Deliverable 1

Laboratori 1 - Data Preparation

Lorenzo Ricci and Raul Bometon

October 22, 2023

Contents

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

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

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

- manufacturer Factor: Audi, BMW, Mercedes or Volkswagen
- model Car model
- year registration year
- price price in £
- transmission type of gearbox
- mileage distance used
- fuelType engine fuel
- tax road tax
- mpg Consumption in miles per gallon

- engineSize size in litres

2 Load Required Packages for this deliverable

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

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

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

```
## Loading required package: effects

## Loading required package: carData

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

## Loading required package: FactoMineR

## Loading required package: car

## Loading required package: factoextra

## Loading required package: ggplot2

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

## Loading required package: RColorBrewer

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

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

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

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

## Loading required package: ggmap

## The legacy packages maptools, rgdal, and rgeos, underpinning the sp package,
## which was just loaded, were retired in October 2023.
## Please refer to R-spatial evolution reports for details, especially
## https://r-spatial.org/r/2023/05/15/evolution4.html.
## It may be desirable to make the sf package available;
## package maintainers should consider adding sf to Suggests:.

## i Google's Terms of Service: <https://mapsplatform.google.com>
## i Please cite ggmap if you use it! Use `citation("ggmap")` for details.
## Loading required package: ggthemes
##
## Loading required package: knitr
```

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

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

2.1 Select a sample of 5000 records

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

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

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

setwd("C:/Users/renzo/Documents/ADEI")
filepath<-"C:/Users/renzo/Documents/ADEI/"

#Used seed (120700)
load(paste0(filepath,"Deliverable1_Sample_Data.RData"))
```

2.2 Some useful functions

3 Initialization of counts for missings, outliers and errors

Initialization of counts for missings, outliers and errors. All numerical variables have to be checked before.

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


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

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

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

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

4 Univariate Descriptive Analysis

```
summary(df)
```

```
##     model                year          price        transmission
##  Length:5000        Min.   :1998   Min.   :  1200   Length:5000
##  Class :character   1st Qu.:2016   1st Qu.: 14070   Class :character
##  Mode  :character   Median :2017   Median : 19700   Mode  :character
```

```
##                     Mean   :2017   Mean   : 21715
##                     3rd Qu.:2019   3rd Qu.: 26499
##                     Max.   :2020   Max.   :149948
##     mileage         fuelType              tax              mpg
##  Min.   :     1   Length:5000        Min.   :  0.0   Min.   :  5.50
##  1st Qu.:  5921   Class :character   1st Qu.:125.0   1st Qu.: 44.80
##  Median : 16402   Mode  :character   Median :145.0   Median : 53.30
##  Mean   : 23096                      Mean   :125.5   Mean   : 54.45
##  3rd Qu.: 33410                      3rd Qu.:145.0   3rd Qu.: 61.40
##  Max.   :152420                      Max.   :580.0   Max.   :470.80
##    engineSize    manufacturer
##  Min.   :0.00   Length:5000
##  1st Qu.:1.50   Class :character
##  Median :2.00   Mode  :character
##  Mean   :1.93
##  3rd Qu.:2.00
##  Max.   :6.20
```

```r
names(df)
```

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

4.1 Qualitative Variables (Factor) / Categorical

Original numeric variables corresponding to qualitative concepts have to be converted to factors. New factors grouping original levels will be considered very positively. We need to do an analysis of all the variables to be able to identify missings, errors and outliers. We will also try to factorize each variable to make it easier to understand the sample.

4.1.1 1. Model

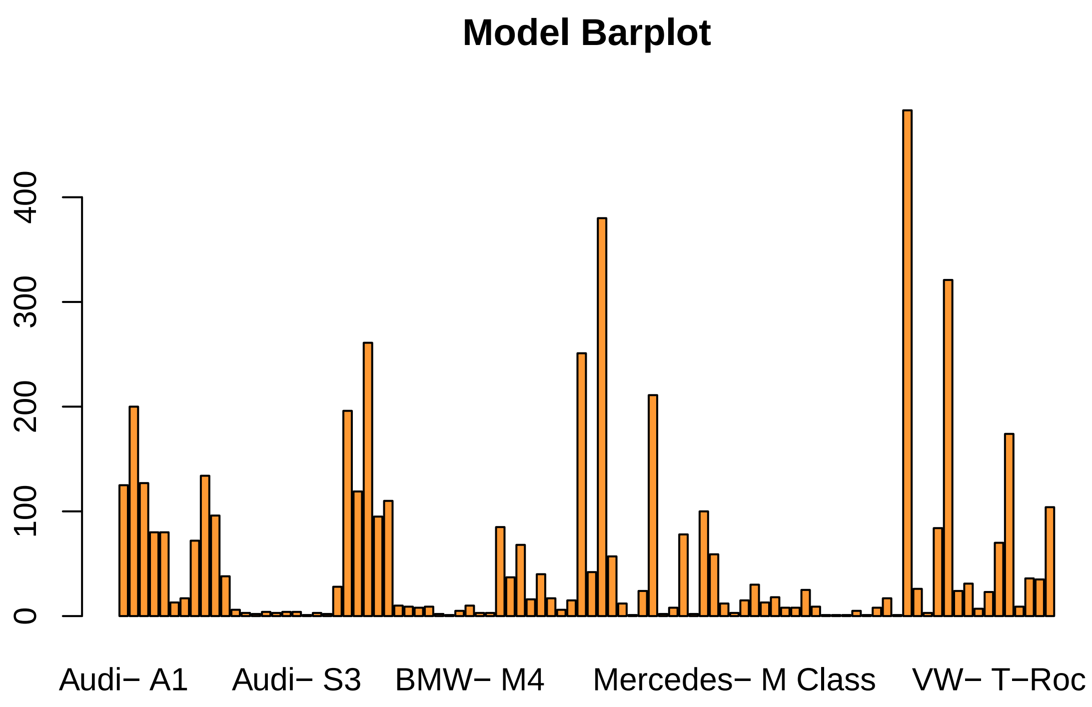
This variable expresses the model of the car, we decide to combine it with the manufacturer to make it more accurate. With the initial summary we see that this variable does not have any missing value, so we proceed to factor it.

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

```
##  [1] "Audi- A1"             "Audi- A3"             "Audi- A4"
##  [4] "Audi- A5"             "Audi- A6"             "Audi- A7"
##  [7] "Audi- A8"             "Audi- Q2"             "Audi- Q3"
## [10] "Audi- Q5"             "Audi- Q7"             "Audi- Q8"
## [13] "Audi- R8"             "Audi- RS3"            "Audi- RS4"
## [16] "Audi- RS5"            "Audi- RS6"            "Audi- S3"
## [19] "Audi- S8"             "Audi- SQ5"            "Audi- SQ7"
## [22] "Audi- TT"             "BMW- 1 Series"        "BMW- 2 Series"
## [25] "BMW- 3 Series"        "BMW- 4 Series"        "BMW- 5 Series"
## [28] "BMW- 6 Series"        "BMW- 7 Series"        "BMW- 8 Series"
## [31] "BMW- i3"              "BMW- i8"              "BMW- M2"
## [34] "BMW- M3"              "BMW- M4"              "BMW- M5"
## [37] "BMW- M6"              "BMW- X1"              "BMW- X2"
## [40] "BMW- X3"              "BMW- X4"              "BMW- X5"
## [43] "BMW- X6"              "BMW- X7"              "BMW- Z4"
## [46] "Mercedes- A Class"    "Mercedes- B Class"    "Mercedes- C Class"
## [49] "Mercedes- CL Class"   "Mercedes- CLA Class"  "Mercedes- CLC Class"
## [52] "Mercedes- CLS Class"  "Mercedes- E Class"    "Mercedes- G Class"
## [55] "Mercedes- GL Class"   "Mercedes- GLA Class"  "Mercedes- GLB Class"
## [58] "Mercedes- GLC Class"  "Mercedes- GLE Class"  "Mercedes- GLS Class"
## [61] "Mercedes- M Class"    "Mercedes- S Class"    "Mercedes- SL CLASS"
## [64] "Mercedes- SLK"        "Mercedes- V Class"    "Mercedes- X-CLASS"
```

```
## [67] "VW- Amarok"          "VW- Arteon"          "VW- Beetle"
## [70] "VW- Caddy"           "VW- Caddy Life"      "VW- Caddy Maxi"
## [73] "VW- Caddy Maxi Life" "VW- California"      "VW- Caravelle"
## [76] "VW- CC"              "VW- Fox"             "VW- Golf"
## [79] "VW- Golf SV"         "VW- Jetta"           "VW- Passat"
## [82] "VW- Polo"            "VW- Scirocco"        "VW- Sharan"
## [85] "VW- Shuttle"         "VW- T-Cross"         "VW- T-Roc"
## [88] "VW- Tiguan"          "VW- Tiguan Allspace" "VW- Touareg"
## [91] "VW- Touran"          "VW- Up"
```

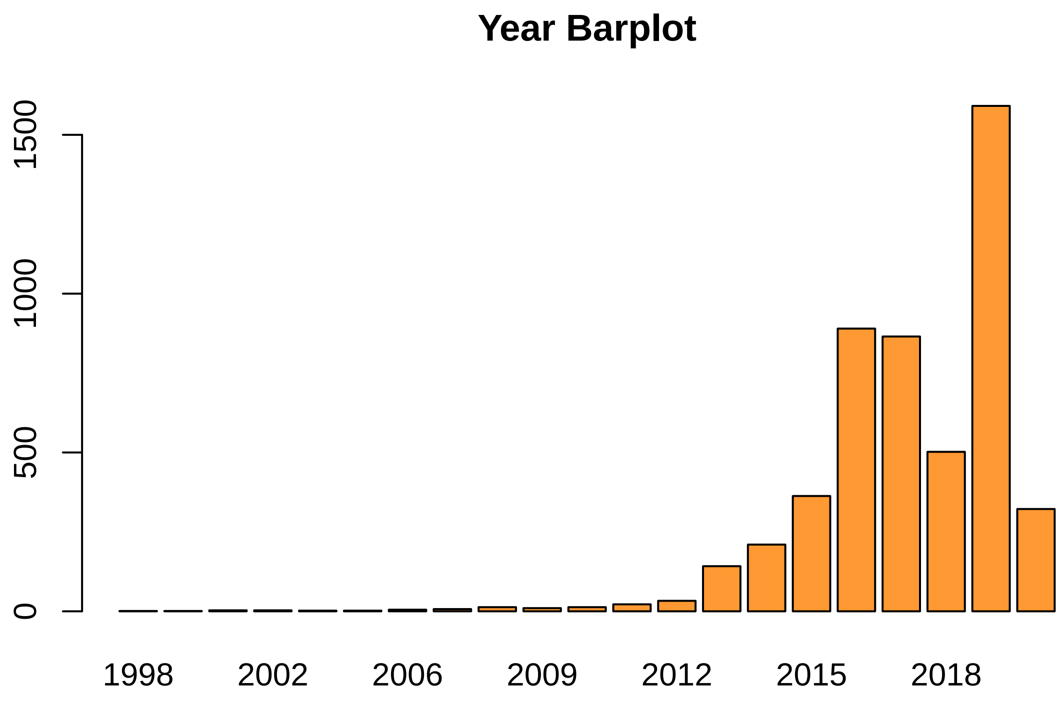
```r
barplot(summary(df$model),main="Model Barplot",col = "#FF9933")
```



4.1.2 2. Year

This variable expresses the different years that we can have as numerical values. Also we create a new variable "age" from year in order to better analyze the data later. We do so by subtracting the corresponding year from 2020 which is the year this data was taken.

```r
df$age <-  2020 - df$year
df$year<-factor(df$year)
barplot(summary(df$year),main="Year Barplot",col = "#FF9933")
```

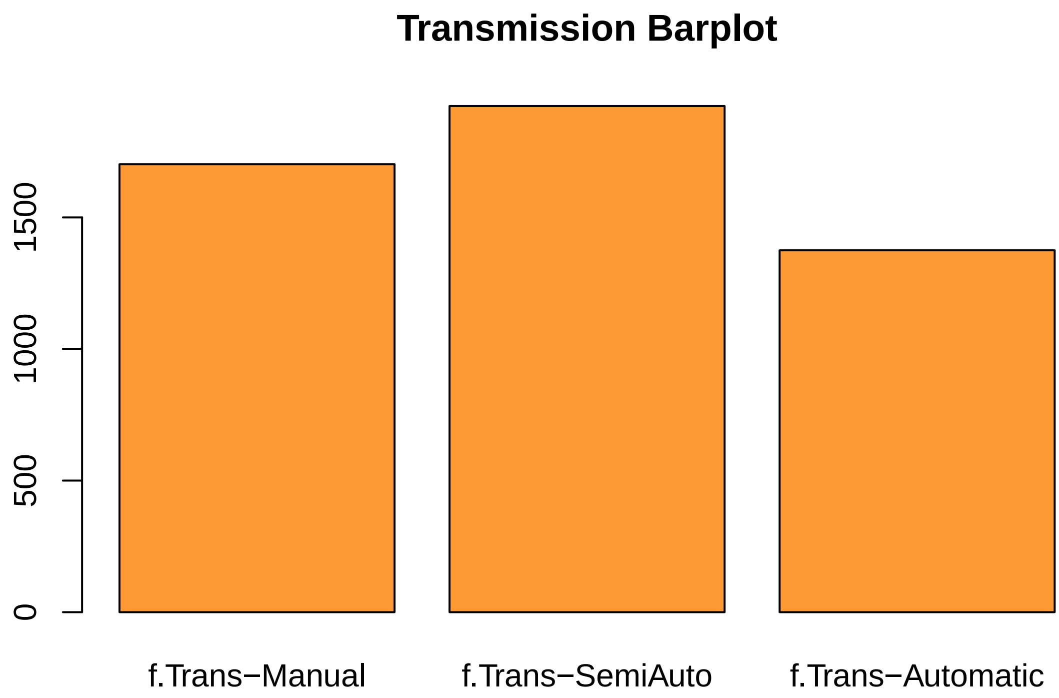


4.1.3 4. Transmission

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

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

```
df$transmission <- factor( df$transmission, levels = c("Manual","Semi-Auto","Automatic"),labels = paste(
barplot(summary(df$transmission),main="Transmission Barplot",col = "#FF9933")
```



4.1.4 6. Fuel Type

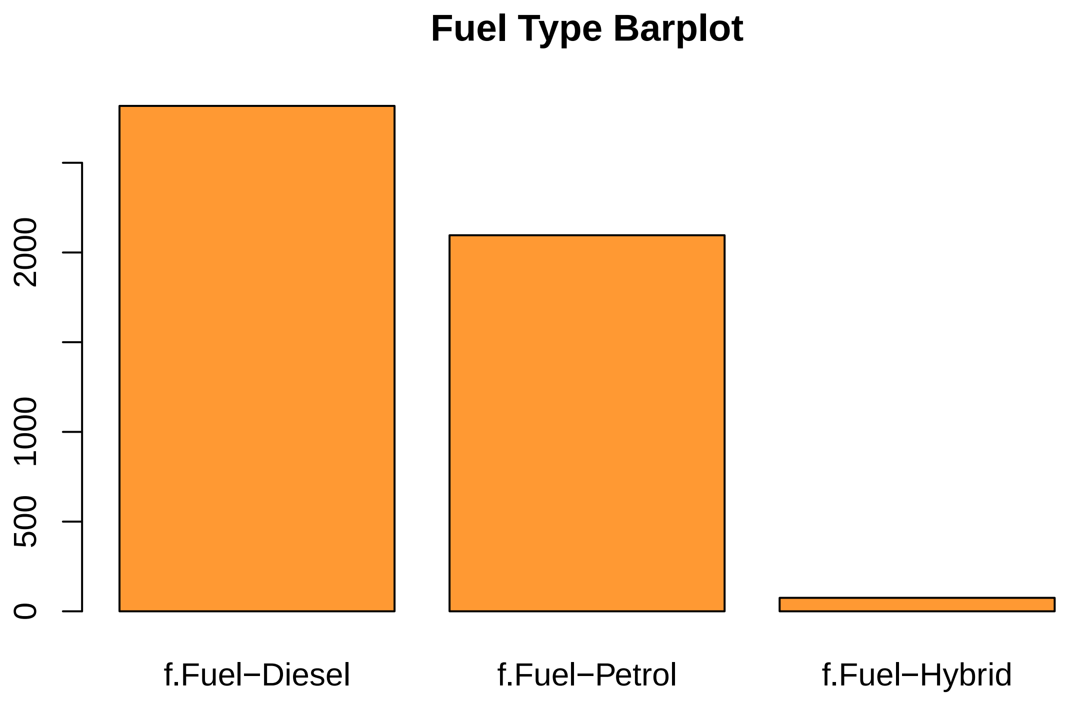
```
df$fuelType <- factor( df$fuelType )
levels(df$fuelType)
```

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

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

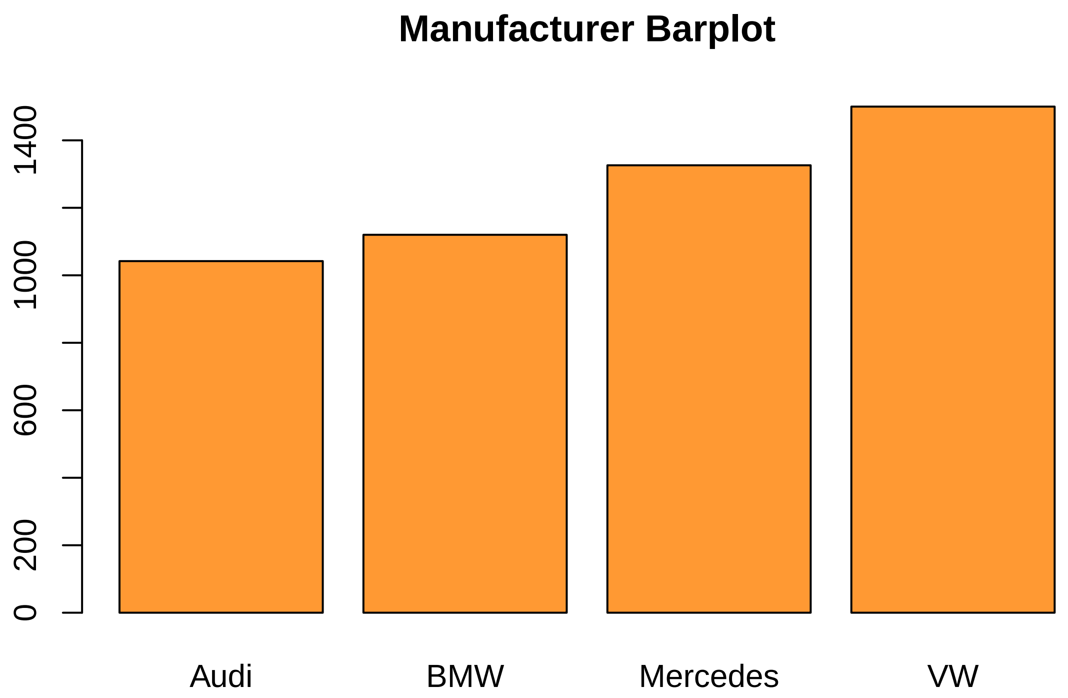
4.1.4.1 Error detection We have taken into account that initially there are 12 missing values of fuel type, in order to analyze the data we will delete this values.

```
sel <- which( is.na( df$fuelType ) )
imis[sel]<-imis[sel]+1
jmis[6]<-length(sel)
df <- df[ -sel, ]
barplot(summary(df$fuelType),main="Fuel Type Barplot",col = "#FF9933")
```



4.1.5 10. Manufacturer

```
df$manufacturer <- factor( df$manufacturer )
barplot(summary(df$manufacturer),main="Manufacturer Barplot",col = "#FF9933")
```



4.2 Quantitative Variables

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

4.2.1 New Variable Age

Further analysis of the new variable age.

```r
summary(df$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    1.00    3.00    2.81    4.00   22.00
```

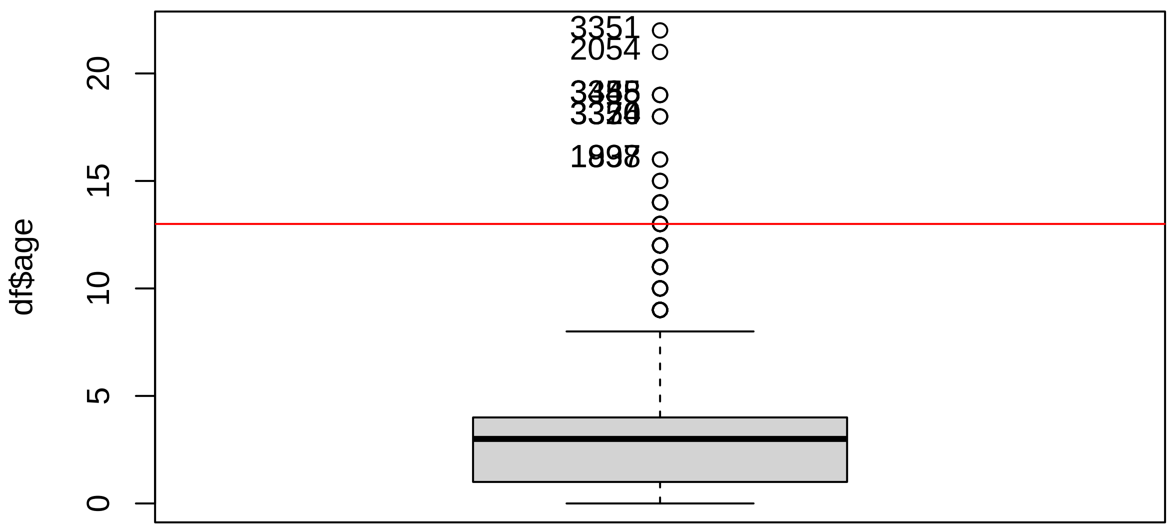
We see on the summary that there are not NA values, so we proceed to the outlier and error detection.

4.2.1.1 Outlier Detection In order to evaluate our data, we decide to set 10+ years old cars as outliers.

```r
Boxplot(df$age)
```

```
##  [1] 3351 2054 3358 3385 3445 3326 3350 3374 1837 1998
```

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




```r
llout<-which((df$age<0)|(df$age>10))
iouts[llout]<-iouts[llout]+1
jouts[11]<-length(llout)
df[llout,"age"]<-NA
```

4.2.2 3. Price

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1200   14000   19700   21719   26500  149948
```

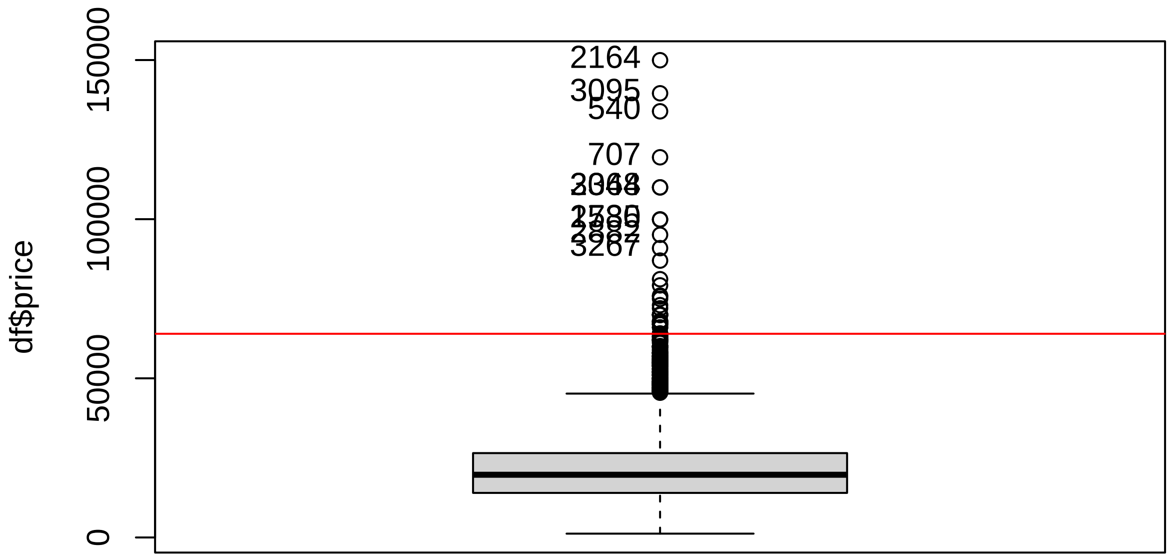
We see on the summary that there are no NA values, so we proceed to the outlier and error detection.

4.2.2.1 Outlier Detection In order to evaluate our data, we decide to set 60000$+ priced cars as outliers.

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

```
##  [1] 2164 3095  540  707 2368 3044 1580 2735 2882 3267
```

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




```r
llout<-which((df$price<0)|(df$price>60000))
iouts[llout]<-iouts[llout]+1
jouts[3]<-length(llout)
df[llout,"price"]<-NA
```

4.2.2.2 Error detection Remove the rows with NA in price because the variable to describe cannot have NA. Further it is better not to make imputations in the case of the target variable

```r
sel <- which( is.na( df$price ) )
df <- df[ -sel, ]
```

4.2.3 5. Mileage

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

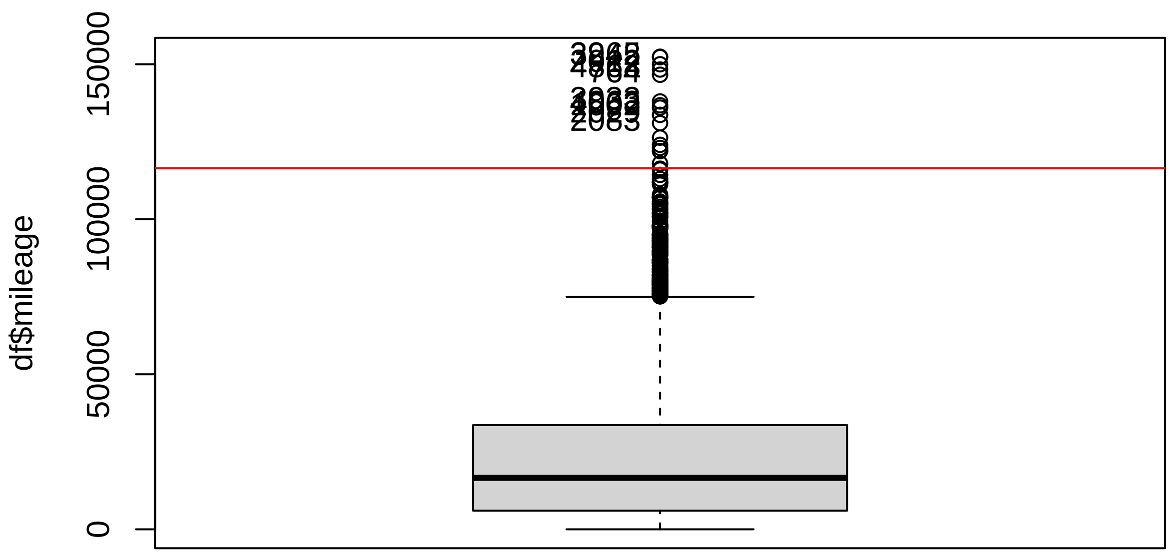
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1    6000   16584   23279   33622  152420
```

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

4.2.3.1 Outlier detection

```
##  [1] 3965 3242 2012 4863  764 3933 4063 1002 2029 2083
```

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




```
llout<-which((df$mileage<0)|(df$mileage>80000))
iouts[llout]<-iouts[llout]+1
jouts[5]<-length(llout)
df[llout,"mileage"]<-NA
```

4.2.4 7. Tax

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.0   125.0   145.0   125.2   145.0   580.0
```

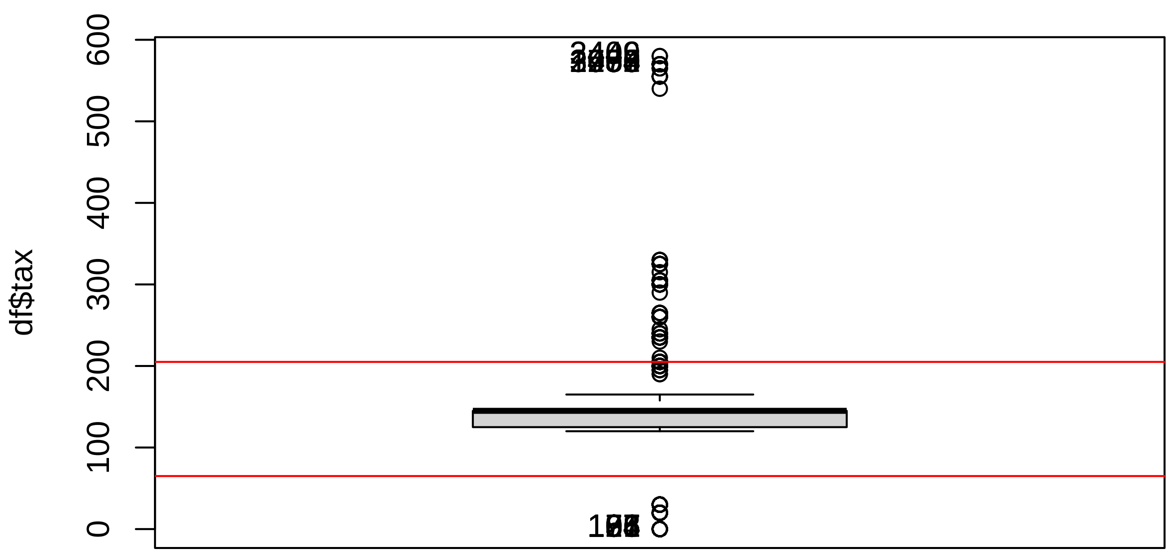
We see on the summary that there are not NA values, so we proceed to the outlier and error detection.

4.2.4.1 Outlier Detection In order to evaluate our data, we decide to set less than 50$ and more than 200$ taxed cars as outliers.

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

```
##  [1]   21   24   68   77  106  107  131  156  185  196 2449 3406  486  972 1758
## [16] 1990 2005 2058 3277 3284
```

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




```
llout<-which((df$tax<50)|(df$tax>200))
iouts[llout]<-iouts[llout]+1
jouts[7]<-length(llout)
df[llout,"tax"]<-NA
```

4.2.5 8. MPG

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     5.5    45.6    53.3    54.4    61.4   470.8
```

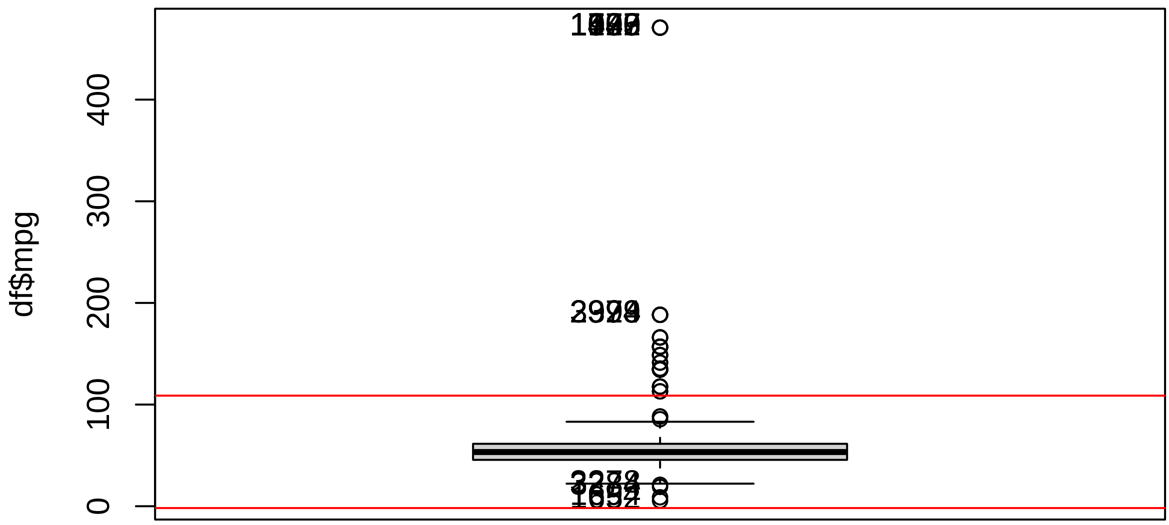
We see on the summary that there are no NA values, so we proceed to the outlier and error detection.

4.2.5.1 Outlier Detection In order to evaluate our data, we decide to set 120+ mpg cars as outliers.

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

```
##  [1] 1652 1694 3278 3284 1095 1109 1420 1543 1737 1972 2979 2993 2994 3324
```

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



```r
llout<-which((df$mpg<0)|(df$mpg>120))
iouts[llout]<-iouts[llout]+1
jouts[8]<-length(llout)
df[llout,"mpg"]<-NA
```

4.2.6 9. Engine Size

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.500   2.000   1.915   2.000   6.200
```

4.2.6.1 Error detection An engine size of 0.0 seems to be an error for non-electric cars.

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

```
##             model year price      transmission mileage      fuelType tax   mpg
## 7517     Audi- Q3 2020 29944    f.Trans-Manual    1500 f.Fuel-Petrol 145  40.9
## 7522     Audi- Q5 2020 49790 f.Trans-Automatic    1500 f.Fuel-Petrol 135 117.7
## 7592     Audi- Q5 2019 33390 f.Trans-Automatic      45 f.Fuel-Diesel 145  39.2
## 11290     BMW- i3 2017 19998 f.Trans-Automatic   41949 f.Fuel-Hybrid 140    NA
## 11447     BMW- i3 2017 19998 f.Trans-Automatic   41146 f.Fuel-Hybrid  NA    NA
## 14582     BMW- i3 2017 18500 f.Trans-Automatic   36429 f.Fuel-Hybrid  NA    NA
## 15845     BMW- i3 2017 21444 f.Trans-Automatic   22063 f.Fuel-Hybrid  NA    NA
## 17483     BMW- i3 2017 21494 f.Trans-Automatic   16867 f.Fuel-Hybrid 135    NA
## 19928     BMW- i3 2015 12500 f.Trans-Automatic   79830 f.Fuel-Hybrid  NA    NA
## 20650     BMW- X5 2016 39948 f.Trans-Automatic   49000 f.Fuel-Petrol  NA  25.4
```

```
## 40920 VW- Passat 2017 16000    f.Trans-Manual   13593 f.Fuel-Diesel 150   68.9
##       engineSize manufacturer age
## 7517           0         Audi   0
## 7522           0         Audi   0
## 7592           0         Audi   1
## 11290          0          BMW   3
## 11447          0          BMW   3
## 14582          0          BMW   3
## 15845          0          BMW   3
## 17483          0          BMW   3
## 19928          0          BMW   5
## 20650          0          BMW   4
## 40920          0           VW   3
```

```r
sel<-which(df$engineSize ==0)
ierrs[sel]<-ierrs[sel]+1
jerrs[9]<-length(sel)
sel
```

```
##  [1]  731  732  743 1095 1109 1420 1543 1737 1972 2058 4058
```

```r
df[sel,"engineSize"]<-NA
selmiss <- sel
```

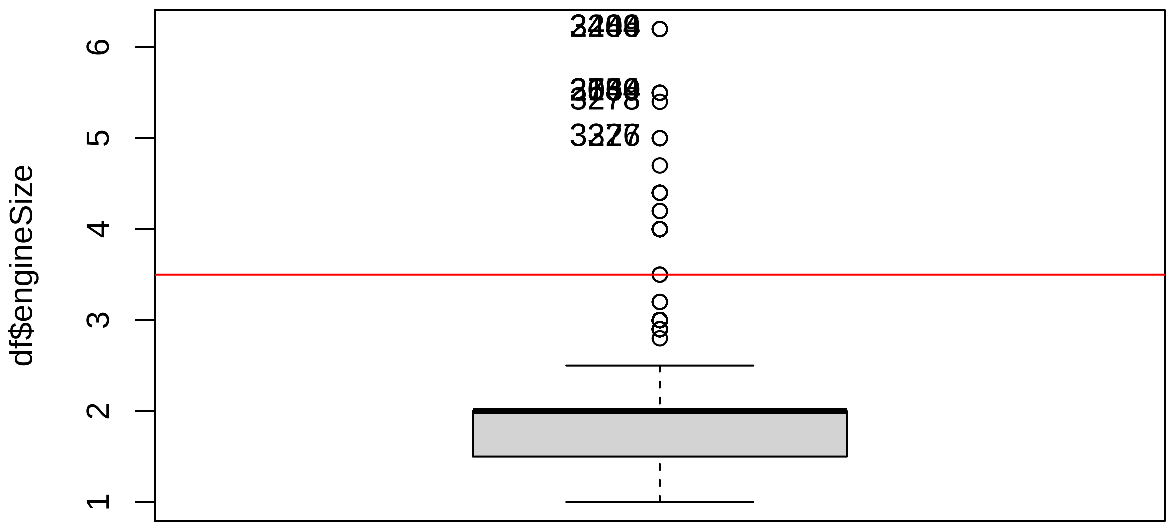
4.2.6.2 Outlier Detection

In order to evaluate our data, we decide to set 3.5+ engine sized cars as outliers.

```r
Boxplot(df$engineSize)
```

```
##  [1] 2449 3284 3406 2134 2544 2760 3059 3278 3277 3326
```

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



```r
llout<-which((df$engineSize<0)|(df$engineSize>3.5))
iouts[llout]<-iouts[llout]+1
jouts[9]<-length(llout)
df[llout,"engineSize"]<-NA
```

5 Data Quality Report

5.1 Per variable

Per each variable, we have to count the following: • number of missing values • number of errors (including inconsistencies) • number of outliers • rank variables according the sum of missing values (and errors).

5.1.1 Number of missing values

```r
#missings_ranking_sortlist <- sort.list(jmis, decreasing = TRUE)
#for (j in missings_ranking_sortlist) {
#print(paste(names(df)[j], " : ", mis1$mis_col$mis_x[j]))
#}
```

5.1.2 Number of errors

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

```
## [1] "engineSize  :  11"
## [1] "model  :  0"
## [1] "year  :  0"
## [1] "price  :  0"
## [1] "transmission  :  0"
## [1] "mileage  :  0"
## [1] "fuelType  :  0"
## [1] "tax  :  0"
## [1] "mpg  :  0"
## [1] "manufacturer  :  0"
## [1] "age  :  0"
```

5.1.3 Number of outliers per each variable

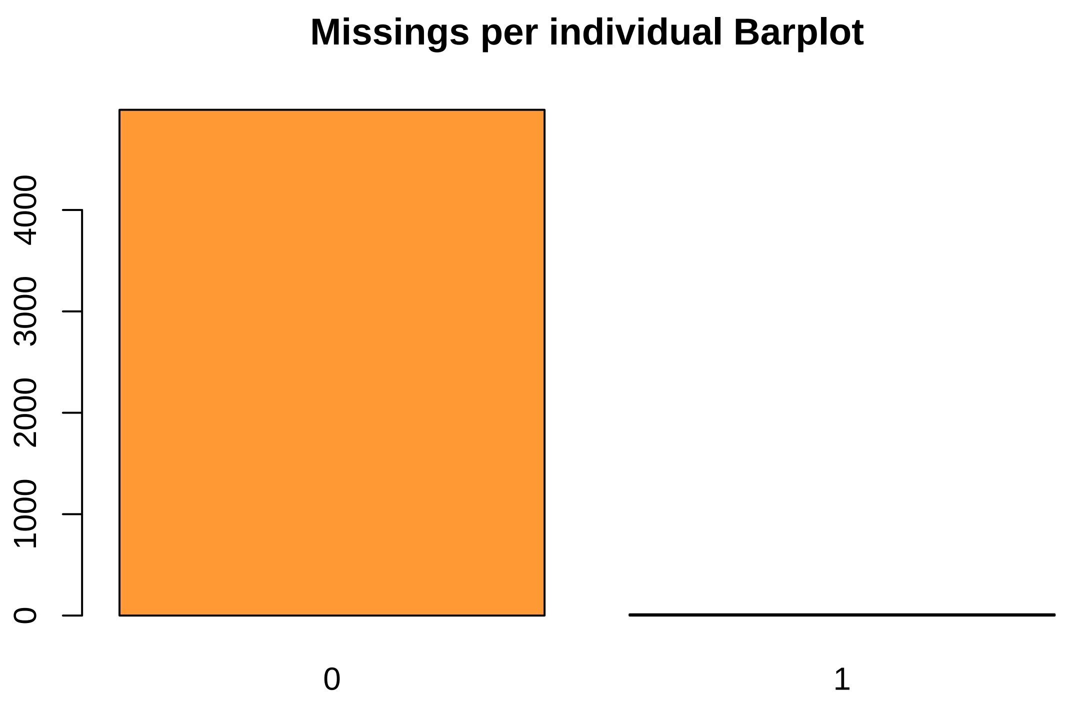
```r
errors_ranking_sortlist <- sort.list(jouts, decreasing = TRUE)
for (j in errors_ranking_sortlist) {
if(!is.na(names(df)[j])) print(paste(names(df)[j], " : ", jouts[j]))
}
```

```
## [1] "tax  :  1287"
## [1] "mileage  :  121"
## [1] "mpg  :  49"
## [1] "price  :  48"
## [1] "age  :  47"
## [1] "engineSize  :  45"
## [1] "model  :  0"
## [1] "year  :  0"
## [1] "transmission  :  0"
## [1] "fuelType  :  0"
## [1] "manufacturer  :  0"
```

##Per Individual Per each individuals, we have to count the following: • number of missing values • number of errors • number of outliers

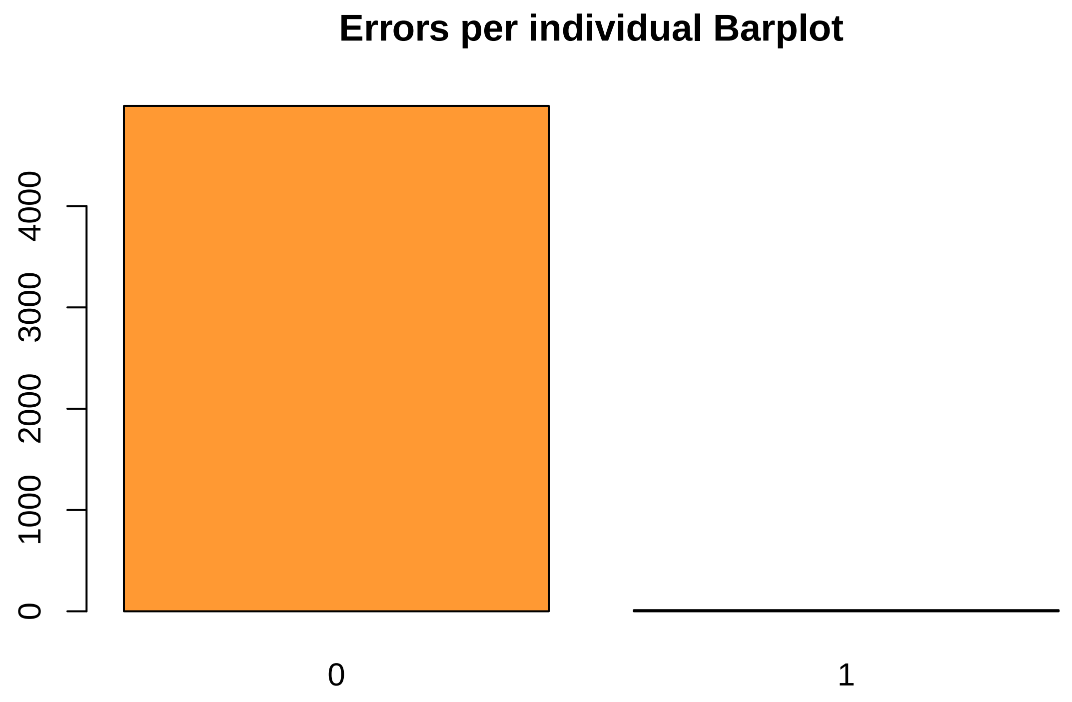
5.1.4 Number of missing values

```
# table(imis)
barplot(table(imis),main="Missings per individual Barplot",col = "#FF9933")
```



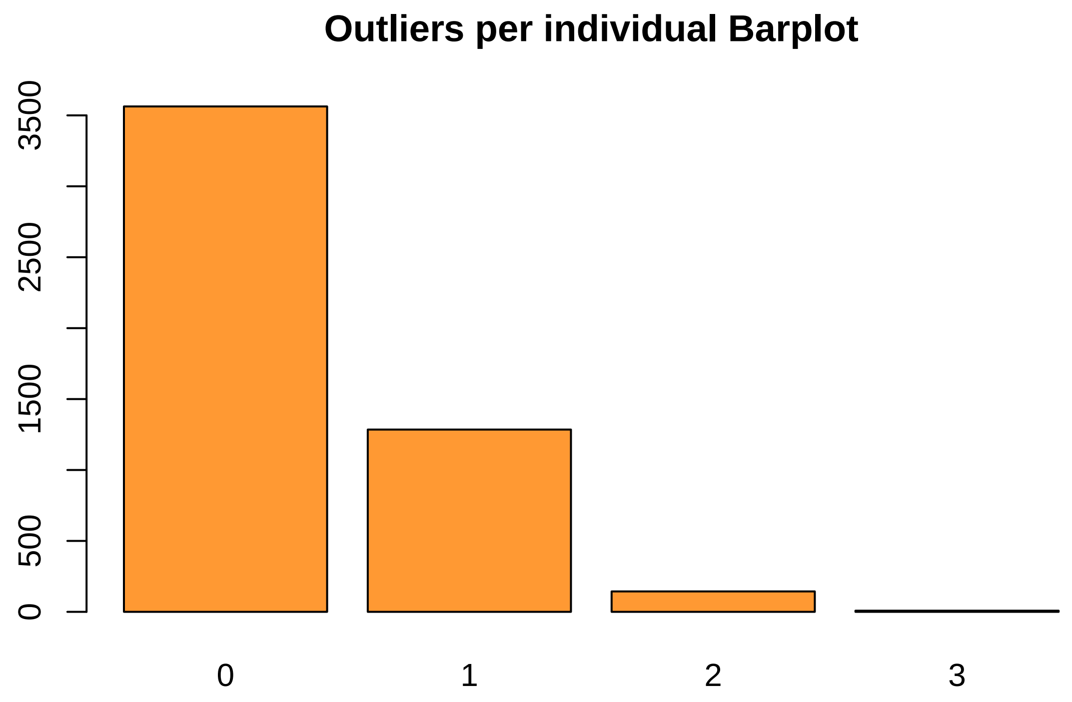
5.1.5 Number of errors

```
# table(ierrs)
barplot(table(ierrs),main="Errors per individual Barplot",col = "#FF9933")
```



5.1.6 Number of outliers

```
# table(iouts)
barplot(table(iouts),main="Outliers per individual Barplot",col = "#FF9933")
```



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

```r
total_missings <- 0; total_outliers <- 0; total_errors <- 0;
for (m in imis) {total_missings <- total_missings + m}
for (o in iouts) {total_outliers <- total_outliers + o}
for (e in ierrs) {total_errors <- total_errors + e}
```

Printing of the variables:

```r
total_missings
```

```
## [1] 12
```

```r
total_outliers
```

```
## [1] 1597
```

```r
total_errors
```

```
## [1] 11
```

6 Imputation

6.1 Imputation of numeric variables

```r
library(missMDA)
# Now one by one describe vars and put them on lists
names(df)
```

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

```r
vars_con<-names(df)[c(3,5,7:9, 11)]
vars_dis<-names(df)[c(1:2, 4, 6, 10)]
vars_res<-names(df)[c(3)]

summary(df[,vars_con])
```

```
##      price          mileage           tax              mpg
##  Min.   : 1200   Min.   :    1   Min.   :120.0   Min.   :  5.50
##  1st Qu.:14000   1st Qu.: 5859   1st Qu.:145.0   1st Qu.: 44.80
##  Median :19499   Median :15948   Median :145.0   Median : 53.30
##  Mean   :21177   Mean   :21369   Mean   :146.7   Mean   : 53.03
##  3rd Qu.:26247   3rd Qu.:32129   3rd Qu.:145.0   3rd Qu.: 61.40
##  Max.   :59995   Max.   :80000   Max.   :200.0   Max.   :117.70
##                  NA's   :121     NA's   :1287    NA's   :49
##    engineSize         age
##  Min.   :1.000   Min.   : 0.000
##  1st Qu.:1.500   1st Qu.: 1.000
##  Median :2.000   Median : 3.000
##  Mean   :1.895   Mean   : 2.723
##  3rd Qu.:2.000   3rd Qu.: 4.000
##  Max.   :3.500   Max.   :10.000
##  NA's   :56      NA's   :47
```

```r
res.impca<-imputePCA(df[,vars_con],ncp=5)
summary(res.impca$completeObs)
```

```
##      price          mileage           tax              mpg
##  Min.   : 1200   Min.   :     1   Min.   :120.0   Min.   :  5.50
##  1st Qu.:14000   1st Qu.: 6000   1st Qu.:145.0   1st Qu.: 45.60
##  Median :19499   Median :16584   Median :145.0   Median : 53.30
##  Mean   :21177   Mean   :22033   Mean   :146.8   Mean   : 53.02
##  3rd Qu.:26247   3rd Qu.:33499   3rd Qu.:147.3   3rd Qu.: 61.40
##  Max.   :59995   Max.   :82702   Max.   :200.0   Max.   :117.70
##    engineSize         age
##  Min.   :1.000   Min.   : 0.000
##  1st Qu.:1.500   1st Qu.: 1.000
##  Median :2.000   Median : 3.000
##  Mean   :1.906   Mean   : 2.758
##  3rd Qu.:2.000   3rd Qu.: 4.000
##  Max.   :4.144   Max.   :10.000
```

```
# Check one by one
res.impca$completeObs[ selmiss,"engineSize"]
```

```
##     7517     7522     7592    11290    11447    14582    15845    17483
## 1.906257 2.869078 2.166035 2.002763 2.032363 1.914243 1.945529 1.842204
##    19928    20650    40920
## 2.177520 3.262746 1.610189
```

```
res.impca$completeObs[ selmiss,"engineSize"]<-3

res.impca$completeObs[ selmiss,"tax"]
```

```
##     7517     7522     7592    11290    11447    14582    15845    17483
## 145.0000 135.0000 145.0000 140.0000 147.1759 146.8105 147.2405 135.0000
##    19928    20650    40920
## 149.0245 156.9591 150.0000
```

```
res.impca$completeObs[ selmiss,"tax"]<-145

res.impca$completeObs[ selmiss,"age"]
```

```
##  7517  7522  7592 11290 11447 14582 15845 17483 19928 20650 40920
##     0     0     1     3     3     3     3     3     5     4     3
```

```
res.impca$completeObs[ selmiss,"age"]<-1

res.impca$completeObs[ selmiss,"mpg"]
```

```
##      7517      7522      7592     11290     11447     14582     15845     17483
##  40.90000 117.70000  39.20000  56.71214  55.08010  55.73208  52.61896  54.84479
##     19928     20650     40920
##  61.31262  25.40000  68.90000
```

```
res.impca$completeObs[ selmiss,"mpg"]<-50.4

res.impca$completeObs[ selmiss,"price"]
```

```
##  7517  7522  7592 11290 11447 14582 15845 17483 19928 20650 40920
## 29944 49790 33390 19998 19998 18500 21444 21494 12500 39948 16000
```

```
res.impca$completeObs[ selmiss,"price"]<-25650

res.impca$completeObs[ selmiss,"mileage"]
```

```
##  7517  7522  7592 11290 11447 14582 15845 17483 19928 20650 40920
##  1500  1500    45 41949 41146 36429 22063 16867 79830 49000 13593
```

```r
res.impca$completeObs[ selmiss,"mileage"]<-23750

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

6.2 Imputation of qualitative variables

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

```
##                model             year                  transmission
##  VW- Golf          : 481   2019    :1563   f.Trans-Manual    :1701
##  Mercedes- C Class: 376   2016    : 884   f.Trans-SemiAuto :1890
##  VW- Polo          : 319   2017    : 862   f.Trans-Automatic:1349
##  BMW- 3 Series     : 260   2018    : 498
##  Mercedes- A Class: 248   2015    : 361
##  Mercedes- E Class: 210   2020    : 306
##  (Other)           :3046   (Other): 466
##           fuelType       manufacturer
##  f.Fuel-Diesel:2799   Audi     :1031
##  f.Fuel-Petrol:2066   BMW      :1102
##  f.Fuel-Hybrid:  75   Mercedes:1307
##                       VW       :1500
##
##
##
```

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

```
##                model             year                  transmission
##  VW- Golf          : 481   2019    :1563   f.Trans-Manual    :1701
##  Mercedes- C Class: 376   2016    : 884   f.Trans-SemiAuto :1890
##  VW- Polo          : 319   2017    : 862   f.Trans-Automatic:1349
##  BMW- 3 Series     : 260   2018    : 498
##  Mercedes- A Class: 248   2015    : 361
##  Mercedes- E Class: 210   2020    : 306
##  (Other)           :3046   (Other): 466
##           fuelType       manufacturer
##  f.Fuel-Diesel:2799   Audi     :1031
##  f.Fuel-Petrol:2066   BMW      :1102
##  f.Fuel-Hybrid:  75   Mercedes:1307
##                       VW       :1500
##
##
##
```

```r
res.immca$completeObs[ selmiss,"model"]
```

```
##  [1] Audi- Q3   Audi- Q5   Audi- Q5   BMW- i3    BMW- i3    BMW- i3
##  [7] BMW- i3    BMW- i3    BMW- i3    BMW- X5    VW- Passat
## 91 Levels: Audi- A1 Audi- A3 Audi- A4 Audi- A5 Audi- A6 Audi- A7 ... VW- Up
```

```r
res.immca$completeObs[ selmiss,"transmission"]
```

```
##  [1] f.Trans-Manual    f.Trans-Automatic f.Trans-Automatic f.Trans-Automatic
##  [5] f.Trans-Automatic f.Trans-Automatic f.Trans-Automatic f.Trans-Automatic
##  [9] f.Trans-Automatic f.Trans-Automatic f.Trans-Manual
## Levels: f.Trans-Manual f.Trans-SemiAuto f.Trans-Automatic
```

```r
res.immca$completeObs[ selmiss,"fuelType"]
```

```
##  [1] f.Fuel-Petrol f.Fuel-Petrol f.Fuel-Diesel f.Fuel-Hybrid f.Fuel-Hybrid
##  [6] f.Fuel-Hybrid f.Fuel-Hybrid f.Fuel-Hybrid f.Fuel-Hybrid f.Fuel-Petrol
## [11] f.Fuel-Diesel
## Levels: f.Fuel-Diesel f.Fuel-Petrol f.Fuel-Hybrid
```

```r
res.immca$completeObs[ selmiss,"manufacturer"]
```

```
##  [1] Audi Audi Audi BMW  BMW  BMW  BMW  BMW  BMW  BMW  VW
## Levels: Audi BMW Mercedes VW
```

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

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

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

```r
library(mvoutlier)
```

```
## Loading required package: sgeostat
```

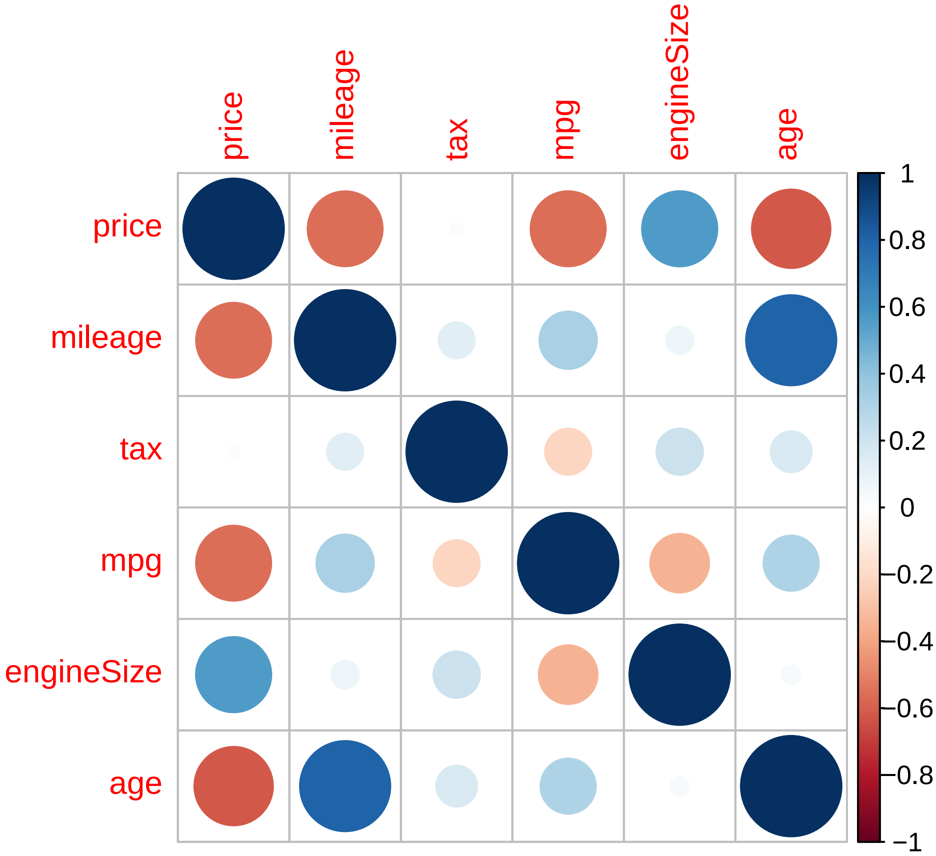
```r
library(FactoMineR)
vars_quantitatives<-names(df)[c(3,5,7:9,11)]
res <- cor(df[,vars_quantitatives])
round(res, 2)
```

```
##            price mileage   tax   mpg engineSize   age
## price       1.00   -0.56  0.02 -0.56       0.56 -0.61
## mileage    -0.56    1.00  0.13  0.33       0.08  0.81
## tax         0.02    0.13  1.00 -0.21       0.21  0.17
## mpg        -0.56    0.33 -0.21  1.00      -0.34  0.30
## engineSize  0.56    0.08  0.21 -0.34       1.00  0.03
## age        -0.61    0.81  0.17  0.30       0.03  1.00
```

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
corrplot(res)
```



6.3 Discretization

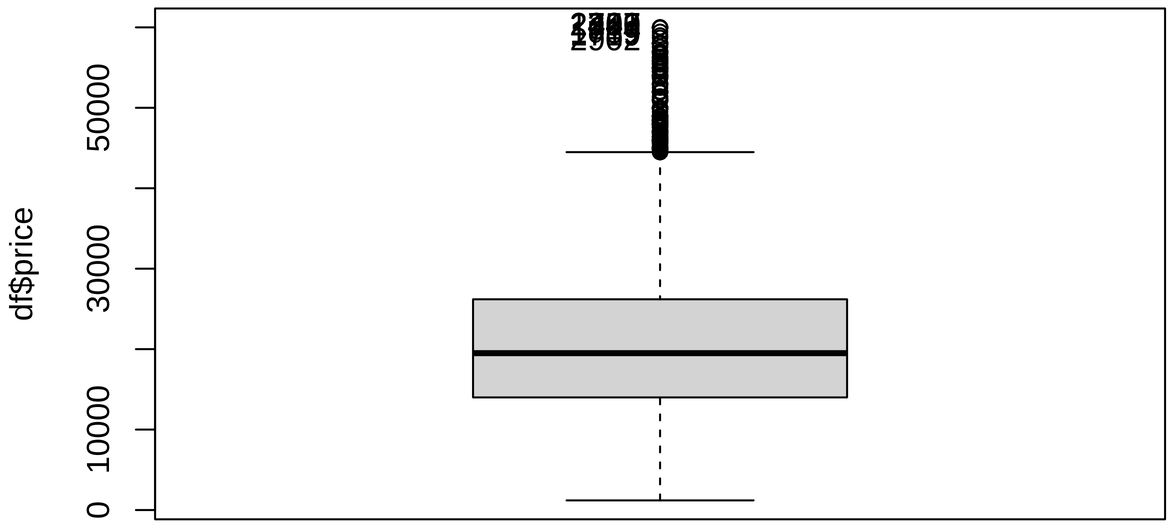
```
####    Discretization of all variables
## Check for missings, outliers and errors

####    Discretization of all numeric variables
vars_con
```

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

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1200   14000   19500   21177   26199   59995
```

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




```
## [1]  707 2327 1243 2429 2451 2750  614 1739 1719 2902
```

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

```
##    0%   25%   50%   75%  100%
##  1200 14000 19500 26199 59995
```

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

```
##      0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
##  1200.0 10325.2 12900.8 15290.0 17490.0 19500.0 21980.0 24890.1 27990.0 33789.1
##    100%
## 59995.0
```

```r
df$aux<-factor(cut(df$price,breaks=c(0,14500,20000,26000, 90000),include.lowest = T ))
summary(df$aux)
```

```
##     [0,1.45e+04] (1.45e+04,2e+04]   (2e+04,2.6e+04]   (2.6e+04,9e+04]
##             1331             1286              1081              1242
```

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

```
##     [0,1.45e+04] (1.45e+04,2e+04]   (2e+04,2.6e+04]   (2.6e+04,9e+04]
##            11250            17490             22990             31743
```

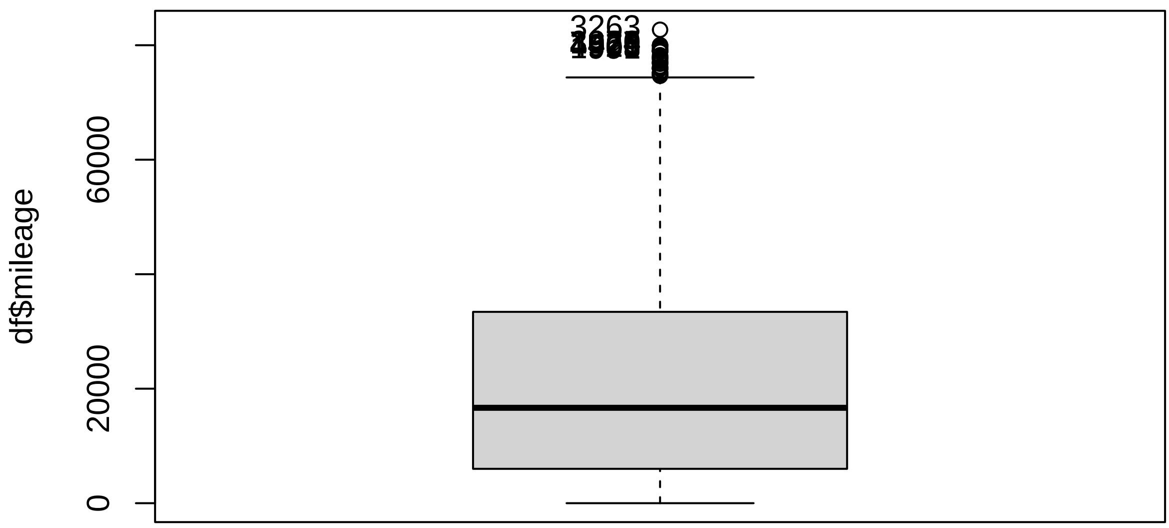
```r
df$f.price<-factor(cut(df$price/1000,breaks=c(0,15,20,26, 90),include.lowest = T ))
levels(df$f.price)<-paste("f.price-",levels(df$f.price),sep="")
table(df$f.price,useNA="always")
```

```
##
##  f.price-[0,15] f.price-(15,20] f.price-(20,26] f.price-(26,90]            <NA>
##            1457            1160            1081            1242               0
```

```r
##Discretization of mileage variable
summary(df$mileage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1    6000   16664   22025   33421   82702
```

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




```
##  [1] 3263 1970 1974 3325 1866 4456  975 4929 1908  991
```

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

```
##       0%      25%      50%      75%     100%
##     1.00  6000.00 16664.00 33420.75 82701.61
```

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

```
##       0%      10%      20%      30%      40%      50%      60%      70%
##     1.00  2094.40  4836.20  7536.90 11506.00 16664.00 23000.00 29924.50
##      80%      90%     100%
## 38426.40 51257.30 82701.61
```

```r
df$aux<-factor(cut(df$mileage,breaks=c(0,5750,17800,36000, 195000),include.lowest = T ))
summary(df$aux)
```

```
##        [0,5.75e+03] (5.75e+03,1.78e+04]  (1.78e+04,3.6e+04]  (3.6e+04,1.95e+05]
##                1185                1379                1265                1111
```

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

```
##        [0,5.75e+03] (5.75e+03,1.78e+04]  (1.78e+04,3.6e+04]  (3.6e+04,1.95e+05]
##                2640               10683               26108               49423
```

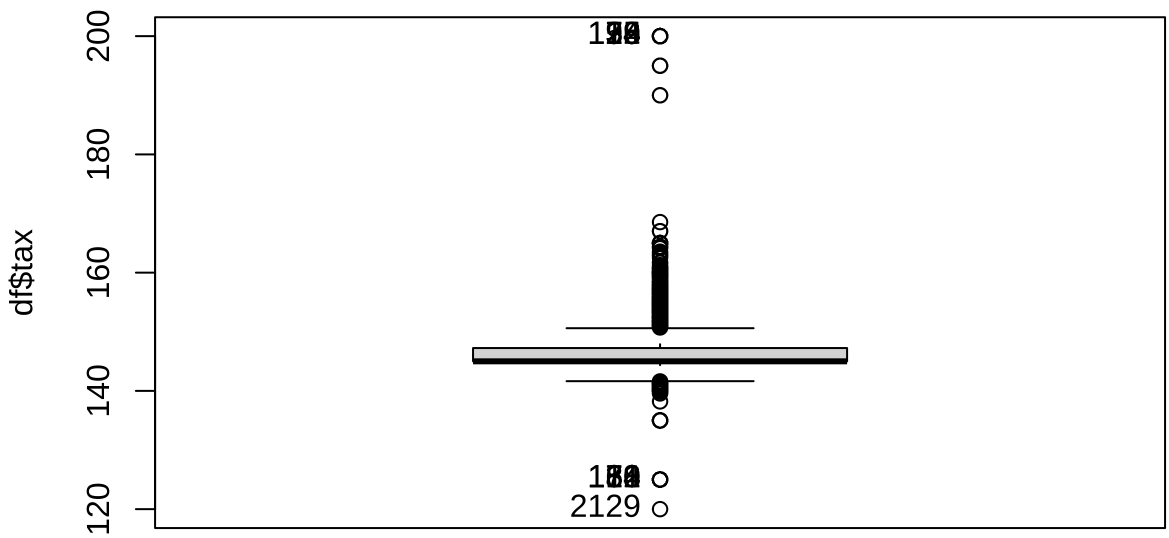
```r
df$f.miles<-factor(cut(df$mileage/1000,breaks=c(0,6,18,36, 195),include.lowest = T ))
levels(df$f.miles)<-paste("f.miles-",levels(df$f.miles),sep="")
table(df$f.miles,useNA="always")
```

```
##
##    f.miles-[0,6]   f.miles-(6,18]  f.miles-(18,36] f.miles-(36,195]
##             1267             1318             1244             1111
##             <NA>
##                0
```

```r
##Discretization of tax variable
summary(df$tax)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   120.0   145.0   145.0   146.8   147.2   200.0
```

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




```
##  [1] 2129   18   33   38   82  110  153  154  161  170    5   12   29   39   54
## [16]   76   98  115  124  138
```

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

```
##       0%      25%      50%      75%     100%
## 120.0000 145.0000 145.0000 147.2468 200.0000
```

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

```
##       0%      10%      20%      30%      40%      50%      60%      70%
## 120.0000 143.5212 145.0000 145.0000 145.0000 145.0000 145.0000 146.0002
##      80%      90%     100%
## 150.0000 150.7529 200.0000
```

```r
# df$aux<-factor(cut(df$tax,breaks=quantile(df$tax,seq(0,1,0.25),na.rm=TRUE),include.lowest = T )) # Doe
df$aux<-factor(cut(df$tax,breaks=c(0, 125, 145, 570),include.lowest = T ))
summary(df$aux)
```

```
##   [0,125] (125,145] (145,570]
##       279      2970      1691
```

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

```
##   [0,125] (125,145] (145,570]
##       125       145       150
```

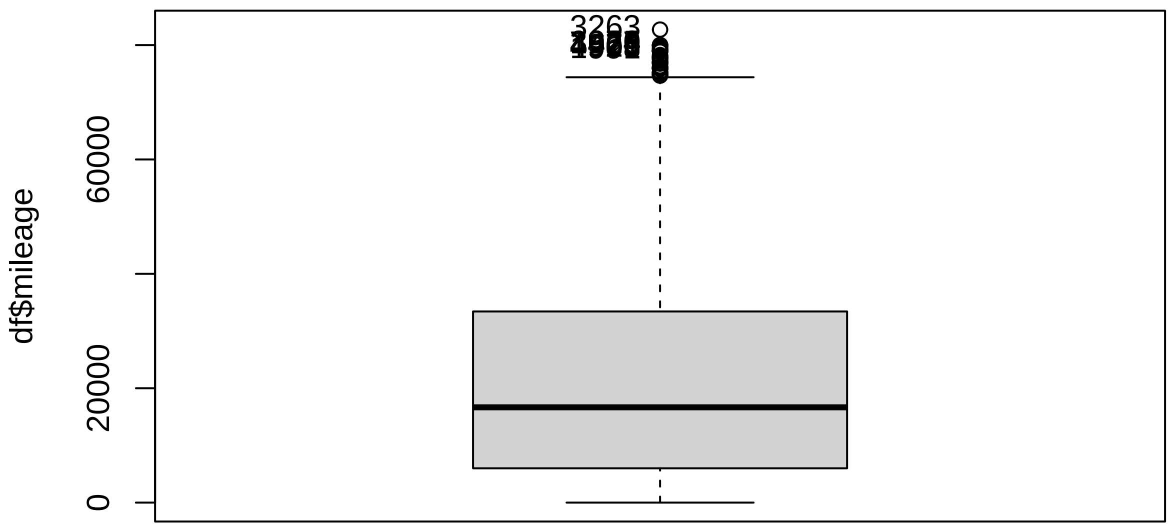
```r
df$f.tax<-factor(cut(df$tax,breaks=c(0, 125, 145, 570),include.lowest = T ))
levels(df$f.tax)<-paste("f.tax-",levels(df$f.tax),sep="")
table(df$f.tax,useNA="always")
```

```
##
##   f.tax-[0,125] f.tax-(125,145] f.tax-(145,570]            <NA>
##             279            2970            1691               0
```

```r
##Discretization of mileage variable
summary(df$mileage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1    6000   16664   22025   33421   82702
```

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




```
##  [1] 3263 1970 1974 3325 1866 4456  975 4929 1908  991
```

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

```
##       0%      25%      50%      75%     100%
##     1.00  6000.00 16664.00 33420.75 82701.61
```

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

```
##       0%      10%      20%      30%      40%      50%      60%      70%
##     1.00  2094.40  4836.20  7536.90 11506.00 16664.00 23000.00 29924.50
##      80%      90%     100%
## 38426.40 51257.30 82701.61
```

```r
df$aux<-factor(cut(df$mileage,breaks=c(0,6000,16500,33500, 153000),include.lowest = T ))
summary(df$aux)
```

```
##            [0,6e+03]    (6e+03,1.65e+04] (1.65e+04,3.35e+04] (3.35e+04,1.53e+05]
##                 1267                1189                1255                1229
```

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

```
##            [0,6e+03]    (6e+03,1.65e+04] (1.65e+04,3.35e+04] (3.35e+04,1.53e+05]
##                 2869               10503               24582               47288
```

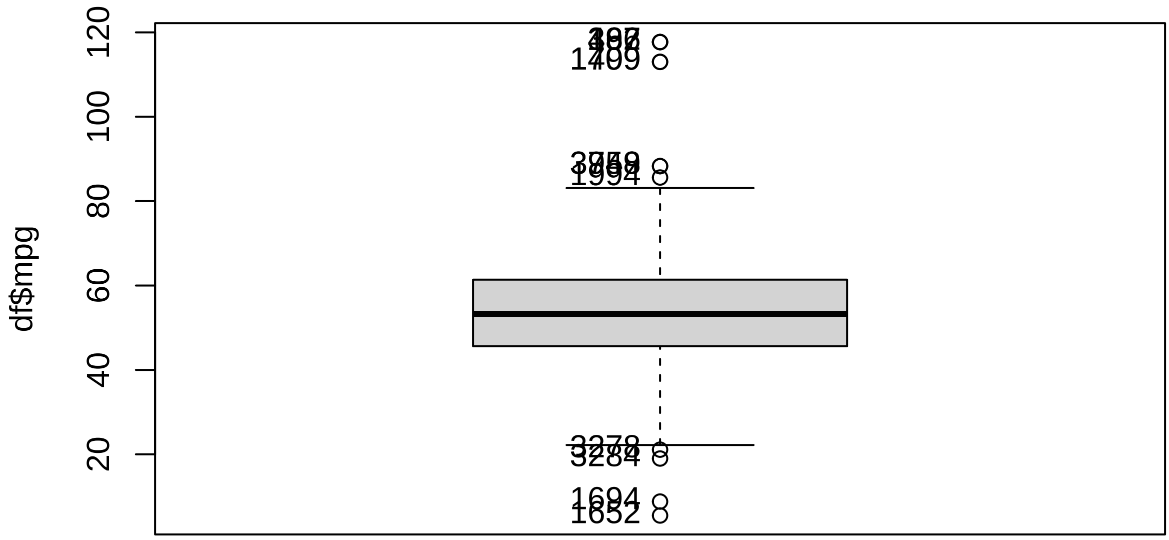
```r
df$f.mileage<-factor(cut(df$mileage/1000,breaks=c(0,6,17,34, 153),include.lowest = T ))
levels(df$f.mileage)<-paste("f.mileage-",levels(df$f.mileage),sep="")
table(df$f.mileage,useNA="always")
```

```
## 
##    f.mileage-[0,6]   f.mileage-(6,17]  f.mileage-(17,34] f.mileage-(34,153] 
##               1267               1238               1235               1200 
##               <NA> 
##                  0
```

```r
##Discretization of mpg variable
summary(df$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    5.50   45.60   53.30   53.01   61.40  117.70
```

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




```
##  [1] 1652 1694 3278 3284  182  366  497 1499 1709 1994 3748 3959
```

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

```
##    0%   25%   50%   75%  100%
##   5.5  45.6  53.3  61.4 117.7
```

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

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##   5.5  38.2  42.8  47.1  49.6  53.3  56.5  60.1  64.2  67.3 117.7
```

```r
df$aux<-factor(cut(df$mpg,breaks=c(5,45,53,62,470),include.lowest = T ))
summary(df$aux)
```

```
##   [5,45]  (45,53]  (53,62] (62,470]
##     1231     1226     1328     1155
```

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

```
##   [5,45]  (45,53]  (53,62] (62,470]
##     39.2     48.7     57.6     67.3
```

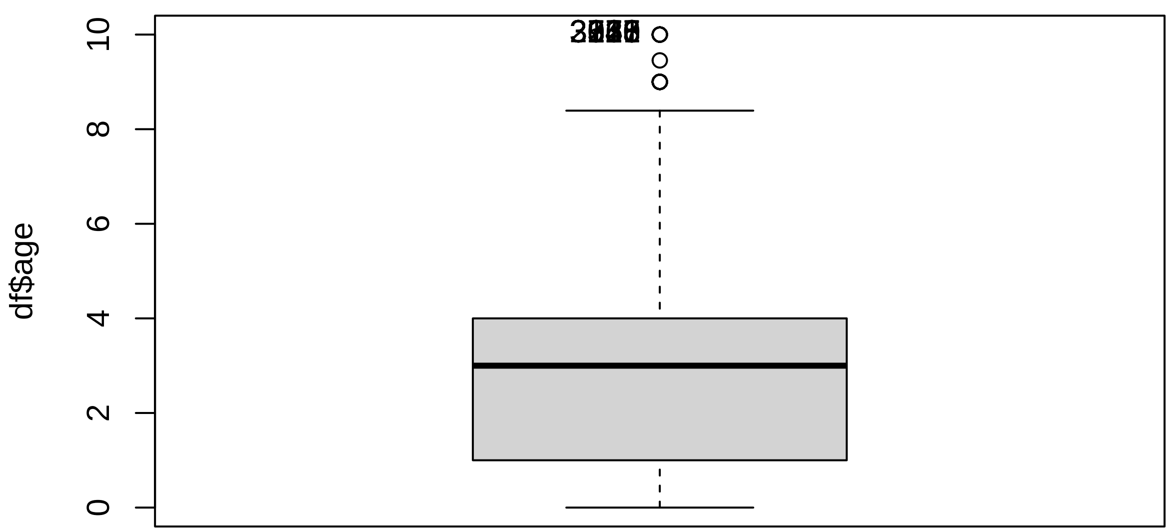
```r
df$f.mpg<-factor(cut(df$mpg,breaks=c(5,45,53,62,470),include.lowest = T ))
levels(df$f.mpg)<-paste("f.mpg-",levels(df$f.mpg),sep="")
table(df$f.mpg,useNA="always")
```

```
##
##   f.mpg-[5,45]  f.mpg-(45,53]  f.mpg-(53,62] f.mpg-(62,470]           <NA>
##           1231           1226           1328           1155              0
```

```r
##Discretization of age variable
summary(df$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   3.000   2.755   4.000  10.000
```

```r
Boxplot(df$age)
```




```
##  [1]  725  948 2127 3073 3136 3250 3261 3316 3323 3511
```

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

```
##   0%  25%  50%  75% 100%
##    0    1    3    4   10
```

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

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    0    1    1    1    2    3    3    4    4    5   10
```

```r
df$aux<-factor(cut(df$age,breaks=c(0,1,3,4,22),include.lowest = T ))
summary(df$aux)
```

```
##  [0,1]  (1,3]  (3,4] (4,22]
##   1877   1354    884    825
```

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

```
##  [0,1]  (1,3]  (3,4] (4,22]
##      1      3      4      6
```

```r
df$f.age<-factor(cut(df$age,breaks=c(0,1,3,4,22),include.lowest = T ))
levels(df$f.age)<-paste("f.age-",levels(df$f.age),sep="")
table(df$f.age,useNA="always")
```

```
##
##  f.age-[0,1]  f.age-(1,3]  f.age-(3,4] f.age-(4,22]         <NA>
##         1877         1354          884          825            0
```

7 Profiling

7.1 Numeric target: Age

We will now initiate the profiling process, which requires us to specify our numeric target (Age).

To examine the association between our numeric target and other variables, we will employ the 'condes' tool. This tool furnishes us with insights regarding the connections between the specified variables and the target.

```r
##############################################################################
#                                    Profiling
#                           Package FactoMineR will be used
##############################################################################

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   3.000   2.755   4.000  10.000
```

```r
# The "variable to describe cannot have NA ##################################
res.condes<-condes(df[,c(vars_res,vars_con,vars_dis)],1)

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

```
##            correlation p.value
## price.1      1.0000000       0
## engineSize   0.5603612       0
## mileage     -0.5565031       0
## mpg         -0.5579457       0
## age         -0.6100068       0
```

price engineSize *tax* mpg *mileage* age

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

```
##                     R2       p.value
## model        0.47692000  0.000000e+00
## year         0.39585964  0.000000e+00
## transmission 0.25934140 1.432790e-322
## manufacturer 0.10555834 4.915533e-119
## fuelType     0.01096686  1.506490e-12
```

model year *transmission* manufacturer *fueltype

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

```
##                                 Estimate       p.value
## year=2019                     16234.5838 1.723948e-242
## transmission=f.Trans-SemiAuto  4450.8868 7.016923e-149
```

```
## year=2020                        19354.9783  6.486056e-72
## year=2016                         4835.1690  1.262207e-63
## model=Mercedes- GLE Class        12834.5081  1.249374e-51
## manufacturer=Mercedes             2954.3930  3.934430e-45
## year=2014                         1081.3662  2.728294e-40
## year=2015                         3299.9761  9.559081e-40
## model=Mercedes- GLC Class         5482.4564  2.720559e-33
## model=BMW- X5                    12613.7807  1.993507e-32
## transmission=f.Trans-Automatic    2322.9136  1.146534e-24
## model=Audi- Q7                    9520.8618  4.477047e-23
## model=Audi- Q5                    1903.5511  1.729411e-16
## year=2017                         7518.3341  5.710926e-14
## model=VW- Touareg                 5389.9008  6.616181e-13
## model=Mercedes- GLS Class        14000.8564  5.162157e-11
## model=Audi- Q8                   25619.9564  6.596983e-11
## model=BMW- X3                     1136.5446  3.175917e-10
## model=BMW- M4                    14060.9008  4.217635e-10
## model=BMW- X6                     8749.8314  7.470139e-10
## model=Audi- RS6                  27535.7897  2.074465e-09
## manufacturer=BMW                  1209.1605  8.135226e-09
## fuelType=f.Fuel-Hybrid            3993.4225  2.390953e-07
## model=BMW- X4                     6094.8314  4.100843e-07
## model=BMW- 7 Series              10187.4564  4.369362e-07
## model=BMW- 8 Series              28552.9564  4.744568e-07
## model=VW- Caravelle              11040.2064  5.458986e-07
## model=Mercedes- V Class           5180.2897  6.532337e-07
## model=BMW- X2                     1523.4564  1.156963e-06
## model=Audi- A8                    4924.5740  2.294991e-06
## model=Mercedes- S Class           6523.7640  2.347011e-06
## model=BMW- i8                    24798.9564  7.049519e-06
## model=Audi- RS5                  23282.4564  1.933155e-05
## manufacturer=Audi                  784.4738  7.679165e-05
## model=VW- California             30548.4564  1.680008e-04
## model=Audi- SQ7                  17774.4564  5.109124e-04
## model=Audi- R8                   27407.4564  5.787508e-04
## model=Audi- RS4                  25052.4564  1.372163e-03
## model=BMW- Z4                     1523.8564  2.019913e-03
## model=Mercedes- X-CLASS           3232.2064  6.016629e-03
## model=BMW- M6                     8885.7897  7.319219e-03
## model=Audi- A7                     794.3025  9.213246e-03
## model=Audi- RS3                  11706.9564  9.398070e-03
## year=2018                        10812.8821  1.381872e-02
## model=Audi- S8                   17548.4564  1.497141e-02
## model=Mercedes- G Class          17547.4564  1.497564e-02
## model=BMW- M2                    17057.4564  1.717798e-02
## model=Mercedes- GLB Class         9752.4564  2.063383e-02
## model=VW- Tiguan Allspace          416.7897  4.039026e-02
## year=1998                        -9397.8910  4.998677e-02
## model=BMW- 4 Series              -4265.7226  4.435221e-02
## model=VW- Fox                   -26192.5436  4.177997e-02
## year=2002                        -1726.5576  4.154873e-02
## year=1999                       -10187.8910  4.126919e-02
## model=VW- Touran                 -9737.7436  3.523245e-02
## model=BMW- 5 Series              -4147.5709  2.172234e-02
## model=Mercedes- CLS Class        -1500.0436  1.680646e-02
## year=2001                        -4059.8910  1.423814e-02
## year=2005                        -7251.3910  1.380174e-02
## model=VW- Golf SV               -11021.3129  1.300045e-02
## model=VW- Arteon                 -1274.5036  1.058630e-02
## model=Audi- A5                   -3390.0186  8.064017e-03
## model=VW- Beetle                -16198.9881  2.307005e-03
## model=Mercedes- SL CLASS          -441.9365  1.589546e-03
## model=VW- Scirocco              -12669.5436  1.311639e-03
## model=VW- Passat                -10215.8651  1.901206e-04
```

```
## year=2006                         -7880.8910  5.333229e-05
## model=Mercedes- SLK              -17338.8513  4.384586e-05
## model=VW- CC                     -16491.7201  1.578326e-05
## model=Mercedes- A Class           -8879.5961  1.577850e-05
## fuelType=f.Fuel-Diesel            -1251.5903  7.964917e-06
## year=2007                         -6765.7481  7.434612e-06
## year=2009                         -4372.9910  4.580015e-06
## model=Mercedes- E Class           -2744.4389  9.627612e-08
## year=2010                         -5295.2756  2.540962e-08
## fuelType=f.Fuel-Petrol            -2741.8322  7.887810e-09
## year=2008                         -6177.2756  3.695575e-09
## model=Mercedes- C Class           -3359.5038  1.988785e-09
## year=2012                          -612.7697  8.539088e-10
## year=2011                         -3367.0338  6.324217e-10
## model=Audi- A3                   -10610.2686  1.370717e-10
## model=BMW- 1 Series              -11485.3192  2.207699e-14
## model=Audi- A1                   -14058.2876  1.432465e-19
## model=VW- Golf                   -11259.9886  1.956573e-32
## year=2013                          -603.7924  2.588555e-38
## model=VW- Up                     -19284.5725  1.520788e-43
## model=VW- Polo                   -16387.3023  2.600650e-84
## manufacturer=VW                   -4948.0272 9.119359e-114
## transmission=f.Trans-Manual       -6773.8003 1.937386e-313
```

7.2 Factor

```
#library(FactoMineR)
#summary(df$y.bin)
# The "variable to describe cannot have NA ###################################
#res.catdes<-catdes(df[,c(vars_res,vars_con,vars_dis)],2)

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


#res.catdes$quanti # Partial association of numeric variables to levels of outcome factor
#res.catdes$test.chi2 # Global association to factors
#res.catdes$category  # Partial association to significative levels in factors
```