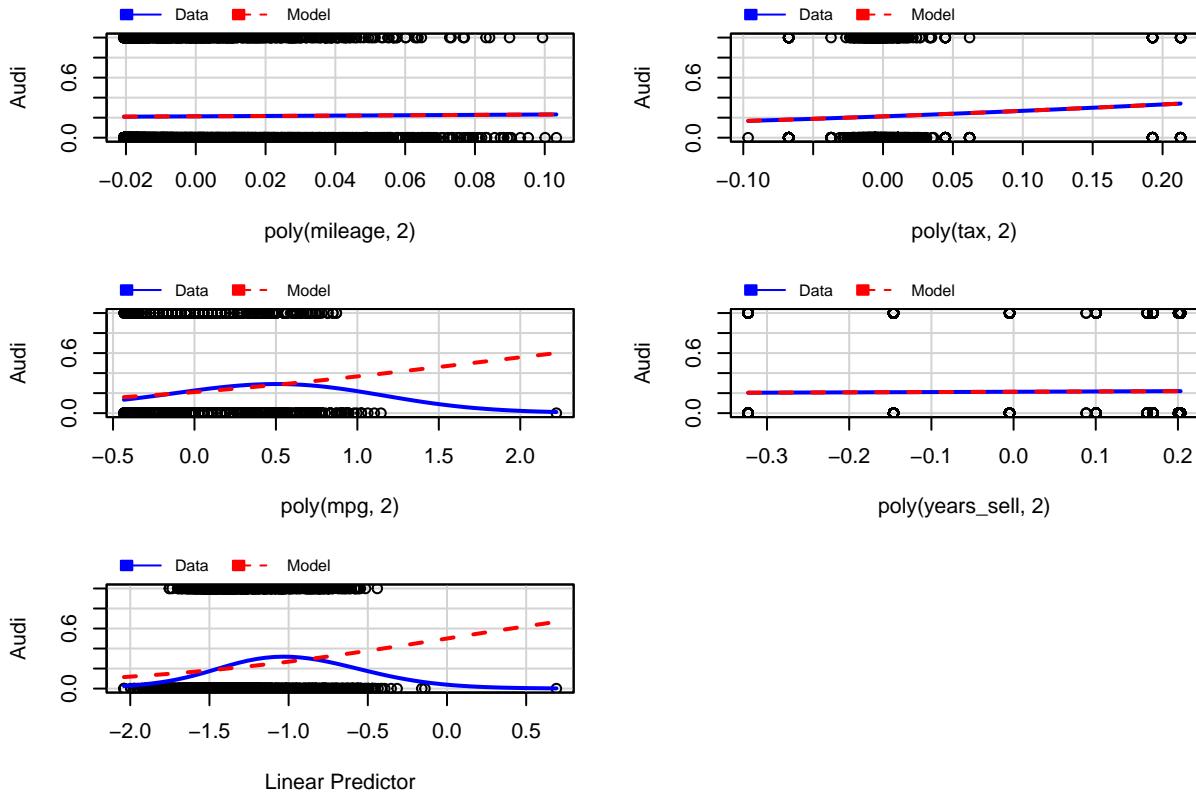
```
m2 <- glm(Audi ~ poly(mileage, 2) + poly(tax, 2) + poly(mpg, 2) + poly(years_sell, 2), family = "binomial",
           data = df_train[!df_train$mout == "YesMOut",])
summary(m2)
```

```
## 
## Call:
## glm(formula = Audi ~ poly(mileage, 2) + poly(tax, 2) + poly(mpg,
##   2) + poly(years_sell, 2), family = "binomial", data = df_train[!df_train$mout ==
##   "YesMOut", ])
## 
## Deviance Residuals:
##   Min     1Q   Median     3Q    Max 
## -1.4811 -0.7004 -0.6402 -0.5605  1.9561 
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.32880   0.04426 -30.025 < 2e-16 ***
## poly(mileage, 2)1  1.22708   4.75213   0.258   0.7962    
## poly(mileage, 2)2  0.22819   3.21601   0.071   0.9434    
## poly(tax, 2)1     2.14704   2.37485   0.904   0.3660    
## poly(tax, 2)2     0.12713   2.47990   0.051   0.9591    
## poly(mpg, 2)1     -17.57350  2.93269  -5.992  2.07e-09 ***
## poly(mpg, 2)2      4.54899   2.44141   1.863   0.0624 .  
## poly(years_sell, 2)1  8.43532   4.92964   1.711   0.0871 .  
## poly(years_sell, 2)2 -3.33701   3.16421  -1.055   0.2916    
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
## Null deviance: 3288.2 on 3169 degrees of freedom
## Residual deviance: 3237.0 on 3161 degrees of freedom
## AIC: 3255
## 
## Number of Fisher Scoring iterations: 4
```

```
marginalModelPlots(m2)
```

```
## Warning in mmpls(...): Splines and/or polynomials replaced by a fitted linear
## combination
```

## Marginal Model Plots



Trying the quadratic terms of the variables we can see that the only useful variable is mpg. Also, in the marginalModelPlots we can see that this model captures the data worse (bottom left plot).

We have tried adding the factors to this formula and the years\_sell variable has become significant, but we do not consider it to be a very useful variable to predict whether a car is Audi or not, so we have decided not to apply transformations on the regressors.

### 3.3 Including factors

```
m2 <- update(m1, ~.+fuelType+transmission+engineSize_num, data=df_train[!df_train$mout=="YesMOut",])
m2pet <- update(m1, ~.+transmission+engineSize_num, data=df_train[!df_train$mout=="YesMOut",])
anova( m2pet, m2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Audi ~ mileage + tax + mpg + years_sell + transmission + engineSize_num
## Model 2: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##           engineSize_num
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3161      3161.5
## 2      3158      3140.7  3      20.85 0.0001131 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see fuelType variable is useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

```
m2 <- update(m1, ~.+fuelType+transmission+engineSize_num, data=df_train[!df_train$mout=="YesMOut",])
m2pet <- update(m1, ~.+fuelType+engineSize_num, data=df_train[!df_train$mout=="YesMOut",])
anova( m2pet, m2, test="Chisq")
```

```
## Analysis of Deviance Table
##
```

```

## Model 1: Audi ~ mileage + tax + mpg + years_sell + fuelType + engineSize_num
## Model 2: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##   engineSize_num
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3160      3148.5
## 2      3158      3140.7  2     7.8134  0.02011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

As we can see transmission variable is useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

```

m2 <- update(m1, ~.+fuelType+transmission+engineSize_num,data=df_train[!df_train$mout=="YesMOut",])
m2pet <- update(m1, ~.+fuelType+transmission,data=df_train[!df_train$mout=="YesMOut",])
anova( m2pet, m2, test="Chisq")

```

```

## Analysis of Deviance Table
##
## Model 1: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission
## Model 2: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##   engineSize_num
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3160      3215.4
## 2      3158      3140.7  2     74.674 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

As we can see engineSize\_num variable is useful, because the p-value of anova test is less than 0.05, that means that the 2 models are not equivalent and, therefore, the big one is better.

```

m2 <- update(m1, ~.+fuelType+transmission+engineSize_num,data=df_train[!df_train$mout=="YesMOut",])
anova( m1, m2, test="Chisq")

```

```

## Analysis of Deviance Table
##
## Model 1: Audi ~ mileage + tax + mpg + years_sell
## Model 2: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##   engineSize_num
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3165      3242.1
## 2      3158      3140.7  7     101.4 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
summary(m2)
```

```

##
## Call:
## glm(formula = Audi ~ mileage + tax + mpg + years_sell + fuelType +
##   transmission + engineSize_num, family = "binomial", data = df_train[!df_train$mout ==
##   "YesMOut", ])
##
## Deviance Residuals:
##   Min      1Q      Median      3Q      Max
## -1.3795 -0.7340 -0.6213 -0.3787  2.5279
##
## Coefficients:
##   Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.105e+00 1.273e+00 -0.868 0.38544
## mileage      4.047e-06 5.132e-06  0.789 0.43035
## tax          3.620e-03 3.981e-03  0.909 0.36317

```

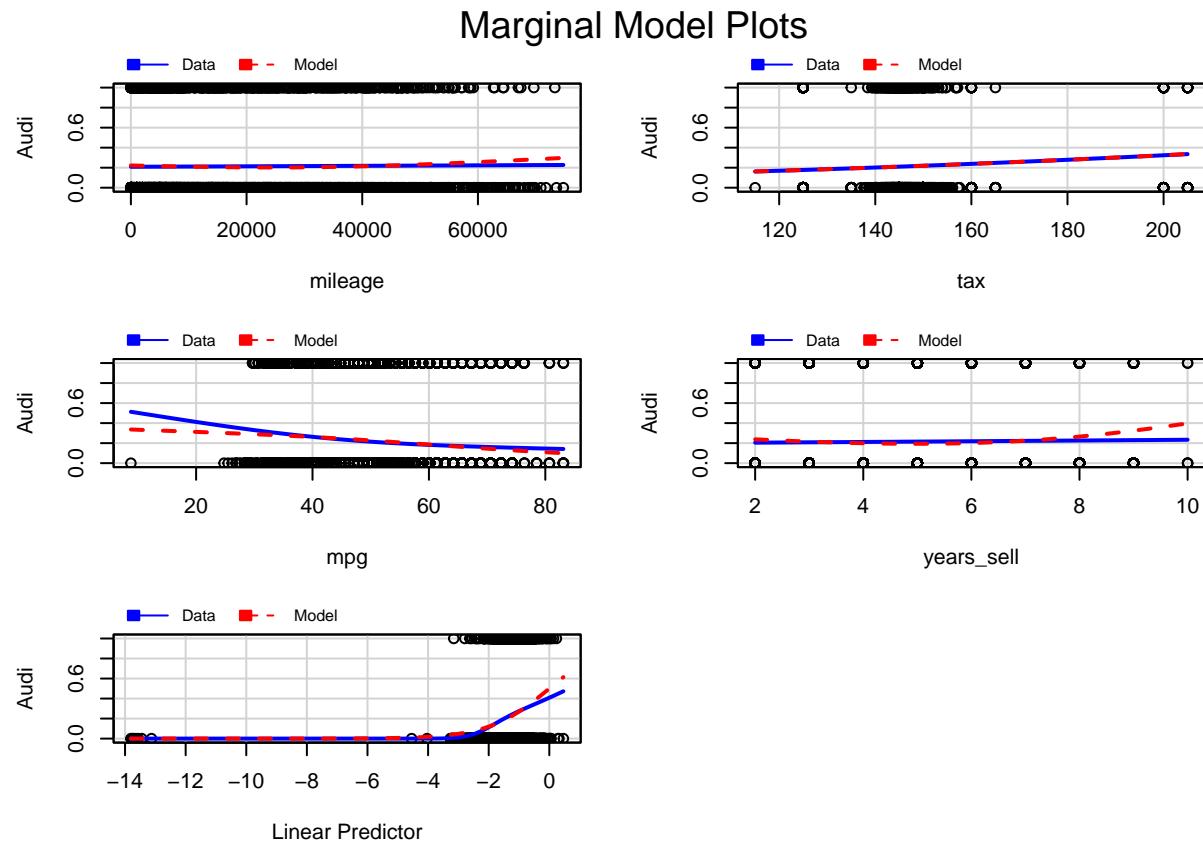
```

## mpg           -5.002e-02  5.766e-03  -8.676  < 2e-16 ***
## years_sell    1.161e-01  4.849e-02   2.394  0.01668 *
## fuelTypef.Fuel-Hybrid -1.658e+00  7.485e-01  -2.215  0.02674 *
## fuelTypef.Fuel-Other  -1.261e+01  2.013e+02  -0.063  0.95003
## fuelTypef.Fuel-Petrol -3.539e-01  1.077e-01  -3.286  0.00102 **
## transmissionf.Trans-SemiAuto -3.090e-01  1.130e-01  -2.734  0.00626 **
## transmissionf.Trans-Automatic -1.244e-01  1.240e-01  -1.004  0.31558
## engineSize_nummedium_engine  5.821e-01  1.087e+00   0.535  0.59235
## engineSize_numsmall_engine   1.780e+00  1.084e+00   1.642  0.10068
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3288.2 on 3169 degrees of freedom
## Residual deviance: 3140.7 on 3158 degrees of freedom
## AIC: 3164.7
##
## Number of Fisher Scoring iterations: 12

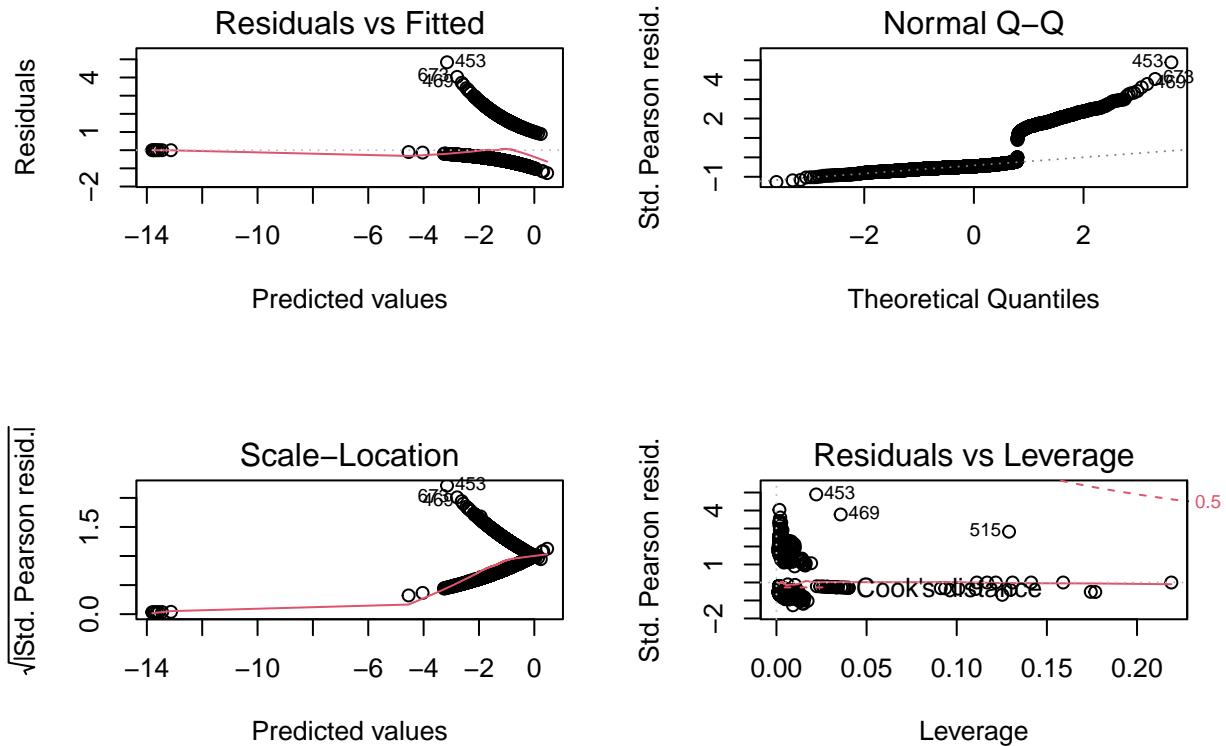
```

```
marginalModelPlots(m2)
```

```
## Warning in mmpls(...): Interactions and/or factors skipped
```



```
par(mfrow=c(2,2))
plot(m2)
```



As we can see the decrease in deviance is significant, because the p-value of anova test is less than 0.05.

We can see: Being hybrid the probability of being Audi decreases by  $(\exp(-1.66) = 0.19 \rightarrow 100*(1-0.19) = 81\%)$  81 %. Being petrol the probability of being Audi decreases by 29.5 %. Being semiAuto the probability of being Audi decreases by 26.7 %.

Also, we can see that the model captures the data well and patterns in the residuals.

## 3.4 Interactions

### 3.4.1 Factors interaction

We are going to see if Audi variable (Y response) is related to fuelType (factor A) and transmission (factor B) variables.

```
m3 <- update(m1, ~.+fuelType*transmission+engineSize_num, data=df_train[!df_train$mout=="YesMOut",])
anova(m2,m3,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##           engineSize_num
## Model 2: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##           engineSize_num + fuelType:transmission
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3158    3140.7
## 2      3154    3122.0  4    18.723 0.0008909 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m3)
```

```
##
## Call:
## glm(formula = Audi ~ mileage + tax + mpg + years_sell + fuelType +
```

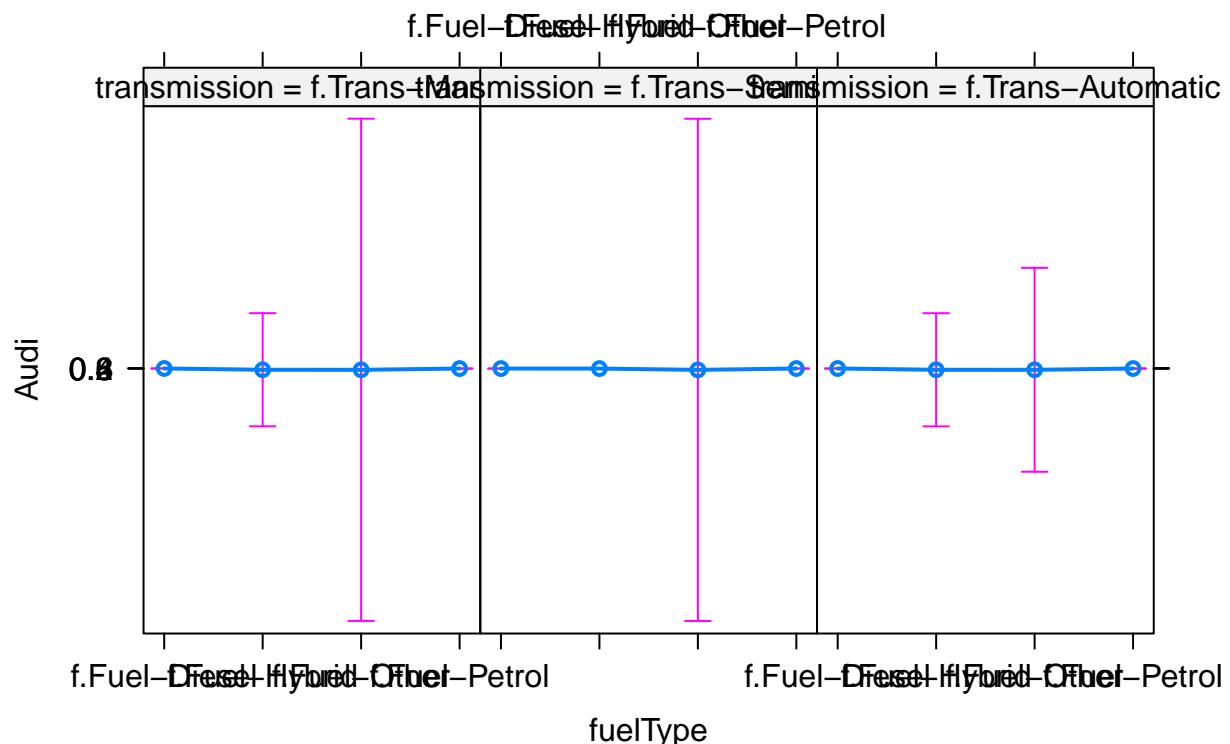
```

##      transmission + engineSize_num + fuelType:transmission, family = "binomial",
##      data = df_train[!df_train$mout == "YesMOOut", ])
##
## Deviance Residuals:
##      Min      1Q   Median      3Q      Max
## -1.4546  -0.7389  -0.6097  -0.3632   2.4425
##
## Coefficients: (2 not defined because of singularities)
##                                     Estimate Std. Error
## (Intercept)                   -1.022e+00 1.274e+00
## mileage                      2.666e-06 5.163e-06
## tax                           3.955e-03 4.008e-03
## mpg                          -4.811e-02 5.795e-03
## years_sell                   1.195e-01 4.870e-02
## fuelTypef.Fuel-Hybrid        -1.437e+01 3.279e+02
## fuelTypef.Fuel-Other          -1.495e+01 1.455e+03
## fuelTypef.Fuel-Petrol         -7.491e-01 1.538e-01
## transmissionf.Trans-SemiAuto -7.452e-01 1.584e-01
## transmissionf.Trans-Automatic -3.744e-01 1.615e-01
## engineSize_nummedium_engine   6.550e-01 1.088e+00
## engineSize_numsmall_engine    1.782e+00 1.085e+00
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto 1.390e+01 3.279e+02
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto    NA      NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto  8.146e-01 2.086e-01
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic    NA      NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic  3.820e-01 1.571e+03
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic  4.661e-01 2.359e-01
## (Intercept)                   -0.802   0.4226
## mileage                      0.516   0.6056
## tax                           0.987   0.3238
## mpg                          -8.302  < 2e-16 ***
## years_sell                   2.453   0.0142 *
## fuelTypef.Fuel-Hybrid        -0.044   0.9651
## fuelTypef.Fuel-Other          -0.010   0.9918
## fuelTypef.Fuel-Petrol         -4.872  1.11e-06 ***
## transmissionf.Trans-SemiAuto -4.705  2.54e-06 ***
## transmissionf.Trans-Automatic -2.318   0.0204 *
## engineSize_nummedium_engine   0.602   0.5474
## engineSize_numsmall_engine    1.643   0.1005
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto  0.042   0.9662
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto    NA      NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto  3.905  9.44e-05 ***
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic    NA      NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic  0.000   0.9998
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic  1.976   0.0482 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3288.2 on 3169 degrees of freedom
## Residual deviance: 3122.0 on 3154 degrees of freedom
## AIC: 3154
##
## Number of Fisher Scoring iterations: 14

plot(allEffects(m3), selection = 6)

```

## fuelType\*transmission effect plot



As we can see the decrease in deviance is significant, because the p-value of anova test is less than 0.05, so we can say that Audi is related to fuelType and transmission.

We can see that the only significant interactions are: The fact that a car is petrol and with semiAuto transmission makes the probability of being Audi increases by ( $\exp(0.81) = 2.25 \rightarrow 100*(2.25-1) = 125\%$ ) 125 %, because the p-value is less than 0.05.

The fact that a car is petrol and with automatic transmission makes the probability of being Audi increases by ( $\exp(0.47) = 1.6 \rightarrow 100*(1.6-1) = 60\%$ ) 60 %, because the p-value is less than 0.05.

### 3.4.2 Factor and covariate interaction

```
m3 <- glm(Audi~mileage+tax+years_sell+fuelType*transmission+fuelType*mpg+engineSize_num,
            family="binomial", data=df_train[!df_train$mout=="YesMOut",])
anova(m2,m3, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##           engineSize_num
## Model 2: Audi ~ mileage + tax + years_sell + fuelType * transmission +
##           fuelType * mpg + engineSize_num
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3158     3140.7
## 2      3151     3115.0  7   25.665 0.0005781 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m3)
```

```
##
## Call:
## glm(formula = Audi ~ mileage + tax + years_sell + fuelType *
##       transmission + fuelType * mpg + engineSize_num, family = "binomial",
```

```

##      data = df_train[!df_train$mout == "YesMOut", ])
##
## Deviance Residuals:
##      Min      1Q  Median      3Q     Max
## -1.5546  -0.7427  -0.6164  -0.3509   2.4782
##
## Coefficients: (2 not defined because of singularities)
##                                         Estimate Std. Error
## (Intercept)                   -3.012e-01  1.310e+00
## mileage                      3.781e-06  5.210e-06
## tax                          3.767e-03  4.016e-03
## years_sell                   1.088e-01  4.920e-02
## fuelTypeef.Fuel-Hybrid      -2.392e+01  3.254e+02
## fuelTypeef.Fuel-Other        -1.597e+01  2.944e+03
## fuelTypeef.Fuel-Petrol       -1.975e+00  5.902e-01
## transmissionf.Trans-SemiAuto -8.186e-01  1.629e-01
## transmissionf.Trans-Automatic -4.459e-01  1.658e-01
## mpg                          -5.691e-02  6.983e-03
## engineSize_nummedium_engine  5.213e-01  1.089e+00
## engineSize_numsmall_engine   1.649e+00  1.085e+00
## fuelTypeef.Fuel-Hybrid:transmissionf.Trans-SemiAuto 1.476e+01  3.252e+02
## fuelTypeef.Fuel-Other:transmissionf.Trans-SemiAuto   NA        NA
## fuelTypeef.Fuel-Petrol:transmissionf.Trans-SemiAuto  1.000e+00  2.265e-01
## fuelTypeef.Fuel-Hybrid:transmissionf.Trans-Automatic  NA        NA
## fuelTypeef.Fuel-Other:transmissionf.Trans-Automatic  4.730e-01  1.572e+03
## fuelTypeef.Fuel-Petrol:transmissionf.Trans-Automatic  6.550e-01  2.525e-01
## fuelTypeef.Fuel-Hybrid:mpg      1.694e-01  2.082e-01
## fuelTypeef.Fuel-Other:mpg      1.861e-02  5.255e+01
## fuelTypeef.Fuel-Petrol:mpg     2.226e-02  1.039e-02
##
## (Intercept)                   -0.230  0.818179
## mileage                      0.726  0.468044
## tax                          0.938  0.348326
## years_sell                   2.211  0.027057 *
## fuelTypeef.Fuel-Hybrid      -0.074  0.941403
## fuelTypeef.Fuel-Other       -0.005  0.995671
## fuelTypeef.Fuel-Petrol      -3.346  0.000821 ***
## transmissionf.Trans-SemiAuto -5.024  5.06e-07 ***
## transmissionf.Trans-Automatic -2.689  0.007158 **
## mpg                          -8.149  3.67e-16 ***
## engineSize_nummedium_engine  0.479  0.632164
## engineSize_numsmall_engine   1.520  0.128593
## fuelTypeef.Fuel-Hybrid:transmissionf.Trans-SemiAuto  0.045  0.963805
## fuelTypeef.Fuel-Other:transmissionf.Trans-SemiAuto   NA        NA
## fuelTypeef.Fuel-Petrol:transmissionf.Trans-SemiAuto  4.417  1.00e-05 ***
## fuelTypeef.Fuel-Hybrid:transmissionf.Trans-Automatic  NA        NA
## fuelTypeef.Fuel-Other:transmissionf.Trans-Automatic  0.000  0.999760
## fuelTypeef.Fuel-Petrol:transmissionf.Trans-Automatic  2.594  0.009489 **
## fuelTypeef.Fuel-Hybrid:mpg     0.813  0.415948
## fuelTypeef.Fuel-Other:mpg     0.000  0.999717
## fuelTypeef.Fuel-Petrol:mpg    2.141  0.032249 *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3288.2 on 3169 degrees of freedom
## Residual deviance: 3115.0 on 3151 degrees of freedom
## AIC: 3153
##
## Number of Fisher Scoring iterations: 14

```

As we can see the decrease in deviance is significant, because the p-value of anova test is less than 0.05, so we can say that Audi is related to fuelType and mpg.

We can see that the only significant interactions are: The fact that a car is petrol and with semiAuto transmission makes the probability of being Audi increases by ( $\exp(1) = 2.72 \rightarrow 100(2.72 - 1) = 172\%$ ) 172 %, because the p-value is less than 0.05. The fact that a car is petrol and with automatic transmission makes the probability of being Audi increases by ( $\exp(0.66) = 1.93 \rightarrow 100(1.93-1) = 93\%$ ) 93 %, because the p-value is less than 0.05. That within/in the petrol category, the mpg variable causes the logarithm of the price increases by ( $\exp(0.022) = 1.02 \rightarrow 100*(1.02-1) = 2$ ) 2 %, because the p-value is less than 0.05.

### 3.5 Best model selection

```

m4 <- step(m3)

## Start:  AIC=3153.01
## Audi ~ mileage + tax + years_sell + fuelType * transmission +
##       fuelType * mpg + engineSize_num
##
##                         Df Deviance     AIC
## - mileage                 1  3115.5 3151.5
## - tax                      1  3115.9 3151.9
## <none>                   3115.0 3153.0
## - fuelType:mpg              3  3122.0 3154.0
## - years_sell                1  3119.9 3155.9
## - fuelType:transmission     4  3139.7 3169.7
## - engineSize_num             2  3178.0 3212.0
##
## Step:  AIC=3151.54
## Audi ~ tax + years_sell + fuelType + transmission + mpg + engineSize_num +
##       fuelType:transmission + fuelType:mpg
##
##                         Df Deviance     AIC
## - tax                      1  3116.5 3150.5
## <none>                   3115.5 3151.5
## - fuelType:mpg              3  3122.2 3152.2
## - fuelType:transmission     4  3140.4 3168.4
## - years_sell                1  3136.2 3170.2
## - engineSize_num             2  3178.2 3210.2
##
## Step:  AIC=3150.46
## Audi ~ years_sell + fuelType + transmission + mpg + engineSize_num +
##       fuelType:transmission + fuelType:mpg
##
##                         Df Deviance     AIC
## <none>                   3116.5 3150.5
## - fuelType:mpg              3  3123.2 3151.2
## - fuelType:transmission     4  3141.2 3167.2
## - years_sell                1  3141.4 3173.4
## - engineSize_num             2  3179.1 3209.1

summary(m4)

##
## Call:
## glm(formula = Audi ~ years_sell + fuelType + transmission + mpg +
##       engineSize_num + fuelType:transmission + fuelType:mpg, family = "binomial",
##       data = df_train[!df_train$mout == "YesMOut", ])
##
## Deviance Residuals:
##      Min      1Q      Median      3Q      Max
## -1.5446 -0.7415 -0.6175 -0.3555  2.4928
##
## Coefficients: (2 not defined because of singularities)

```

```

##                                     Estimate Std. Error
## (Intercept)                   1.937e-01  1.153e+00
## years_sell                     1.447e-01  2.877e-02
## fuelTypef.Fuel-Hybrid          -2.415e+01  3.254e+02
## fuelTypef.Fuel-Other            -1.599e+01  2.937e+03
## fuelTypef.Fuel-Petrol           -1.942e+00  5.879e-01
## transmissionf.Trans-SemiAuto  -8.147e-01  1.627e-01
## transmissionf.Trans-Automatic -4.493e-01  1.657e-01
## mpg                            -5.767e-02  6.781e-03
## engineSize_nummedium_engine    5.385e-01  1.088e+00
## engineSize_numsmall_engine     1.660e+00  1.085e+00
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto 1.478e+01  3.251e+02
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto      NA      NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto    9.998e-01  2.263e-01
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic      NA      NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic     4.617e-01  1.573e+03
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic    6.572e-01  2.525e-01
## fuelTypef.Fuel-Hybrid:mpg          1.730e-01  2.085e-01
## fuelTypef.Fuel-Other:mpg          1.930e-02  5.239e+01
## fuelTypef.Fuel-Petrol:mpg         2.133e-02  1.031e-02
##                                     z value Pr(>|z|)
## (Intercept)                   0.168 0.8666571
## years_sell                     5.031 4.87e-07 ***
## fuelTypef.Fuel-Hybrid          -0.074 0.940835
## fuelTypef.Fuel-Other            -0.005 0.995658
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto  -3.303 0.000958 ***
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto    -5.006 5.55e-07 ***
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto  -2.711 0.006715 **
## mpg                            -8.505 < 2e-16 ***
## engineSize_nummedium_engine    0.495 0.620741
## engineSize_numsmall_engine     1.531 0.125790
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto  0.045 0.963735
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto      NA      NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto    4.418 9.95e-06 ***
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic      NA      NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic     0.000 0.999766
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic    2.603 0.009244 **
## fuelTypef.Fuel-Hybrid:mpg          0.830 0.406672
## fuelTypef.Fuel-Other:mpg          0.000 0.999706
## fuelTypef.Fuel-Petrol:mpg         2.068 0.038614 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
## Null deviance: 3288.2 on 3169 degrees of freedom
## Residual deviance: 3116.5 on 3153 degrees of freedom
## AIC: 3150.5
## 
## Number of Fisher Scoring iterations: 14

```

Our best model (model that has the minimum possible deviance with/using the least number of possible variables) has a deviance of 3116.5.

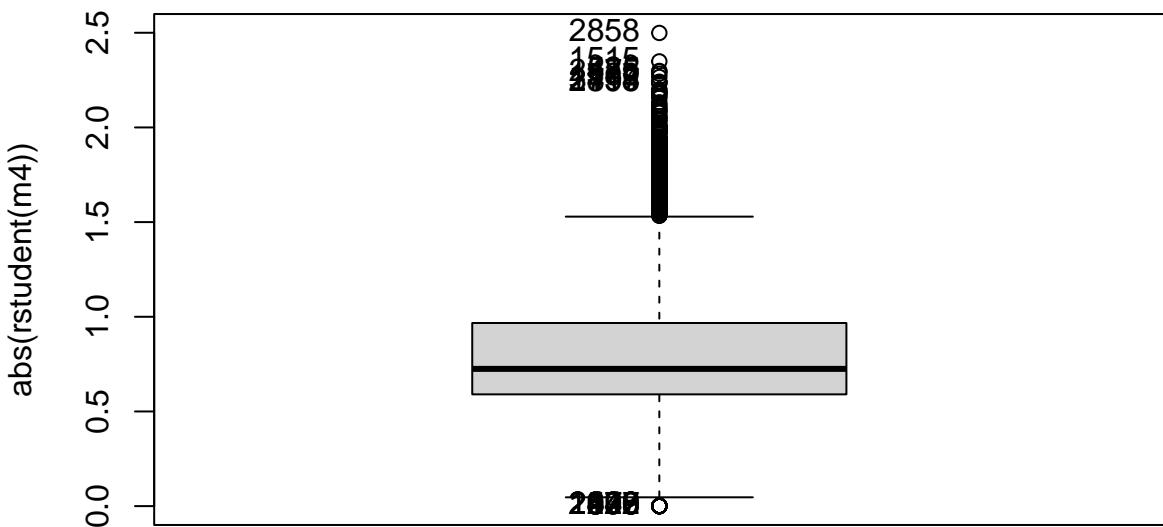
### 3.6 Diagnostics

```

dfwork <- df_train[!df_train$mout=="YesMOut",]

par(mfrow=c(1,1))
Boxplot(abs(rstudent(m4)))

```



```
## [1] 1847 929 2640 578 2422 1292 2079 120 2938 406 2858 1515 585 525 2475
## [16] 1539 2755 3117 1414 2898
```

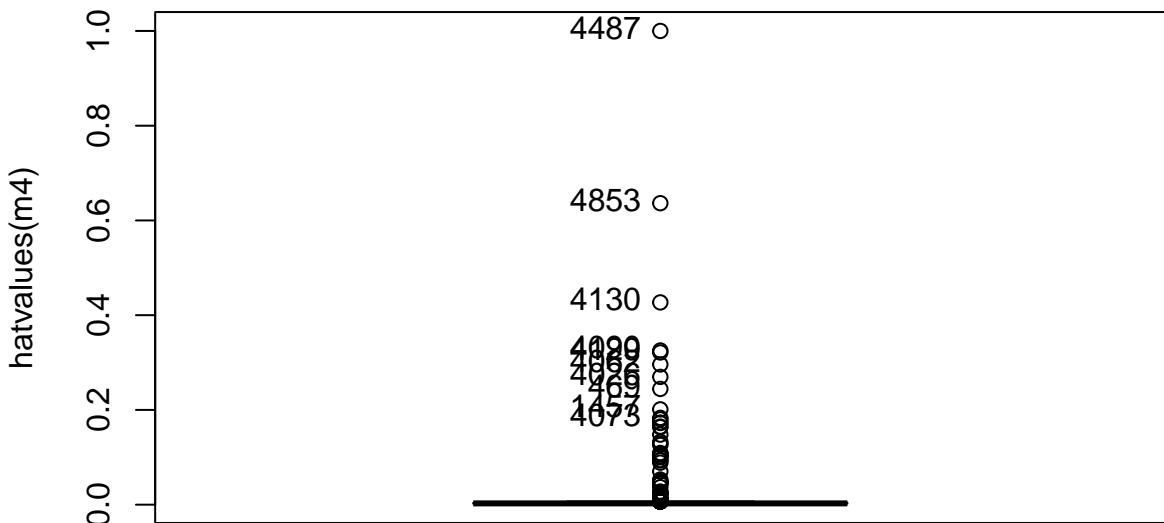
```
llres <- which(abs(rstudent(m4))>2.3);llres
```

```
## 296 673
## 1515 2858
```

```
length(llres)
```

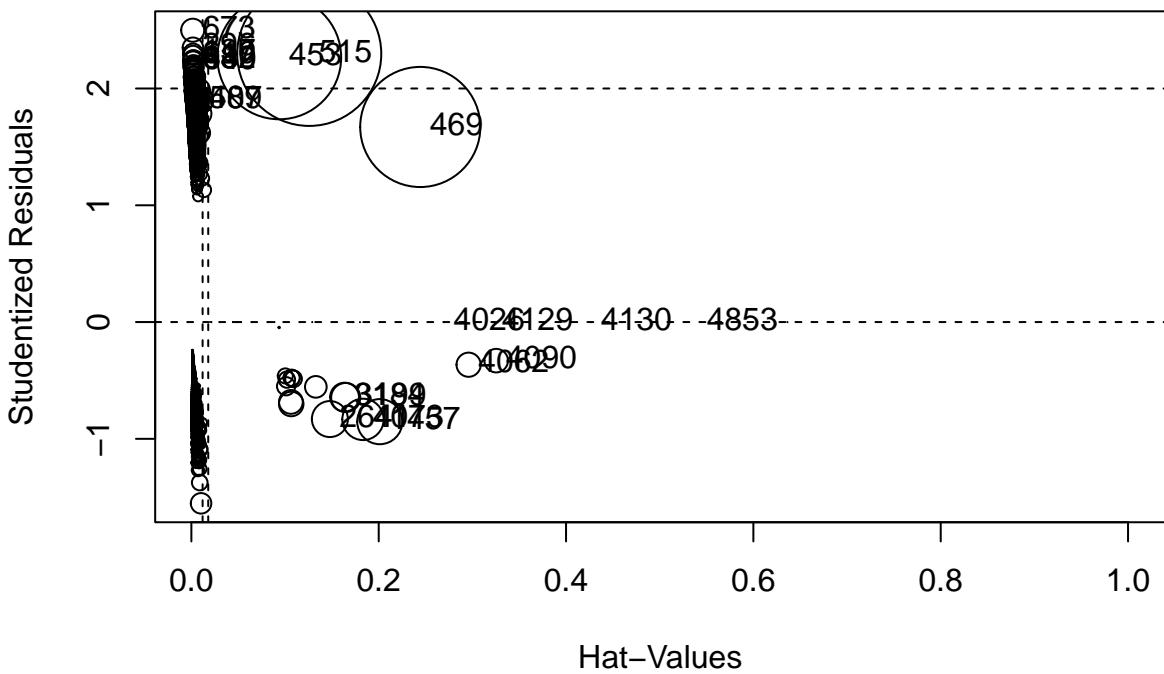
```
## [1] 2
```

```
Boxplot(hatvalues(m4), id=list(labels=row.names(dfwork)))
```



```
## [1] "4487" "4853" "4130" "4090" "4129" "4062" "4026" "469"   "1457" "4073"

influencePlot(m4, id=list(n=10))
```



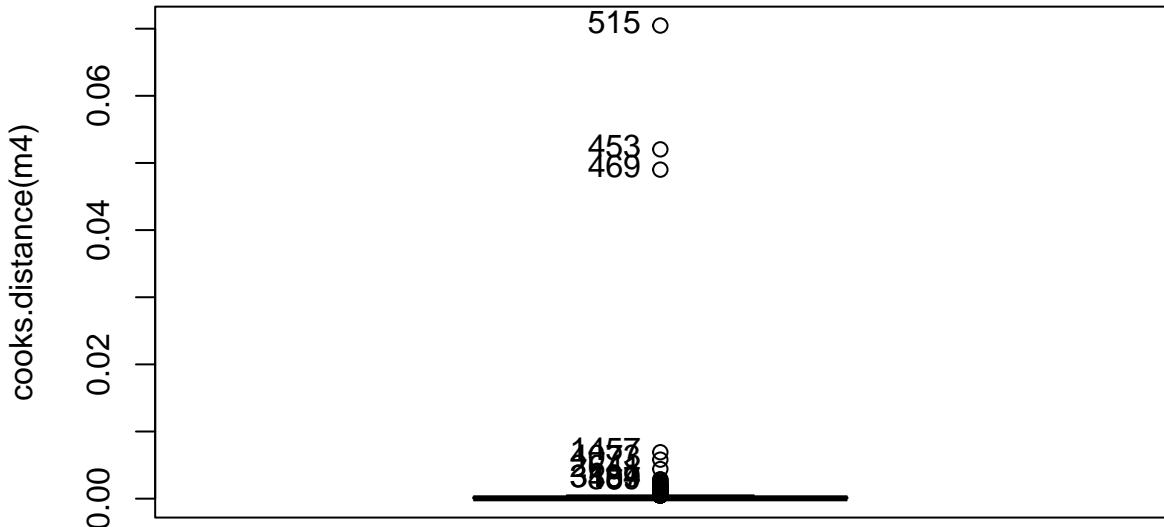
```
##             StudRes        Hat      CookD
## 4130 -0.0006586387 0.426960903 1.208676e-08
```

```

## 3184 -0.6428063500 0.163562794 2.820319e-03
## 489 1.8999970558 0.008552104 2.534071e-03
## 515 2.2964501221 0.125831943 7.046707e-02
## 137 2.2983808864 0.001786319 1.357241e-03
## 2641 -0.8307120179 0.147868032 4.404702e-03
## 3199 -0.6463170903 0.164770856 2.880610e-03
## 1457 -0.8529318526 0.201149886 6.911298e-03
## 4062 -0.3653341578 0.295962419 1.981350e-03
## 4090 -0.3305089636 0.325434897 1.884224e-03
## 682 2.2388520430 0.001800724 1.183851e-03
## 296 2.3488654494 0.001662117 1.430953e-03
## 453 2.2664506546 0.093850400 5.202563e-02
## 4487 NaN 1.0000000000 NaN
## 4026 -0.0006439113 0.269771914 5.207606e-09
## 507 1.8999970558 0.008552104 2.534071e-03
## 4853 -0.0007232717 0.636405952 3.949903e-08
## 4073 -0.8336172288 0.183226560 5.808469e-03
## 4129 -0.0007733337 0.321305178 9.921021e-09
## 610 2.2837248410 0.001691206 1.240339e-03
## 98 2.2405351894 0.002101929 1.385813e-03
## 469 1.6705719361 0.244366660 4.900554e-02
## 673 2.4989786627 0.001435293 1.808261e-03
## 640 2.2314071531 0.002233374 1.439725e-03
## 439 2.2405351894 0.002101929 1.385813e-03

```

```
Boxplot(cooks.distance(m4), id=list(labels=row.names(dfwork)))
```



```
## [1] "515"  "453"  "469"  "1457" "4073" "2641" "3199" "3184" "489"  "507"
```

```
llout<-which(abs(cooks.distance(m4))>0.02);  
length(llout)
```

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

```

llrem<-unique(c(llout,llres));llrem

## [1] 525 1539 2793 1515 2858

m4 <- update(m4,data=dfwork[-llrem,])
summary(m4)

## 
## Call:
## glm(formula = Audi ~ years_sell + fuelType + transmission + mpg +
##     engineSize_num + fuelType:transmission + fuelType:mpg, family = "binomial",
##     data = dfwork[-llrem, ])
## 
## Deviance Residuals:
##      Min      1Q  Median      3Q      Max 
## -1.5536 -0.7417 -0.6149 -0.3402  2.3160 
## 
## Coefficients: (2 not defined because of singularities)
##                                         Estimate Std. Error
## (Intercept)                      -1.342e+01  5.452e+02
## years_sell                         1.461e-01  2.888e-02
## fuelTypef.Fuel-Hybrid             -1.756e+01  1.760e+03
## fuelTypef.Fuel-Other              -1.598e+01  2.944e+03
## fuelTypef.Fuel-Petrol             -1.924e+00  5.896e-01
## transmissionf.Trans-SemiAuto    -8.430e-01  1.637e-01
## transmissionf.Trans-Automatic   -4.454e-01  1.659e-01
## mpg                                -5.833e-02  6.822e-03
## engineSize_nummedium_engine       1.415e+01  5.452e+02
## engineSize_numsmall_engine        1.531e+01  5.452e+02
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto 3.898e-01  6.035e+02
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto      NA      NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto    1.025e+00  2.270e-01
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic    NA      NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic    4.603e-01  1.573e+03
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic   6.616e-01  2.524e-01
## fuelTypef.Fuel-Hybrid:mpg            5.770e-02  3.116e+01
## fuelTypef.Fuel-Other:mpg            1.909e-02  5.254e+01
## fuelTypef.Fuel-Petrol:mpg          2.087e-02  1.034e-02
## 
## (Intercept)                      -0.025  0.98036
## years_sell                         5.061  4.18e-07 ***
## fuelTypef.Fuel-Hybrid             -0.010  0.99204
## fuelTypef.Fuel-Other              -0.005  0.99567
## fuelTypef.Fuel-Petrol             -3.263  0.00110 **
## transmissionf.Trans-SemiAuto    -5.150  2.61e-07 ***
## transmissionf.Trans-Automatic   -2.685  0.00726 **
## mpg                                -8.550  < 2e-16 ***
## engineSize_nummedium_engine       0.026  0.97929
## engineSize_numsmall_engine        0.028  0.97760
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto 0.001  0.99948
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto      NA      NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto    4.515  6.35e-06 ***
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic    NA      NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic   0.000  0.99977
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic   2.621  0.00877 **
## fuelTypef.Fuel-Hybrid:mpg          0.002  0.99852
## fuelTypef.Fuel-Other:mpg          0.000  0.99971
## fuelTypef.Fuel-Petrol:mpg         2.018  0.04362 *
## 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)

```

```

## Null deviance: 3272.7 on 3164 degrees of freedom
## Residual deviance: 3088.5 on 3148 degrees of freedom
## AIC: 3122.5
##
## Number of Fisher Scoring iterations: 14

m0<-glm(Audi ~ 1, family="binomial", data=dfwork[-llrem,])

```

We have 2 outliers on the regression. We have 3 a posteriori influential observations. As we can see in the Boxplot the 515, 453 and 469 observations are the most significant ones.

To achieve the best model we remove the a posteriori influential observations from the model.

These model has a deviance of 3088.5.

### 3.7 Prediction

```
library(ResourceSelection)
```

```
## Warning: package 'ResourceSelection' was built under R version 4.1.3
```

```
## ResourceSelection 0.3-5 2019-07-22
```

```
pred_test <- predict(m4, newdata=df_test, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
library("ROCR")
```

```
## Warning: package 'ROCR' was built under R version 4.1.3
```

```
library("AUC")
```

```
## Warning: package 'AUC' was built under R version 4.1.3
```

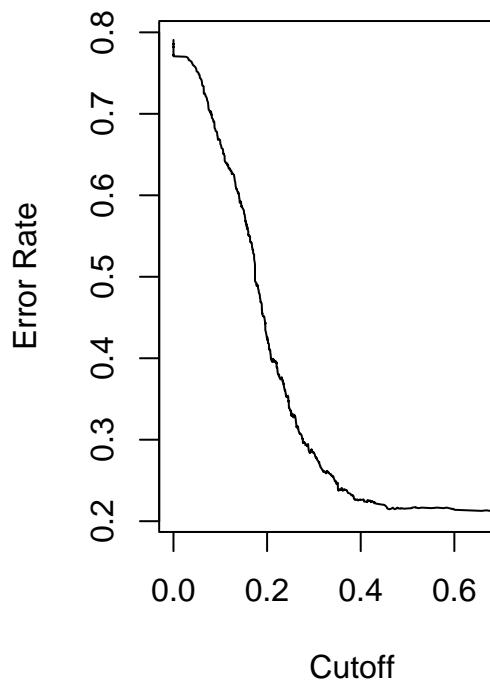
```
## AUC 0.3.2
```

```
## Type AUCNews() to see the change log and ?AUC to get an overview.
```

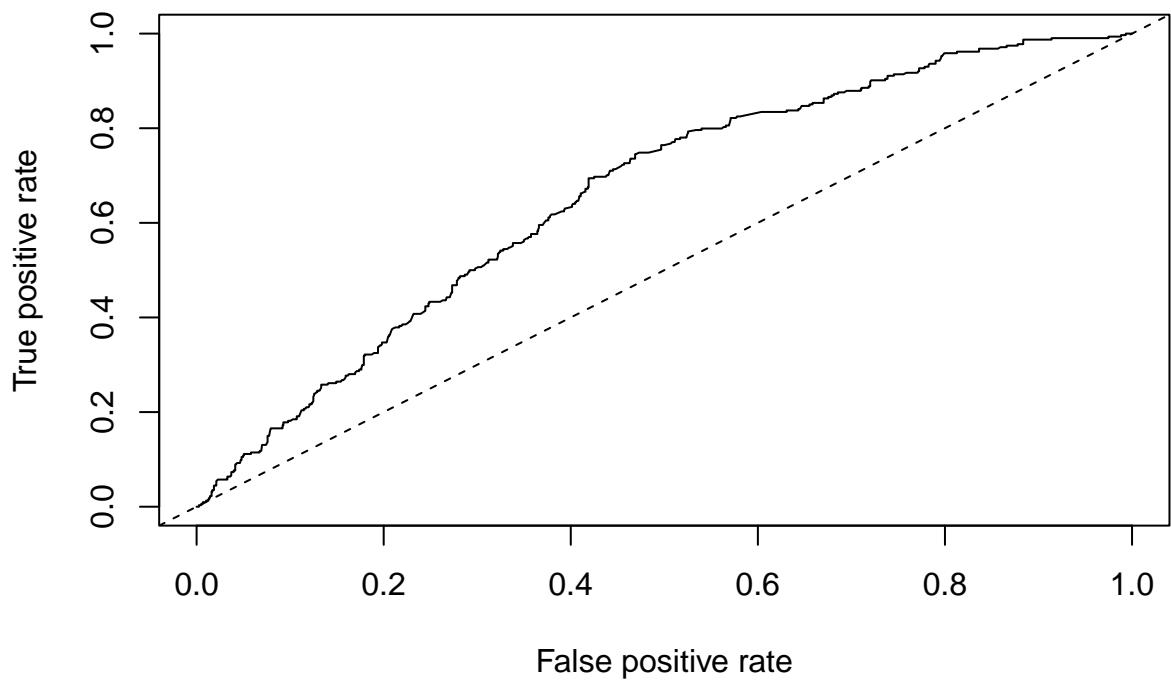
```
dadesroc<-prediction(pred_test,df_test$Audi)
par(mfrow=c(1,2))
performance(dadesroc,"auc",fpr.stop=0.05)
```

```
## A performance instance
## 'Area under the ROC curve'
```

```
plot(performance(dadesroc,"err"))
par(mfrow=c(1,1))
```



```
plot(performance(dadesroc, "tpr", "fpr"))
abline(0,1, lty=2)
```



```
#roc(pred_test,df_test$Audi)
library(cvAUC)
```

```
## Warning: package 'cvAUC' was built under R version 4.1.3
```

```
AUC(pred_test,df_test$Audi)
```

```
## [1] 0.6592853
```

### 3.8 Confusion matrix

```
threshold <- 0.5
audi.est <- ifelse(pred_test<threshold,0,1)
tt<-table(audi.est,df_test$Audi);tt
```

```
##
## audi.est Audi No Audi Yes
##      0     1167     305
##      1      19      9
```

```
100*sum(diag(tt))/sum(tt)
```

```
## [1] 78.4
```

```
# Model na?ve
prob.audi <- predict(m0, newdata=df_test, type="response")
audi.est <- ifelse(prob.audi<0.5,0,1)
tt<-table(audi.est,df_test$Audi);tt
```

```
##
## audi.est Audi No Audi Yes
##      0     1186     314
```

```
100*tt[1,1]/sum(tt)
```

```
## [1] 79.06667
```

Finally, we execute the predictions and generate a confusion matrix with the results of it. To conclude, the diagonal of this confusion matrix shows a performance of 78.4% hit rate in our prediction model against our testing data set, and the model m0 without any coefficient would give us a 79.1% of hit rate. This means that the regressors are not adequate to predict if a car is Audi or not. Curve ROC of our chosen method is displayed above. Also, we can say that the model has a tendency to predict not Audi.