UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Estadística
i Investigació Operativa

ASSIGNMENT: CAR PRICES

Projects form an important part of the education of software engineers. They form an active method of teaching, as defined by Piaget, leading to a "training in self-discipline and voluntary effort", which is important to software engineering professionals. Two purposes served by these projects are: education in professional practice, and outcome-based assessment.

Data cleaning or data scrubbing is one of the most important steps previous to any data decision-making or modelling process. Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

Data cleaning is the process that removes data that does not belong to the dataset or it is not useful for modelling purposes. Data transformation is the process of converting data from one format or structure into another format. Transformation processes can also be referred to as data wrangling, or data munging, transforming and mapping data from one "raw" data form into another format. **Essentially, real-world data is messy data and for model building: garbage data in is garbage analysis out**.

This practical assignment belongs to Data Science Master at the UPC, any dataset for modelling purposes should include a first methodological step on **data preparation** about:

- Removing duplicate or irrelevant observations
- Fix structural errors (usually coding errors, trailing blanks in labels, lower/upper case consistency, etc.).
- Check data types. Dates should be coded as such and factors should have level names (if possible, levels have to be set and clarify the variable they belong to). This point is sometimes included under data transformation process. New derived variables are to be produced sometimes scaling and/or normalization (range/shape changes to numeric variables) or category regrouping for factors (nominal/ordinal).
- Filter unwanted outliers. Univariate and multivariate outliers have to be highlighted. Remove register/erase values and set NA for univariate outiers.
- Handle missing data: figure out why the data is missing. Data imputation is to be considered when the aim is modelling (imputation has to be validated).
- Data validation is mixed of 'common sense and sector knowledge': Does the data make sense? Does the data follow the appropriate rules for its field? Does it prove or disprove the working theory, or bring any insight to light? Can you find trends in the data to help you form a new theory? If not, is that because of a data quality issue?

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 to be randomly selected from Mercedes, BMW, Volkwagen and Audi manufacturers. So, firstly you have to combine used car from the 4 manufacturers into 1 dataframe, adding a column manufacturer containing the vehicle brand.

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

This course project (Assignment x) is concerned with numeric model building for scraped data of used cars, which have been separated into files corresponding to each car manufacturer (only Mercedes, BMW, Volkswagen and Audi cars are to be considered): Y- Price (Numeric Target).

Aim is to predict how much you should sell your old car. It involves a numeric outcome. A random sample containing 5000 registers combining Audi, VW, Merc and BMW registers has to be retained by each group. Data from:
https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes

Some questions to address:

1. Determine if the response variable (price) has an acceptably normal distribution. Address test to discard serial correlation.
2. Indicate by exploration of the data which are apparently the variables most associated with the response variable (use only the indicated variables).
3. Define a polytomic factor f.age for the covariate car age according to its quartiles and argue if the average price depends on the level of age. Statistically justify the answer.
4. Calculate and interpret the anova model that explains car price according to the age factor and the fuel type.
5. Do you think that the variability of the price depends on both factors? Does the relation between price and age factor depend on fuel type?
6. Calculate the linear regression model that explains the price from the age: interpret the regression line and assess its quality.

7. What is the percentage of the price variability that is explained by the age of the car?
8. Do you think it is necessary to introduce a quadratic term in the equation that relates the price to its age?
9. Are there any additional explanatory numeric variables needed to the car price? Study collinearity effects.
10. After controlling by numerical variables, indicate whether the additive effect of the available factors on the price are statistically significant.
11. Select the best model available so far. Interpret the equations that relate the explanatory variables to the answer (price).
12. Study the model that relates the logarithm of the price to the numerical variables.
13. Once explanatory numerical variables are included in the model, are there any main effects from factors needed?
14. Graphically assess the best model obtained so far.
15. Assess the presence of outliers in the studentized residuals at a 99% confidence level. Indicate what those observations are.
16. Study the presence of *a priori* influential data observations, indicating their number according to the criteria studied in class.
17. Study the presence of *a posteriori* influential values, indicating the criteria studied in class and the actual atypical observations.
18. Given a 5-year old car, the rest of numerical variables on the mean and factors on the reference level, what would be the expected price with a 95% confidence interval?
19. Summarize what you have learned by working with this interesting real dataset.

Data Preparation outline:

Univariate Descriptive Analysis (to be included for each variable**):**

- Original numeric variables corresponding to qualitative concepts have to be converted to factors.
- 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.
- Exploratory Data Analysis for each variable (numeric summary and graphic support).

Data Quality Report:

Per variable, count:

- Number of missing values
- Number of errors (including inconsistencies)
- Number of outliers
- Rank variables according the sum of missing values (and errors).

Per individuals, count:

- number of missing values
- number of errors,
- number of outliers
- Identify individuals considered as multivariant outliers.

Create variable adding the total number missing values, outliers and errors. Describe these variables, to which other variables exist higher associations.

- Compute the correlation with all other variables. Rank these variables according the correlation
- Compute for every group of individuals (group of age, size of town, singles, married, …) the mean of missing/outliers/errors values. Rank the groups according the computed mean.

Imputation:

- Numeric Variables
- Factors

Profiling:

- Target (price)

CAR PRICES: An Example

Lídia Montero

2/12/2022

Table of Contents

```
rm(list=ls())
load("Own_Sample_raw.Rdata")
par(mfrow=c(1,1))
```

Data preperation

First, the data was imported and sampled by seed as asked to. The result is saved as “Own_sample_raw.Rdata” and imported.

Variable Analysis

On each variable of this data, descriptive analysis is performed, a data quality report made and imputation and profiling accounted for.

variable 1: model

Model is a nominal variable without missing values. However, it has a lot of levels (89) with a few very sparsly populated ones, such that converting it to a factor is not feasable.

```
summary(df$model)

##    Length     Class      Mode 
##      5000 character character 

table(df$model)

## 
##         1 Series         2 Series         3 Series         4 Series 
##              206              141              240               90 
##         5 Series         6 Series         7 Series         8 Series 
##              108               11               12                5 
##          A Class               A1               A3               A4 
##              277              151              208              137 
##               A5               A6               A7               A8 
##               62               86                7               14 
##           Amarok           Arteon          B Class           Beetle 
##               15               24               55                8 
##          C Class       Caddy Life  Caddy Maxi Life       California 
##              354                2               10                1 
##        Caravelle               CC         CL Class        CLA Class 
##               13               13               34                8 
##        CLS Class          E Class              Eos              Fox 
##               37              184                2                1 
##          G Class         GL Class        GLA Class        GLC Class 
##                3               15               84              102 
##        GLE Class        GLS Class             Golf          Golf SV 
##               42                9              497               21 
##               i3               i8            Jetta          M Class 
##                6                3                3                8 
##               M2               M3               M4               M5 
##                3                4               15                5 
##           Passat             Polo               Q2               Q3 
##               80              333               87              135 
##               Q5               Q7               Q8              RS3 
##               95               40                1                1 
##              RS4              RS5              RS6          S Class 
##                3                2                3               23 
##               S3               S4               S5               S8 
##                1                1                1                1 
##         Scirocco           Sharan          Shuttle         SL CLASS 
##               25               33                3               27 
##              SLK              SQ5              SQ7          T-Cross 
##               13                4                1               20 
##            T-Roc           Tiguan  Tiguan Allspace          Touareg 
##               70              180                9               39 
##           Touran               TT               Up          V Class 
##               40               34               79               26 
##          X-CLASS               X1               X2               X3 
##                7               74               29               56 
##               X4               X5               X6               X7 
##               25               40               12                5 
##               Z4 
##                6
```

```
sum(is.na(df$model))
```

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

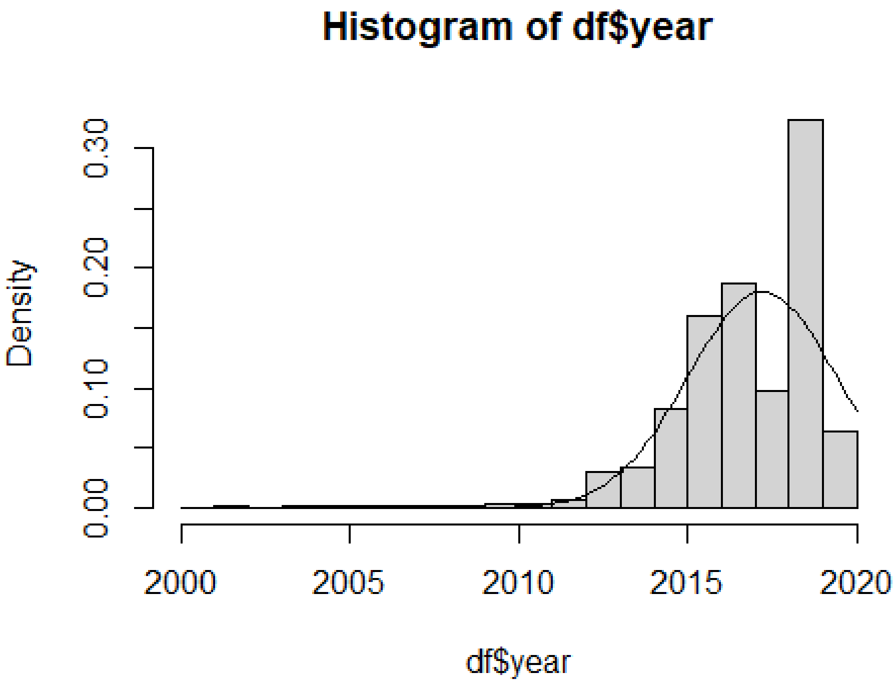
```
df$model[1:5]
```

```
## [1] " A3" " A3" " Q5" " A4" " Q3"
```

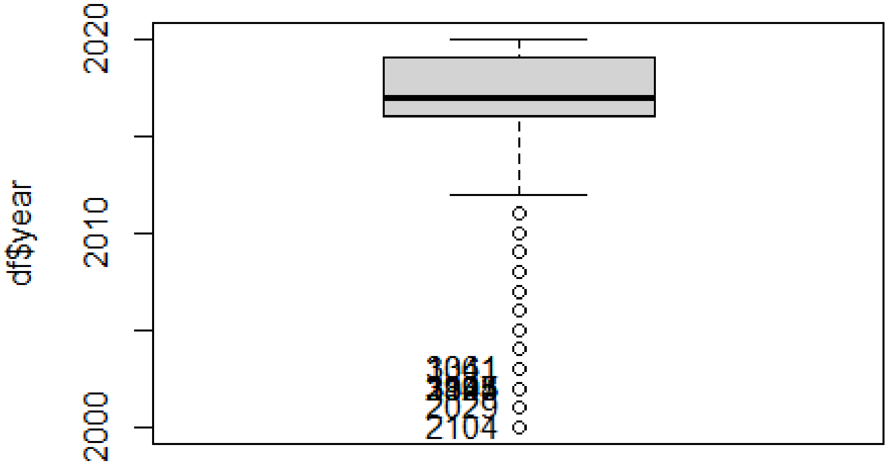
variable 2: year

This is a numeric interval variable. By using a histogram, it is clear that the data set contains mostly recent cars and is not normally distributed (confirmed by the shapiro test's low p-value). It contains no missing values thus imputation is not needed. The year variable contains 80 outliers (out of which 25 severe), all on the lower end of the spectrum. This is due to most of the data being from recently build cars. We create two additional variables: a numeric age variable ‘n.age’ and an age factor “f.age” as discretisation.

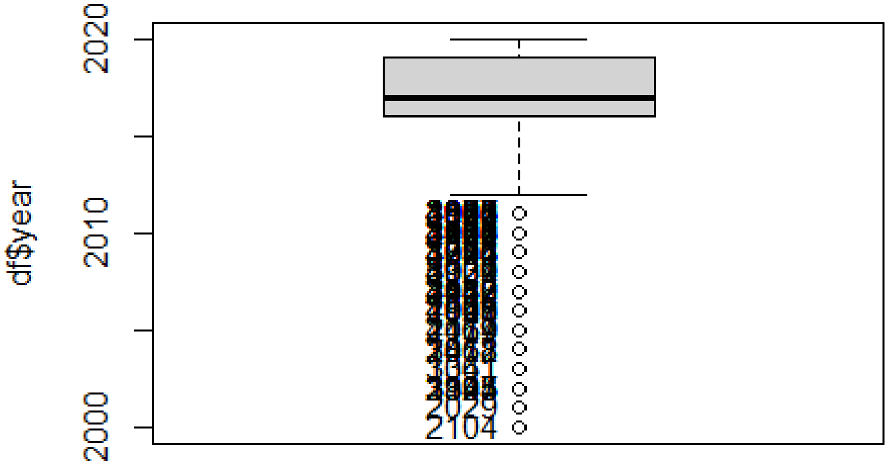
```
summary(df$year)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2000    2016    2017    2017    2019    2020

hist(df$year, breaks = 20, freq = F)
curve(dnorm(x, mean(df$year), sd(df$year)), add = T)
```



```
shapiro.test(df$year)

## 
##  Shapiro-Wilk normality test
## 
## data:  df$year
## W = 0.83902, p-value < 2.2e-16

sum(is.na(df$year))

## [1] 0

Boxplot(df$year)
```



```
##  [1] 2104 2029 1837 2105 3325 3343 3344 3995 1061 3341

length(Boxplot(df$year, id = list(n=Inf)))
```

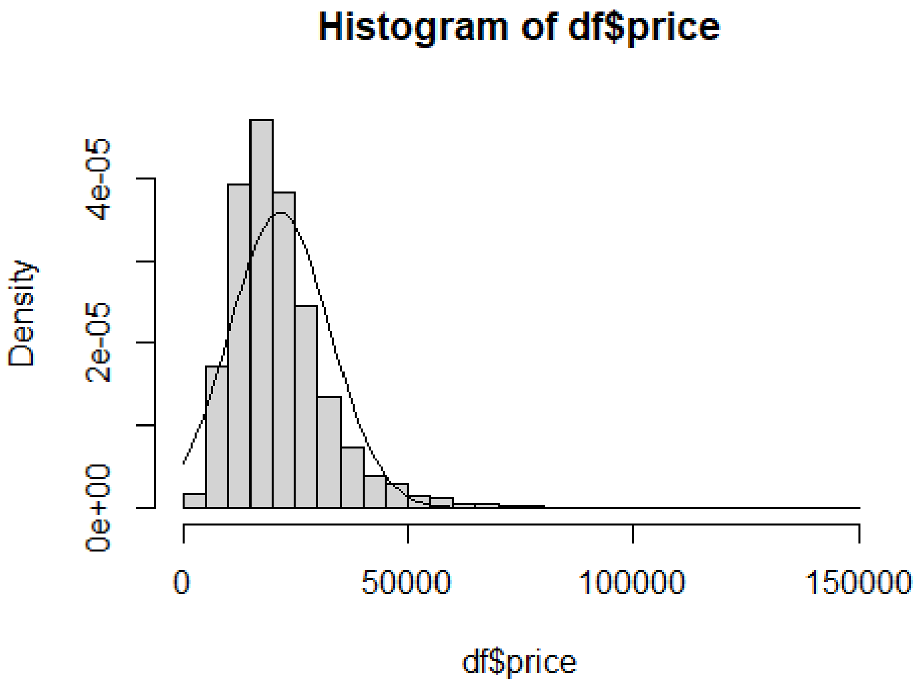


```
## [1] 80

sevout = (quantile(df$year,0.25)-(3*((quantile(df$year,0.75)-quantile(df$
year,0.25)))))
length(which(df$year < sevout))

## [1] 25

df$n.age = max(df$year)-(df$year)

df$f.age <- ifelse(df$n.age <= 1, 1, ifelse(df$n.age > 1 & df$n.age <= 3,
2, ifelse(df$n.age > 3 & df$n.age <= 4, 3, ifelse(df$n.age > 4, 4,0))))
df$f.age <- factor(df$f.age, labels=c("LowAge","LowMidAge","HighMidAge","
HighAge"), order = T, levels=c(1,2,3,4))
table(df$f.age)

##
##     LowAge  LowMidAge HighMidAge    HighAge
##       1938       1426        798        838
```

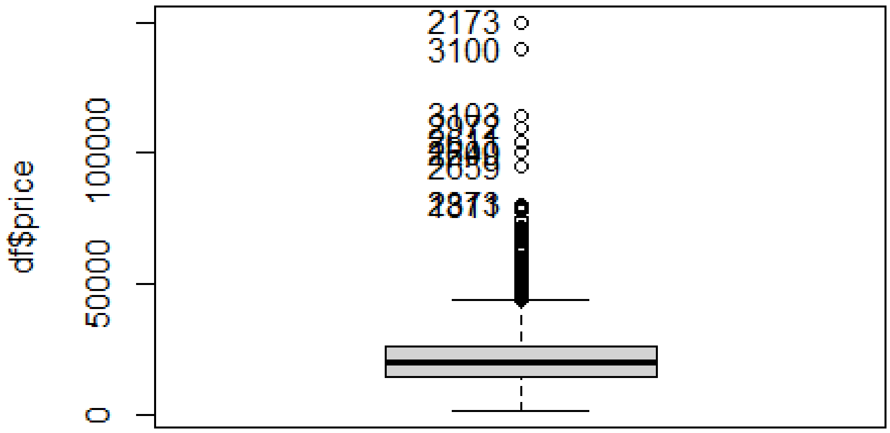
variable 3: price

This is a continuous ratio variable. The data is not normally distributed, but this fact is further answered in question 1 in the next section of this document. Again a histogram is used to visualize the data. It contains no missing values thus imputation is not needed. The price variable contains 207 outliers (out of which 108 severe), all on the higher end of the spectrum. We create an additional ordinal price factor “f.price” to create a discretisation according to the quartiles.

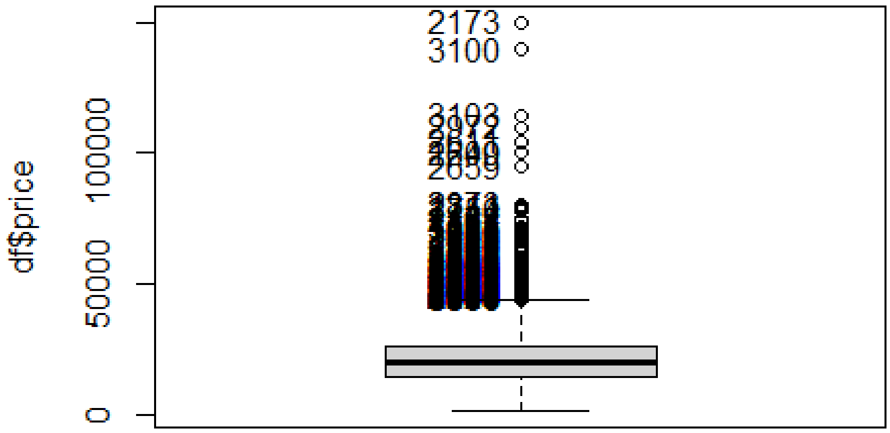
```
summary(df$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1275   14000   19799   21602   26000  149948

hist(df$price, breaks = 30, freq = F)
curve(dnorm(x, mean(df$price), sd(df$price)), add = T)
```



```
shapiro.test(df$price)

##
##  Shapiro-Wilk normality test
##
## data:  df$price
## W = 0.86882, p-value < 2.2e-16

sum(is.na(df$price))

## [1] 0

Boxplot(df$price)
```



```
##  [1] 2173 3100 3103 2972 2611 1609 2240 2659 2373 1811

length(Boxplot(df$price, id = list(n=Inf)))
```



```
## [1] 207
```

```
sevout_price = (quantile(df$price,0.25)+(3*((quantile(df$price,0.75)-quan
tile(df$price,0.25)))))
length(which(df$price > sevout_price))
```

```
## [1] 108
```

```
df$f.price <- ifelse(df$price <= 14000, 1, ifelse(df$price > 14000 & df$p
rice <= 19799, 2, ifelse(df$price > 19799 & df$price <= 26000, 3, ifelse(
df$price > 26000, 4,0))))
df$f.price <- factor(df$f.price, labels=c("LowPrice","LowMidPrice","HighM
idPrice","HighPrice"), order = T, levels=c(1,2,3,4))
table(df$f.price)
```

```
## 
##     LowPrice  LowMidPrice HighMidPrice    HighPrice 
##         1257         1244         1258         1241
```

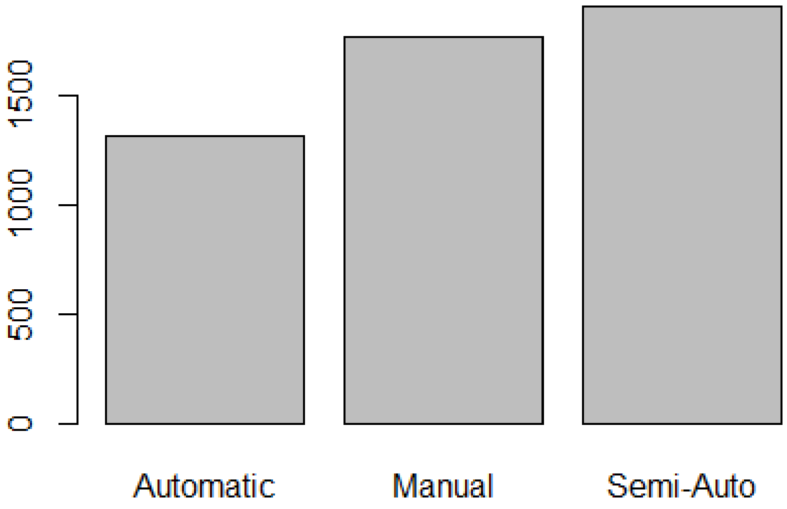
variable 4: transmission

This is a Nominal variable (with three levels) and is thus converted to the factor type. It is visualized by a bar plot, in which it is clear that all levels are well represented. Therefore, no outliers are present. The variable contains no missing values thus imputation is not needed.

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

```
##    Length     Class      Mode 
##      5000 character character
```

```
df$transmission = factor(df$transmission)
plot(df$transmission)
```

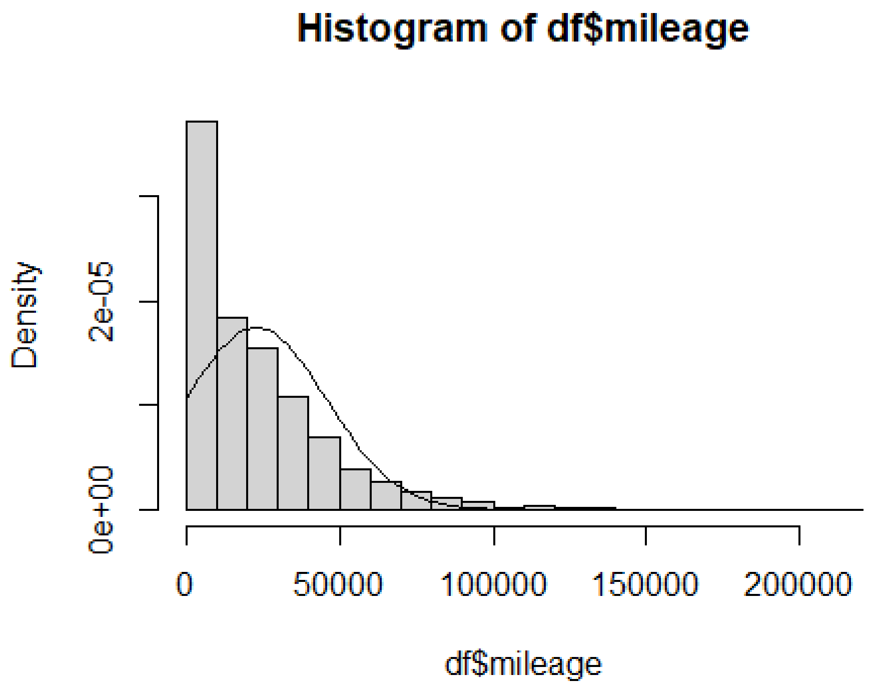


```
sum(is.na(df$transmission))

## [1] 0
```

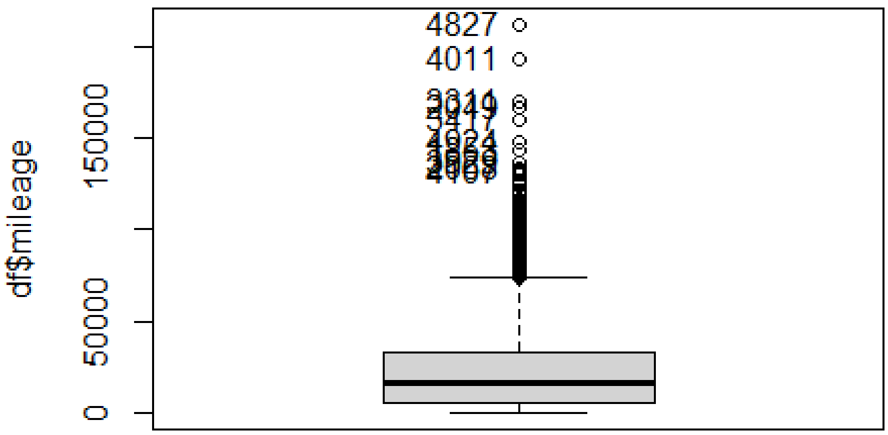
variable 5: mileage

This is a continuous ratio variable. The data does not look normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. Again a histogram is used to visualize the data. The variable contains no missing values thus imputation is not needed. It contains 188 outliers (out of which 98 severe), all on the higher end of the spectrum. We create an additional ordinal mileage factor “f.mileage” to create a discretisation according to the quartiles.

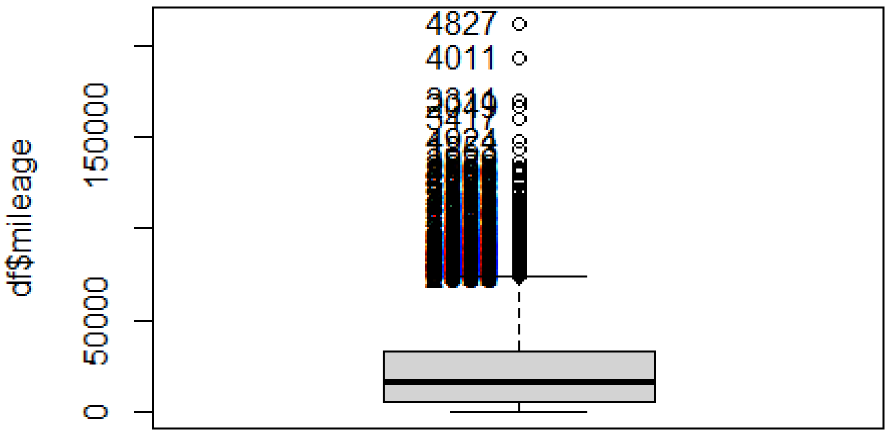
```
summary(df$mileage)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1    5728   16395   23042   33102  212000

hist(df$mileage, breaks = 30, freq = F)
curve(dnorm(x, mean(df$mileage), sd(df$mileage)), add = T)
```



```
shapiro.test(df$mileage)

##
##  Shapiro-Wilk normality test
##
## data:  df$mileage
## W = 0.83769, p-value < 2.2e-16

sum(is.na(df$mileage))

## [1] 0

Boxplot(df$mileage)
```



```
##  [1] 4827 4011 3311 2049 3417 4924 1853 3929 2068 4107

length(Boxplot(df$mileage, id = list(n=Inf)))
```

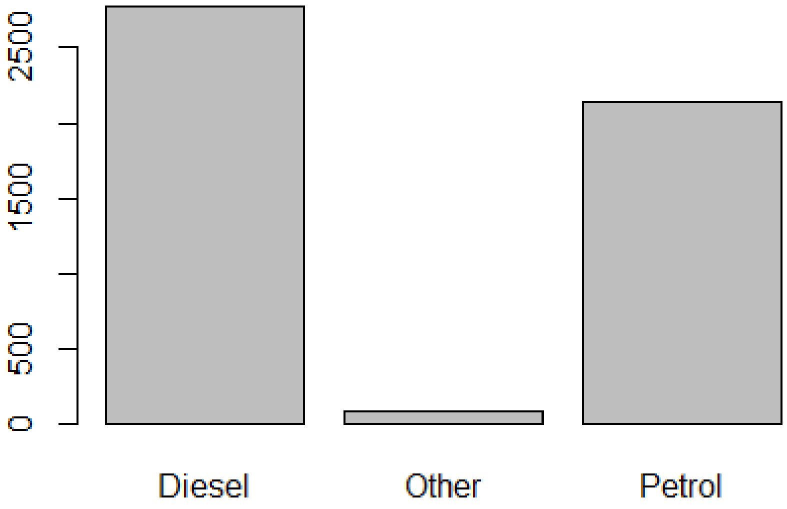


```
## [1] 188

sevout_mileage = (quantile(df$mileage,0.25)+(3*((quantile(df$mileage,0.75
)-quantile(df$mileage,0.25)))))
length(which(df$mileage > sevout_mileage))

## [1] 98

df$f.mileage <- ifelse(df$mileage <= 5728, 1, ifelse(df$mileage > 5728 &
df$mileage <= 16395, 2, ifelse(df$mileage > 16395 & df$mileage <= 33102,
3, ifelse(df$mileage > 33102, 4,0))))
df$f.mileage <- factor(df$f.mileage,labels=c("LowMileage","LowMidMileage"
,"HighMidMileage","HighMileage"), order = T, levels=c(1,2,3,4))
table(df$f.mileage)

## 
##     LowMileage  LowMidMileage HighMidMileage    HighMileage 
##           1250           1250           1250           1250
```

variable 6: fuelType

This is a nominal variable with 5 levels in which ‘electric’, ‘hybrid’ and ‘other’ only combine for less then 2% of the instances combined, such that these are all collapsed into the ‘other’ level. The variable contains no missing values thus imputation is not needed. A bar plot is used to plot the variable.

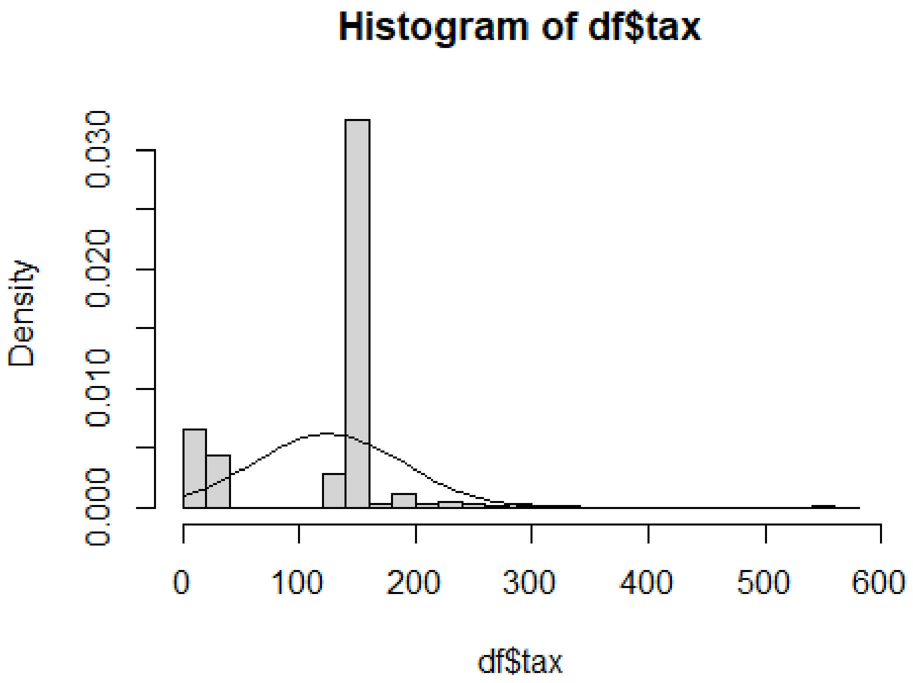
```
summary(df$fuelType)

##    Length     Class      Mode 
##      5000 character character

elec_idx <- which(df$fuelType == 'Electric')
prop.table(table(df$fuelType))

## 
##   Diesel Electric   Hybrid    Other   Petrol 
##   0.5548   0.0002   0.0122   0.0036   0.4292

df$fuelType[which(df$fuelType == 'Hybrid')] = 'Other'
df$fuelType[which(df$fuelType == 'Electric')] = 'Other'
df$fuelType = factor(df$fuelType)

plot(df$fuelType)
```



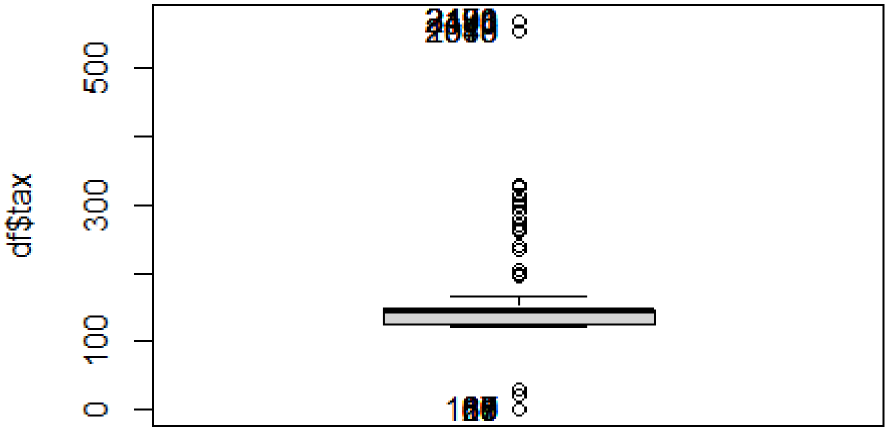
variable 7: tax

This is a continuous ratio variable. The data does not look normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. Again a histogram is used to visualize the data. The variable contains no missing values thus imputation is not needed. It contains 1422 outliers (out of which all severe), on both sides of the spectrum. We create an additional ordinal tax factor “f.tax” to create a discretisation according to the quartiles.

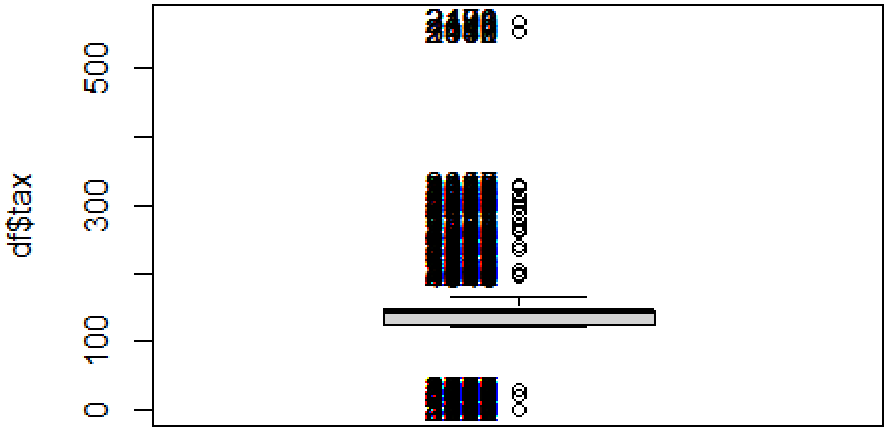
```
summary(df$tax)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.0   125.0   145.0   124.9   145.0   570.0

hist(df$tax, breaks = 30, freq = F)
curve(dnorm(x, mean(df$tax), sd(df$tax)), add = T)
```



```
shapiro.test(df$tax)

##
##  Shapiro-Wilk normality test
##
## data:  df$tax
## W = 0.72815, p-value < 2.2e-16

sum(is.na(df$tax))

## [1] 0

Boxplot(df$tax)
```



```
## [1]    7    9   17   21   66   67   85   88   89  136 2173 2180 2191
2198 3430
## [16] 3431  979 1543 2010 2088

length(Boxplot(df$tax, id = list(n=Inf)))
```

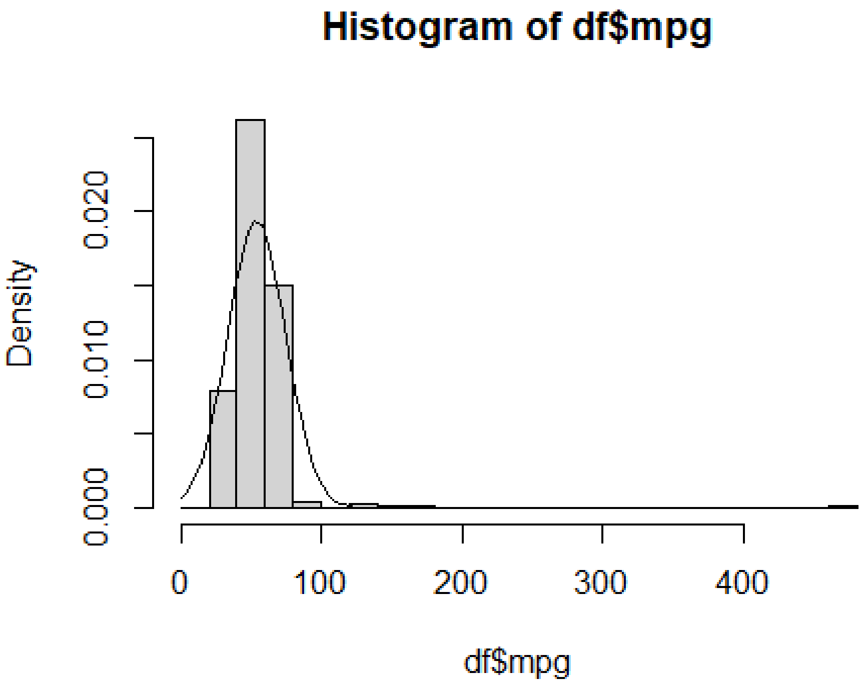


```
## [1] 1422

sevout_tax_upp = (quantile(df$tax,0.25)+(3*((quantile(df$tax,0.75)-quanti
le(df$tax,0.25)))))
sevout_tax_low = (quantile(df$tax,0.25)-(3*((quantile(df$tax,0.75)-quanti
le(df$tax,0.25)))))
length(which(df$tax > sevout_tax_upp))+length(which(df$tax < sevout_tax_l
ow))

## [1] 1422

df$f.tax <- ifelse(df$tax <= 125, 1, ifelse(df$tax > 125 & df$tax < 145,
2, ifelse(df$tax == 145, 3, ifelse(df$tax > 145, 4,0))))
df$f.tax <- factor(df$f.tax,labels=c("Lowtax","LowMidtax","HighMidtax","H
ightax"), order = T, levels=c(1,2,3,4))
table(df$f.tax)

##
##     Lowtax  LowMidtax HighMidtax    Hightax
##       1349         34       2610       1007
```

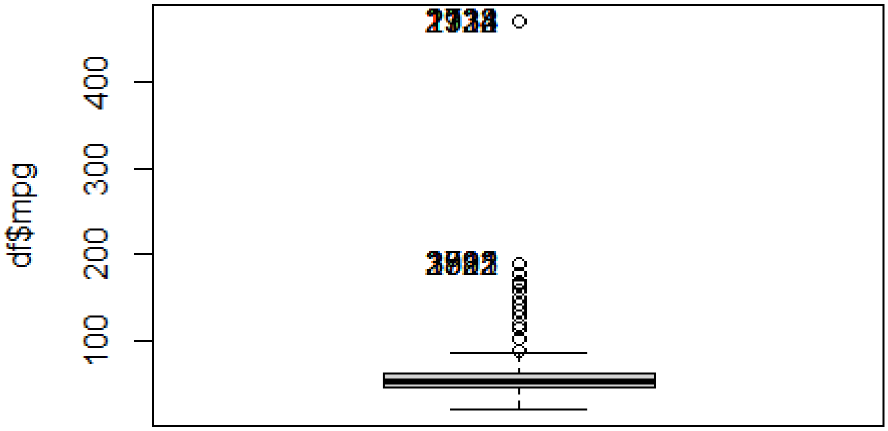
variable 8: mpg

This is a continuous ratio variable. The data does not look normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. Again a histogram is used to visualize the data. The variable contains no missing values thus imputation is not needed. It contains 56 outliers (out of which 53 severe), all on the high side of the spectrum. We create an additional ordinal mpg factor “f.mpg” to create a discretisation according to the quartiles.

```
summary(df$mpg)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.50   44.10   52.80   53.96   61.40  470.80

hist(df$mpg, breaks = 30, freq = F)
curve(dnorm(x, mean(df$mpg), sd(df$mpg)), add = T)
```

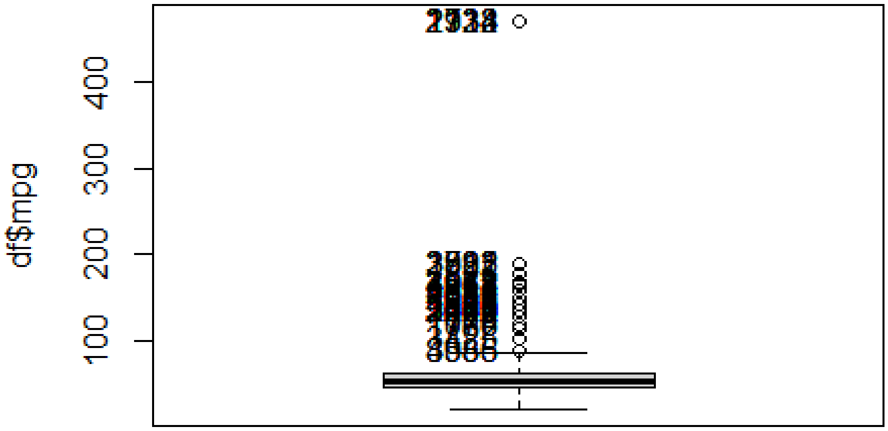


```
shapiro.test(df$mpg)

##
##  Shapiro-Wilk normality test
##
## data:  df$mpg
## W = 0.52394, p-value < 2.2e-16

sum(is.na(df$mpg))

## [1] 0

Boxplot(df$mpg)
```



```
##  [1] 1133 1324 1732 1934 2114 2138 1582 2995 3722 3811
```

```
length(Boxplot(df$mpg, id = list(n=Inf)))
```

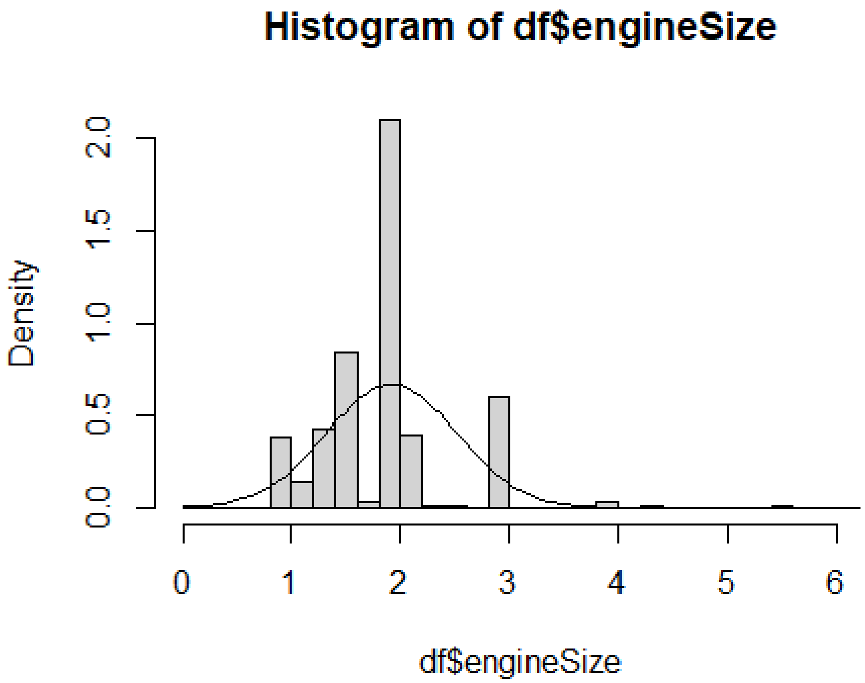


```
## [1] 56

sevout_mpg = (quantile(df$mpg,0.25)+(3*((quantile(df$mpg,0.75)-quantile(d
f$mpg,0.25)))))
length(which(df$mpg > sevout_mpg))

## [1] 53

df$f.mpg <- ifelse(df$mpg <= 44.1, 1, ifelse(df$mpg > 44.1 & df$mpg <= 52
.8, 2, ifelse(df$mpg > 52.8 & df$mpg <= 61.4, 3, ifelse(df$mpg > 61.4, 4,
0))))
df$f.mpg <- factor(df$f.mpg,labels=c("Lowmpg","LowMidmpg","HighMidmpg","H
ighmpg"), order = T, levels=c(1,2,3,4))
table(df$f.mpg)

##
##     Lowmpg  LowMidmpg HighMidmpg    Highmpg
##       1257       1243       1342       1158
```

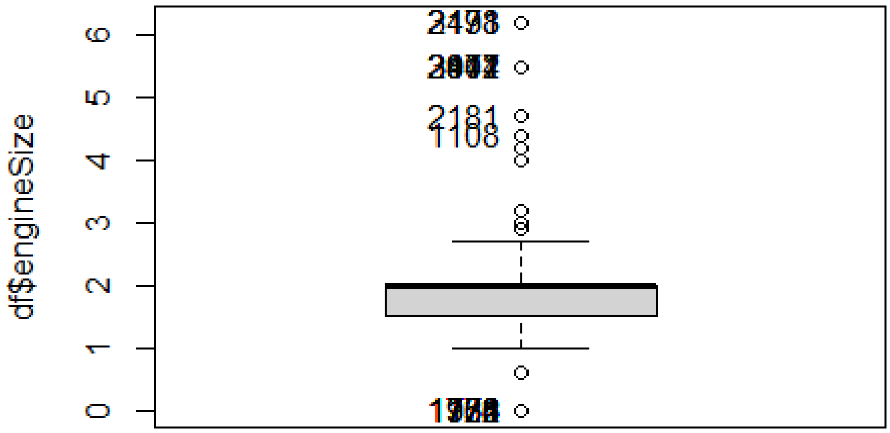
variable 9: engineSize

This is an interval variable. It contains 673 outliers, out of which 55 severe. There are 15 instances of cars without enginesize which seems like missing values. As it was stated in the data description, these instance could denote electric fuelTypes, however, after inspecting each case more closely it appeared that only 1 of these truely denotes an electric engine and the others are missing values at random (MAR). As the hybrid and electric fuel types were previously set to other, all engine sizes which have values equal to 0 are set to NA and then imputed using the MICE algorithm (an algorithm using chained equations using k-Nearest-Neighbour and regression techniques), except for the electric fuel type. We create an additional ordinal enginesize factor “f.engineSize” to create a discretisation according to the quartiles.

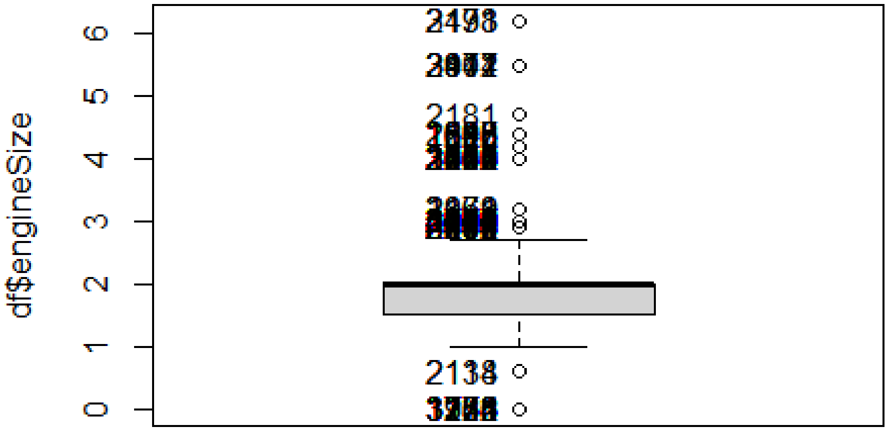
```
summary(df$engineSize)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.500   2.000   1.918   2.000   6.200

hist(df$engineSize, breaks = 30, freq = F)
curve(dnorm(x, mean(df$engineSize), sd(df$engineSize)), add = T)
```



```
shapiro.test(df$engineSize)

##
##  Shapiro-Wilk normality test
##
## data:  df$engineSize
## W = 0.86113, p-value < 2.2e-16

sum(is.na(df$engineSize))

## [1] 0

Boxplot(df$engineSize)
```



```
## [1]  758  768  771  772  773  776 1133 1324 1732 1934 2173 2198 3431
2812 2944
## [16] 3077 3111 3401 2181 1108

length(Boxplot(df$engineSize, id = list(n=Inf)))
```



```
## [1] 673
```

```
sevout_engineSize_upp = (quantile(df$engineSize,0.25)+(3*((quantile(df$en
gineSize,0.75)-quantile(df$engineSize,0.25)))))
sevout_engineSize_down = (quantile(df$engineSize,0.25)-(3*((quantile(df$e
ngineSize,0.75)-quantile(df$engineSize,0.25)))))
length(which(df$engineSize > sevout_engineSize_upp | df$engineSize < sevo
ut_engineSize_down))
```

```
## [1] 55
```

```
df$engineSize[which(df$engineSize == 0)] = NA
df$engineSize[elec_idx] = 0
require(mice)
```

```
## Loading required package: mice
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
options(contrasts = c("contr.treatment", "contr.treatment")) ##set option
s to contr.treatment as mice can only use lm when options is set to this.
imputation = mice(df, method = 'pmm')
```

```
##
##  iter imp variable
##   1   1  engineSize*
##   1   2  engineSize*
##   1   3  engineSize*
##   1   4  engineSize*
##   1   5  engineSize*
##   2   1  engineSize*
##   2   2  engineSize*
##   2   3  engineSize*
##   2   4  engineSize*
##   2   5  engineSize*
##   3   1  engineSize*
##   3   2  engineSize*
##   3   3  engineSize*
##   3   4  engineSize*
##   3   5  engineSize*
##   4   1  engineSize*
##   4   2  engineSize*
##   4   3  engineSize*
##   4   4  engineSize*
##   4   5  engineSize*
##   5   1  engineSize*
##   5   2  engineSize*
##   5   3  engineSize*
##   5   4  engineSize*
##   5   5  engineSize*
```

```
## Warning: Number of logged events: 28
```

```
imputation$imp$engineSize
```

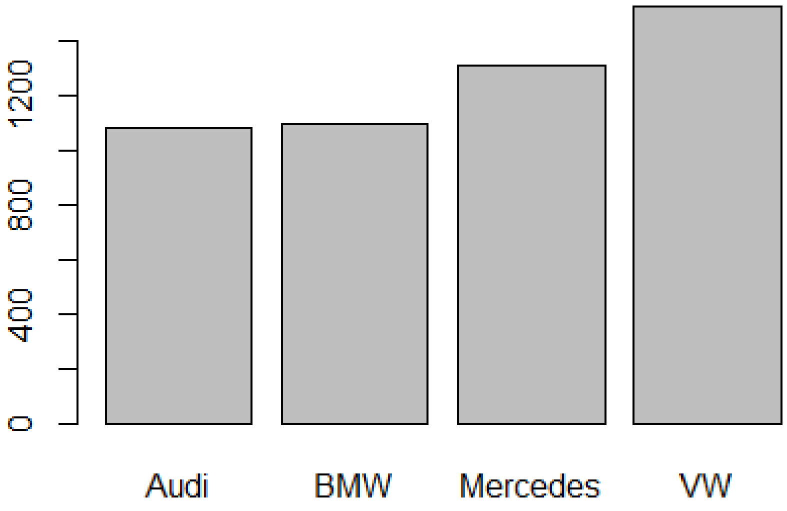
```
##         1   2   3   4   5
## 7519  1.5 2.1 2.0 2.0 2.0
## 7599  1.0 1.0 1.4 1.0 1.4
## 7632  2.0 1.5 1.5 1.5 1.5
## 7645  3.0 3.0 3.0 3.0 3.0
## 7660  2.1 2.0 2.1 2.0 2.0
## 7701  2.0 1.5 1.6 2.0 1.6
## 11290 0.6 1.0 1.0 1.0 1.0
## 13021 0.6 0.0 0.6 1.0 0.0
## 16949 0.0 1.0 0.6 0.6 0.0
## 31068 2.1 2.0 2.0 2.0 3.0
## 31069 1.0 1.0 2.0 1.6 1.6
## 31071 2.0 2.1 2.0 2.0 2.0
## 32125 2.0 2.0 2.0 2.0 2.1
## 32189 2.0 2.0 1.5 1.5 2.0
```

```
df = complete(imputation, 5)
summary(df$engineSize)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.500   2.000   1.922   2.000   6.200

df$f.engineSize <- ifelse(df$engineSize <= 1.5, 1, ifelse(df$engineSize >
1.5 & df$engineSize < 2, 2, ifelse(df$engineSize >= 2 & df$engineSize <=
2, 3, ifelse(df$engineSize > 2, 4,0))))
df$f.engineSize <- factor(df$f.engineSize,labels=c("LowengineSize","LowMi
dengineSize","HighMidengineSize","HighengineSize"), order = T, levels=c(1
,2,3,4))
table(df$f.engineSize)

##
##     LowengineSize  LowMidengineSize HighMidengineSize    HighengineSiz
e
##              1494               333              2105              106
8
```

variable 10: manufacturer

This is a Nominal variable (with four levels) and is thus converted to the factor type. It is visualized by a bar plot, in which it is clear that all levels are well represented. Therefore, no outliers are present. The variable contains no missing values thus imputation is not needed.

```
table(df$manufacturer)

##
##     Audi      BMW Mercedes       VW
##     1075     1096     1308     1521

df$manufacturer = factor(df$manufacturer)
plot(df$manufacturer)
```



```
sum(is.na(df$manufacturer))

## [1] 0
```

Data quality report

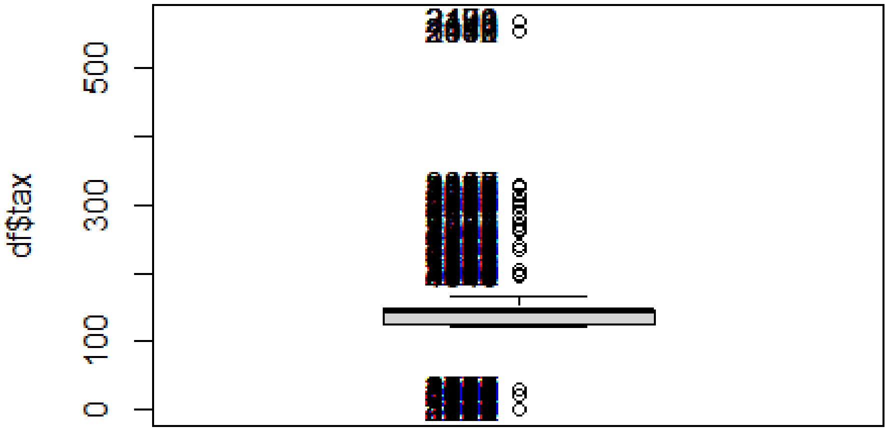
Variables

Now that all the variables have been explored, their general quality can be reported on. In terms of missingness, only the engineSize shows missing data and hence, it ranks last in terms of missingness. If we rank the numeric variables in terms of number of outliers, the following list is obtained, starting with the most outliers: Tax, engine size, price, mileage, year and mpg. Combining these two results, it seems like both tax and engine size are the variables containing the most noise in this data set.

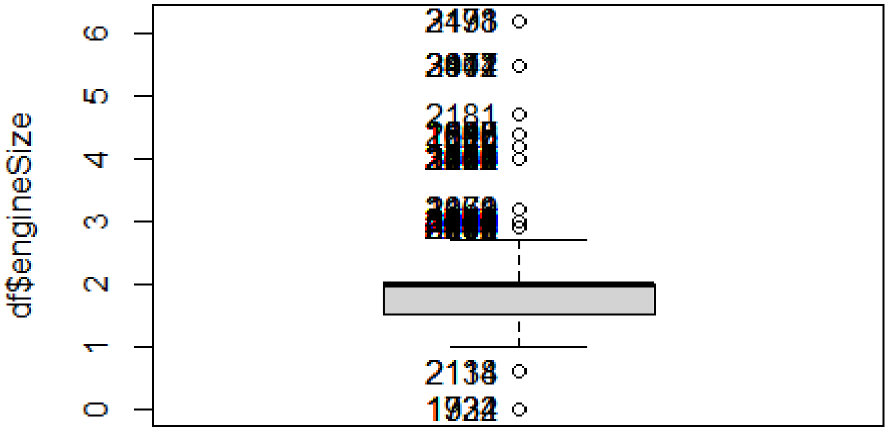
individuals

Now the individuals are investigated. First the number of univariate outliers per individual are counted and added in a new variable called 'univ_outl_count'. Looking at the 8 individuals with the most univariate outliers (4) it can be concluded that they are all old, highly taxed, low mpg, high engine size and most with a low price and high mileage. A correlation matrix confirms this as it shows a significant negative correlation to the year (so the older the car, the more univ outliers) and a significant positive correlation to both mileage and engine size.

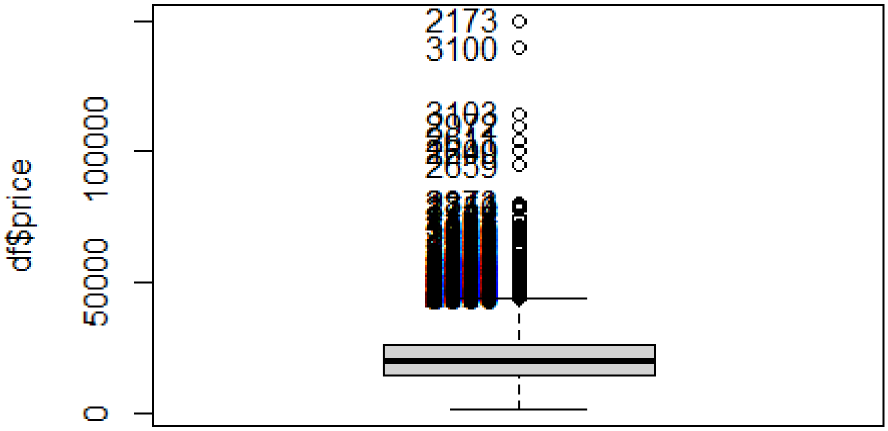
```
df$univ_outl_count <- 0
df$univ_outl_count[Boxplot(df$tax, id = list(n=Inf))] = df$univ_outl_coun
t[Boxplot(df$tax, id = list(n=Inf))] + 1
```



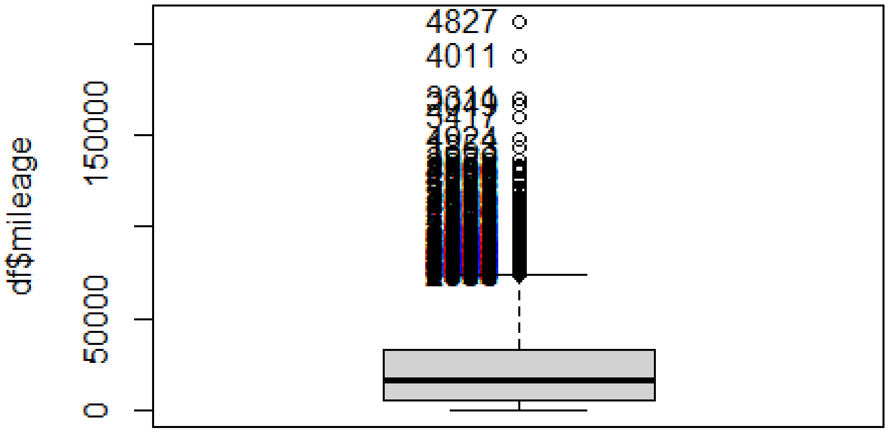
```
df$univ_outl_count[Boxplot(df$engineSize, id = list(n=Inf))] = df$univ_ou
tl_count[Boxplot(df$engineSize, id = list(n=Inf))] + 1
```



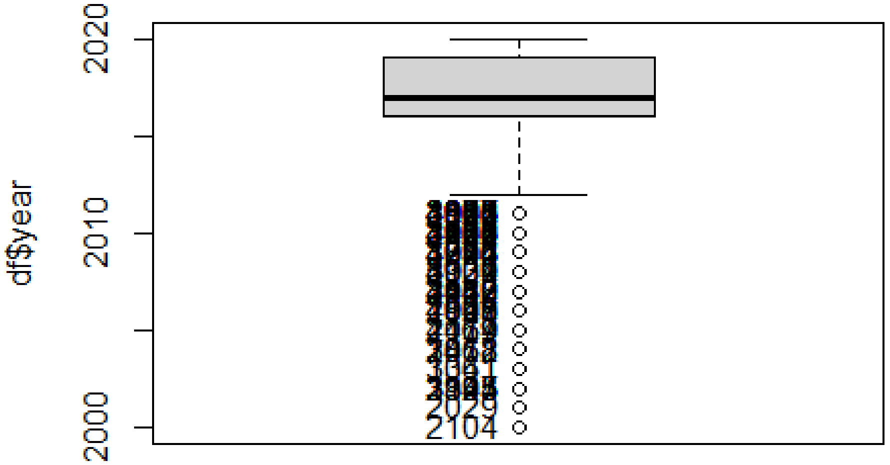
```r
df$univ_outl_count[Boxplot(df$price, id = list(n=Inf))] = df$univ_outl_count[Boxplot(df$price, id = list(n=Inf))] + 1
```



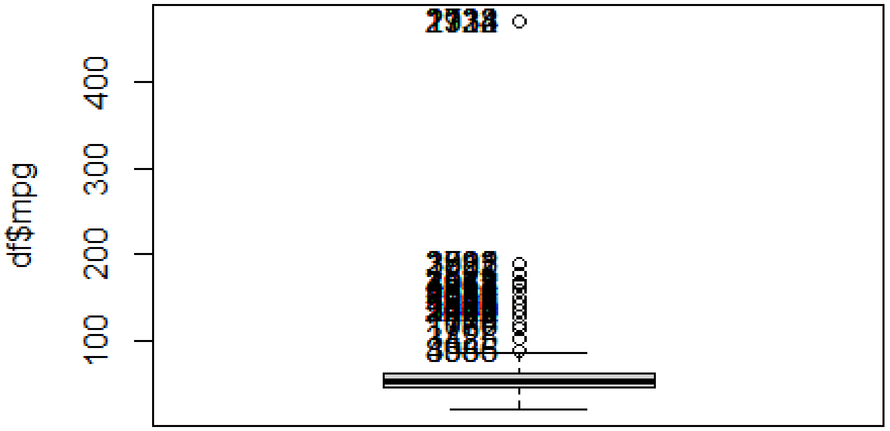
```
df$univ_outl_count[Boxplot(df$mileage, id = list(n=Inf))] = df$univ_outl_
count[Boxplot(df$mileage, id = list(n=Inf))] + 1
```



```
df$univ_outl_count[Boxplot(df$year, id = list(n=Inf))] = df$univ_outl_cou
nt[Boxplot(df$year, id = list(n=Inf))] + 1
```

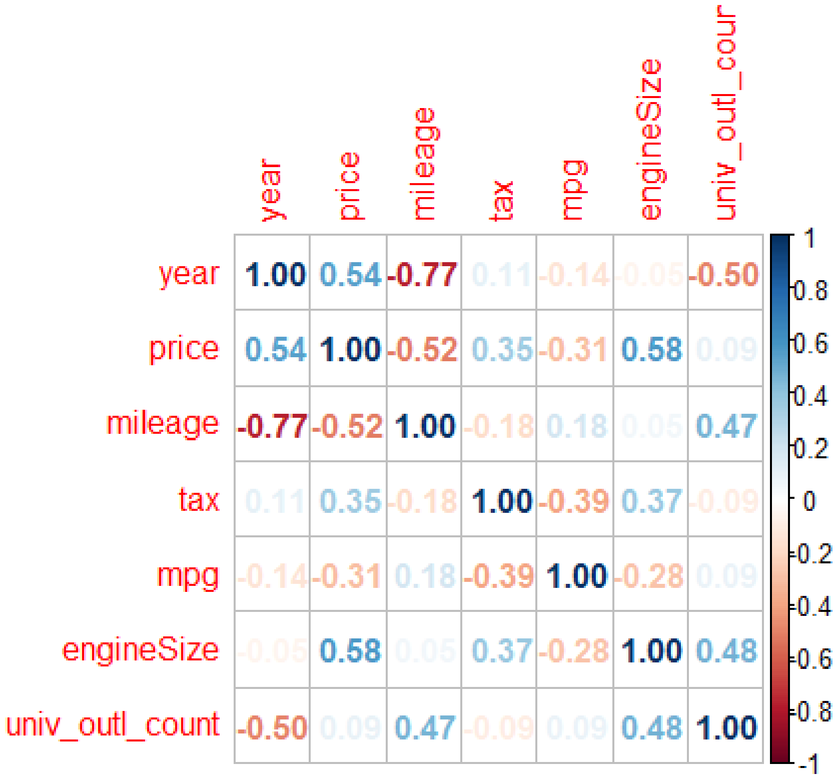


```
df$univ_outl_count[Boxplot(df$mpg, id = list(n=Inf))] = df$univ_outl_coun
t[Boxplot(df$mpg, id = list(n=Inf))] + 1
```



```
max(df$univ_outl_count)
```

```
## [1] 4

df[which(df$univ_outl_count == 4),]

##          model year  price transmission mileage fuelType tax  mpg engi
neSize
## 1014        A4 2005   4990       Manual   87990   Diesel 325 36.7
3.0
## 1023        A8 2006   5595    Automatic  104000   Diesel 325 33.6
3.0
## 2010  6 Series 2006   4999    Automatic  126054   Petrol 555 29.7
3.0
## 2029  3 Series 2001   3050    Automatic   90000   Petrol 325 27.7
3.0
## 2130  3 Series 2008   8790       Manual   85000   Petrol 555 28.5
3.0
## 2169        M3 2005  10999       Manual  115000   Petrol 315 20.8
3.2
## 2173  SL CLASS 2011 149948    Automatic    3000   Petrol 570 21.4
6.2
## 3296   E Class 2009   3995    Automatic  131711   Diesel 300 27.7
3.0
##      manufacturer n.age   f.age   f.price   f.mileage   f.tax  f.mpg
## 1014         Audi    15 HighAge  LowPrice HighMileage Hightax Lowmpg
## 1023         Audi    14 HighAge  LowPrice HighMileage Hightax Lowmpg
## 2010          BMW    14 HighAge  LowPrice HighMileage Hightax Lowmpg
## 2029          BMW    19 HighAge  LowPrice HighMileage Hightax Lowmpg
## 2130          BMW    12 HighAge  LowPrice HighMileage Hightax Lowmpg
## 2169          BMW    15 HighAge  LowPrice HighMileage Hightax Lowmpg
## 2173     Mercedes     9 HighAge HighPrice  LowMileage Hightax Lowmpg
## 3296     Mercedes    11 HighAge  LowPrice HighMileage Hightax Lowmpg
##        f.engineSize univ_outl_count
## 1014 HighengineSize               4
## 1023 HighengineSize               4
## 2010 HighengineSize               4
## 2029 HighengineSize               4
## 2130 HighengineSize               4
## 2169 HighengineSize               4
## 2173 HighengineSize               4
## 3296 HighengineSize               4

df_of_interest = df[,c(2,3,5,7,8,9,18)]
cor_outl = cor(df_of_interest)
require(corrplot)

## Loading required package: corrplot

## corrplot 0.92 loaded

par(mfrow=c(1,1))
corrplot(cor_outl, method = 'number')
```

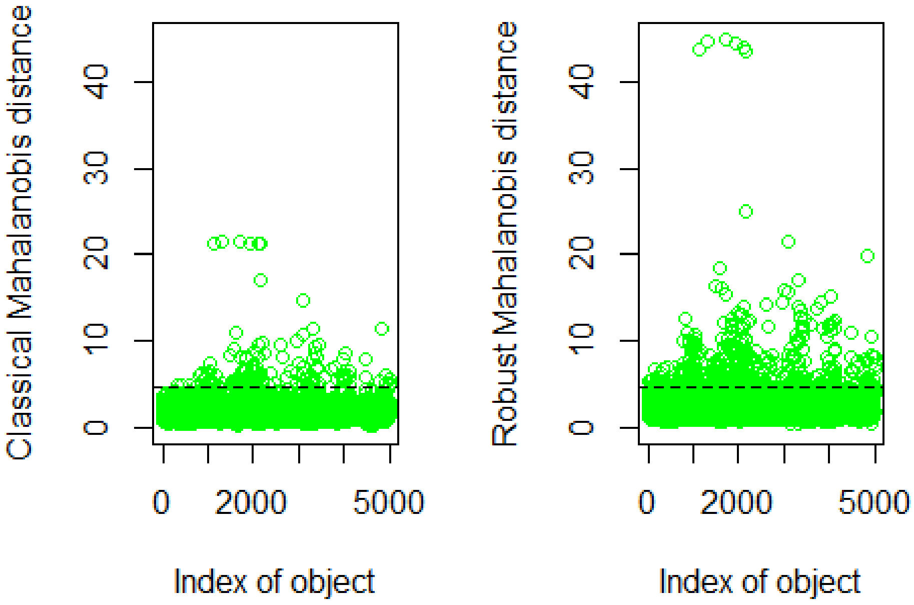


	year	price	mileage	tax	mpg	engineSize	univ_outl_cour
year	1.00	0.54	-0.77	0.11	-0.14	-0.05	-0.50
price	0.54	1.00	-0.52	0.35	-0.31	0.58	0.09
mileage	-0.77	-0.52	1.00	-0.18	0.18	0.05	0.47
tax	0.11	0.35	-0.18	1.00	-0.39	0.37	-0.09
mpg	-0.14	-0.31	0.18	-0.39	1.00	-0.28	0.09
engineSize	-0.05	0.58	0.05	0.37	-0.28	1.00	0.48
univ_outl_count	-0.50	0.09	0.47	-0.09	0.09	0.48	1.00

Multivariate Outliers

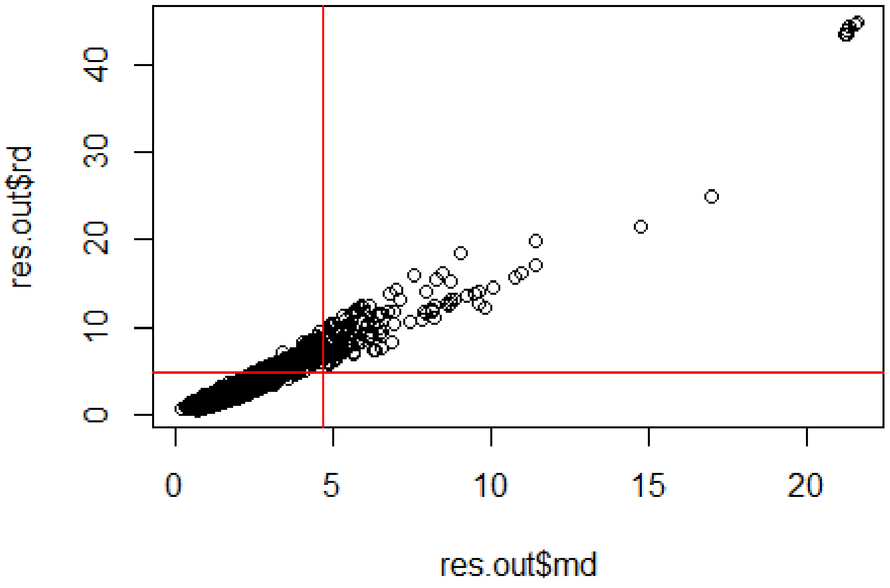
Moutlier is applied on the numerical variables to find multivariate outliers. With the tax variable included, however, the method returns a singular matrix. Therefore this variable is excluded from the calculation. A very mild threshold of 0.05 % is chosen as signficance level because is already returns a significant amount of outliers; This makes up around 3% of the total amount of instances. It is chosen to delete these outliers from the data set for the rest of the project.

```
require(chemometrics)

## Loading required package: chemometrics

## Loading required package: rpart

res.out = Moutlier(df[,c(2,3,5,8,9)], quantile = 0.9995, col="green")
```



```
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

##   [1]  306  409  603  740  776  825  848  888  928  969  997 1000 1004
1014 1023
##  [16] 1038 1058 1061 1108 1133 1154 1217 1299 1324 1508 1523 1543 1554
1582 1609
##  [31] 1638 1642 1648 1687 1697 1714 1732 1741 1750 1806 1811 1835 1837
1839 1845
##  [46] 1847 1850 1853 1861 1881 1892 1906 1934 1939 1954 1956 1981 2010
2029 2048
##  [61] 2049 2058 2068 2078 2088 2104 2105 2114 2130 2138 2167 2169 2172
2173 2180
##  [76] 2181 2182 2198 2240 2352 2373 2467 2611 2659 2744 2812 2944 2952
2972 2995
##  [91] 3077 3087 3100 3103 3111 3148 3265 3284 3296 3311 3325 3328 3329
3332 3341
## [106] 3343 3344 3401 3412 3417 3425 3430 3431 3578 3722 3811 3831 3929
3971 3982
## [121] 3995 4011 4046 4048 4064 4100 4101 4107 4109 4110 4113 4462 4479
4794 4827
## [136] 4922 4924 4929 4995 4998 5000

length(which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))
)/5000

## [1] 0.0282
```

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



```
summary(df[which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cuto
ff)),])

##     model                year          price          transmission
##  Length:141         Min.   :2000   Min.   :  1275   Automatic:74
##  Class :character   1st Qu.:2008   1st Qu.:  5895   Manual   :38
##  Mode  :character   Median :2016   Median : 18950   Semi-Auto:29
##                     Mean   :2013   Mean   : 30259
##                     3rd Qu.:2018   3rd Qu.: 52999
##                     Max.   :2020   Max.   :149948
##     mileage         fuelType       tax              mpg             engin
eSize
##  Min.   :    16   Diesel:46   Min.   :  0.0   Min.   : 19.50   Min.   
:0.000
##  1st Qu.:  8420   Other :34   1st Qu.:135.0   1st Qu.: 30.00   1st Qu.
:2.000
##  Median : 42932   Petrol:61   Median :145.0   Median : 37.70   Median 
:2.000
##  Mean   : 58535               Mean   :194.3   Mean   : 80.42   Mean   
:2.578
##  3rd Qu.:110860               3rd Qu.:300.0   3rd Qu.: 80.70   3rd Qu.
:3.000
```

```
##  Max.   :212000              Max.   :570.0   Max.   :470.80   Max.   
:6.200  
##   manufacturer     n.age             f.age            f.price   
##  Audi    :18    Min.   : 0.00   LowAge    :31   LowPrice    :64  
##  BMW     :54    1st Qu.: 2.00   LowMidAge :27   LowMidPrice :11  
##  Mercedes:41    Median : 4.00   HighMidAge:15   HighMidPrice:15  
##  VW      :28    Mean   : 6.56   HighAge   :68   HighPrice   :51  
##                 3rd Qu.:12.00                                    
##                 Max.   :20.00                                    
##           f.mileage          f.tax            f.mpg               f.engi
neSize  
##  LowMileage    :30   Lowtax    :30   Lowmpg    :76   LowengineSize    
:26  
##  LowMidMileage :19   LowMidtax :23   LowMidmpg : 9   LowMidengineSize 
: 9  
##  HighMidMileage:16   HighMidtax:33   HighMidmpg:10   HighMidengineSize
:40  
##  HighMileage   :76   Hightax   :55   Highmpg   :46   HighengineSize   
:66  
##                                                                         
##                                                                         
##  univ_outl_count
##  Min.   :1.000  
##  1st Qu.:2.000  
##  Median :2.000  
##  Mean   :2.199  
##  3rd Qu.:3.000  
##  Max.   :4.000
```

```
summary(df)
```

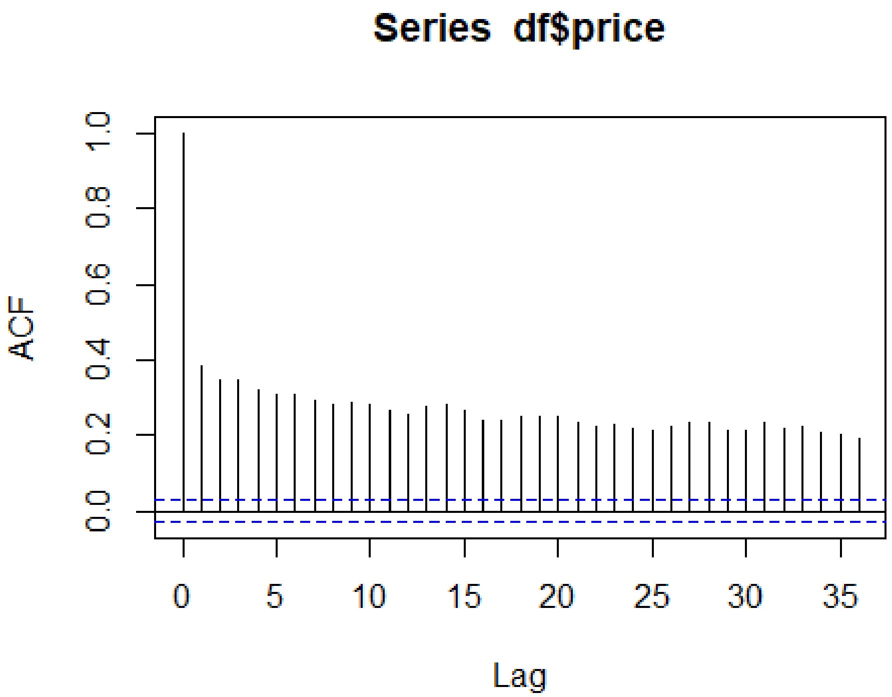
```
##     model                year          price          transmission 
##  Length:5000        Min.   :2000   Min.   :  1275   Automatic:1317  
##  Class :character   1st Qu.:2016   1st Qu.: 14000   Manual   :1775  
##  Mode  :character   Median :2017   Median : 19799   Semi-Auto:1908  
##                     Mean   :2017   Mean   : 21602                   
##                     3rd Qu.:2019   3rd Qu.: 26000                   
##                     Max.   :2020   Max.   :149948                   
##     mileage         fuelType         tax              mpg        
##  Min.   :     1   Diesel:2774   Min.   :  0.0   Min.   : 19.50  
##  1st Qu.:  5728   Other :  80   1st Qu.:125.0   1st Qu.: 44.10  
##  Median : 16395   Petrol:2146   Median :145.0   Median : 52.80  
##  Mean   : 23042                 Mean   :124.9   Mean   : 53.96  
##  3rd Qu.: 33102                 3rd Qu.:145.0   3rd Qu.: 61.40  
##  Max.   :212000                 Max.   :570.0   Max.   :470.80  
##    engineSize      manufacturer      n.age              f.age     
##  Min.   :0.000   Audi    :1075   Min.   : 0.000   LowAge    :1938  
##  1st Qu.:1.500   BMW     :1096   1st Qu.: 1.000   LowMidAge :1426  
##  Median :2.000   Mercedes:1308   Median : 3.000   HighMidAge: 798  
##  Mean   :1.922   VW      :1521   Mean   : 2.788   HighAge   : 838
##  3rd Qu.:2.000                   3rd Qu.: 4.000
##  Max.   :6.200                   Max.   :20.000
##          f.price              f.mileage           f.tax             f.
mpg
##  LowPrice    :1257   LowMileage    :1250   Lowtax    :1349   Lowmpg
:1257
##  LowMidPrice :1244   LowMidMileage :1250   LowMidtax :  34   LowMidmpg
:1243
##  HighMidPrice:1258   HighMidMileage:1250   HighMidtax:2610   HighMidmp
g:1342
##  HighPrice   :1241   HighMileage   :1250   Hightax   :1007   Highmpg
:1158
##
##
##             f.engineSize  univ_outl_count
##  LowengineSize    :1494   Min.   :0.0000
##  LowMidengineSize : 333   1st Qu.:0.0000
##  HighMidengineSize:2105   Median :0.0000
##  HighengineSize   :1068   Mean   :0.5232
##                           3rd Qu.:1.0000
##                           Max.   :4.0000

df = df[-which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff
)),]
```

Profiling

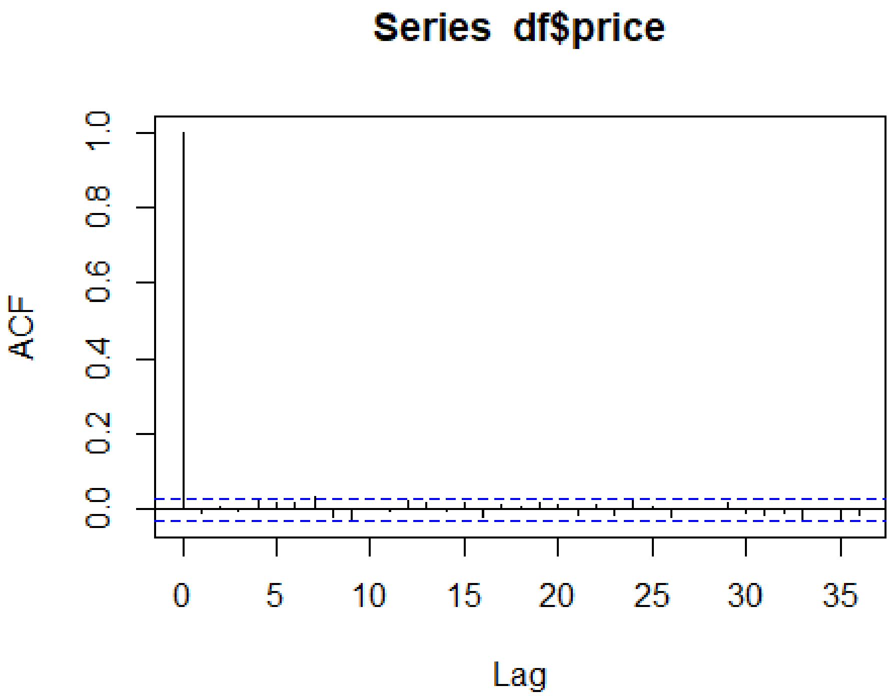
Determine if the response variable (price) has an acceptably normal distribution. Address test to discard serial correlation.

To test for autocorrelation the acf() function is used, of which the result can be seen below: target is not normally distributed and autocorrelation is found, so a randomized dataframe is reset.

```
acf(df$price)
```



```
ll <- sample(1:nrow(df),nrow(df))
df <- df[ll,]
acf(df$price)
```



```
shapiro.test(df$price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$price
## W = 0.92904, p-value < 2.2e-16
```

```
ks.test(df$price, 'pnorm', mean(df$price), sd(df$price))
```

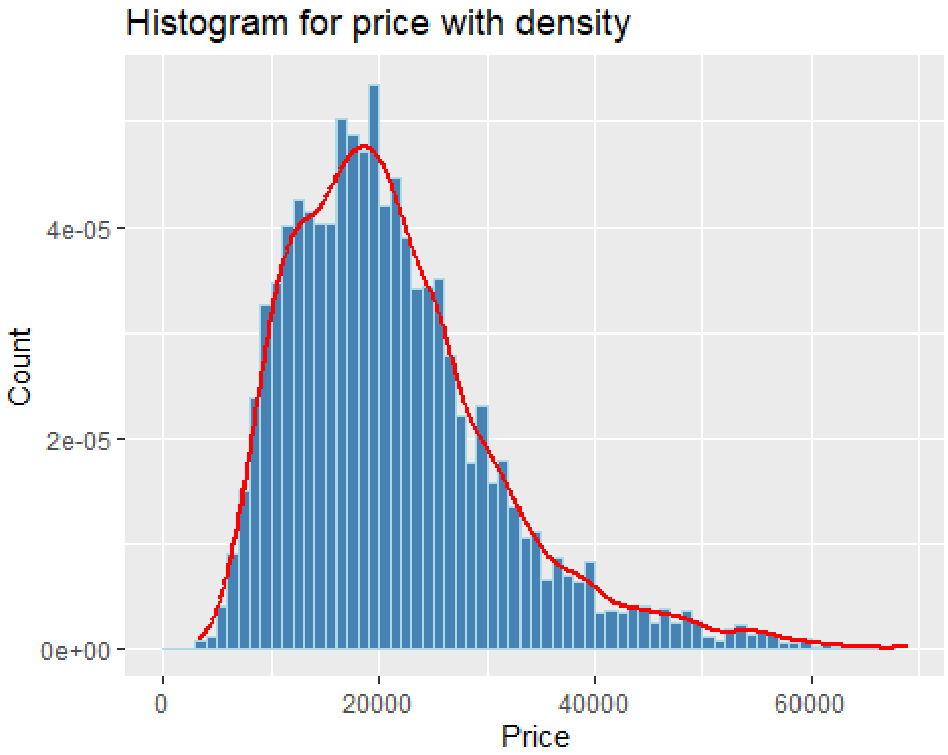
```
## Warning in ks.test.default(df$price, "pnorm", mean(df$price), sd(df$pr
ice)):
## ties should not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  df$price
## D = 0.085964, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ggplot(data=df, aes(price, y = ..density..)) +
  geom_histogram(breaks=seq(0, max(df$price), by=1000),
                 col='lightblue',
                 fill='steelblue') +
  geom_density(lwd=1,
               col='red') +
  labs(title="Histogram for price with density", x="Price", y="Count")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.
4.0.
## ℹ Please use `linewidth` instead.
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot
2 3.4.0.
## ℹ Please use `after_stat(density)` instead.
```



Indicate by exploration of the data which are apparently the variables most associated with the response variable (use only the indicated variables).

To do this, the condes function of the FactoMiner package is used, which for the numeric response variable ‘price’ calculates the correlation of each of the quantitative variables and the coefficient of determination (R^2) for the qualitative variables, together with a p-value for significance.

For the quantitative variables it is clear both engineSize and year are highly significant positively correlated (r > 0.50, p = 0) to the price. This seems logical as the higher the newer the car is and the bigger the engine, the more it would cost. The mileage and miles per gallon (mpg) are highly significant negatively correlated to the price (r < -0.50 p = 0 for mileage) which also to be expected: the more a car has driven, the less its value and small motors are more efficient (lower mpg). Tax has a less but also significant positive correlation to the price (r = 0.35, p ~ 0), which makes sense semantically again. To further illustrate the correlations, a correlation matrix is plotted as well.

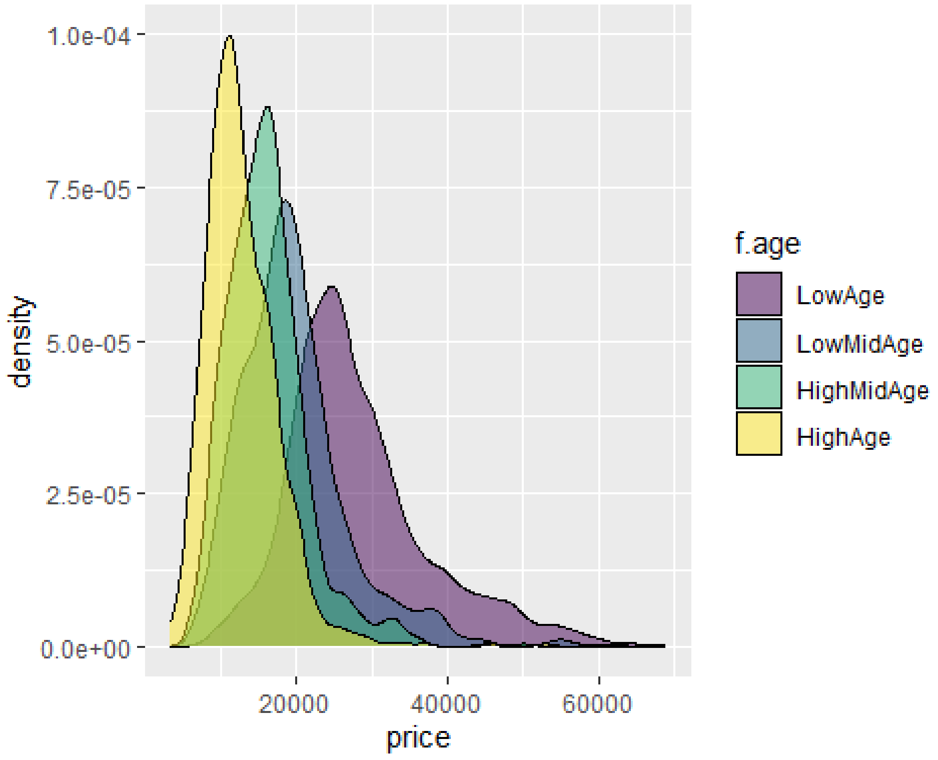
For the qualitative variables it is clear that the model explains the most variance in the price variable (R^2 = 0.45, p = 0) (only the price factor scores higher which is to be expected as it is build on the numeric value). This is to be expected as a specific model subsumes a whole series of other variables. The influence of the other qualitative variables are in order (highest R^2): (price, model), age, mileage, mpg, tax, transmission, engineSize, manufacturer and fuelType. Manufacturer and fuel type are poorly associated as they have R^2-values under 10%.

```
require(FactoMineR)

## Loading required package: FactoMineR

require(corrplot)

res.con = condes(df,3)
res.con$quanti

##                   correlation       p.value
## year               0.61065656  0.000000e+00
## engineSize         0.58268873  0.000000e+00
## tax                0.42742772 4.498770e-215
## univ_outl_count    0.06370257  8.836068e-06
## mileage           -0.54901956  0.000000e+00
## mpg               -0.56232879  0.000000e+00
## n.age             -0.61065656  0.000000e+00

res.con$quali

##                        R2       p.value
## model         0.484396672  0.000000e+00
## f.age         0.375029624  0.000000e+00
## f.price       0.804952680  0.000000e+00
## f.mileage     0.334208180  0.000000e+00
## f.tax         0.269970309  0.000000e+00
## f.mpg         0.350331052  0.000000e+00
## transmission 0.243614811 3.872968e-295
## f.engineSize 0.226361569 7.061380e-270
## manufacturer 0.093357380 8.076981e-103
## fuelType     0.006571193  1.116933e-07
```

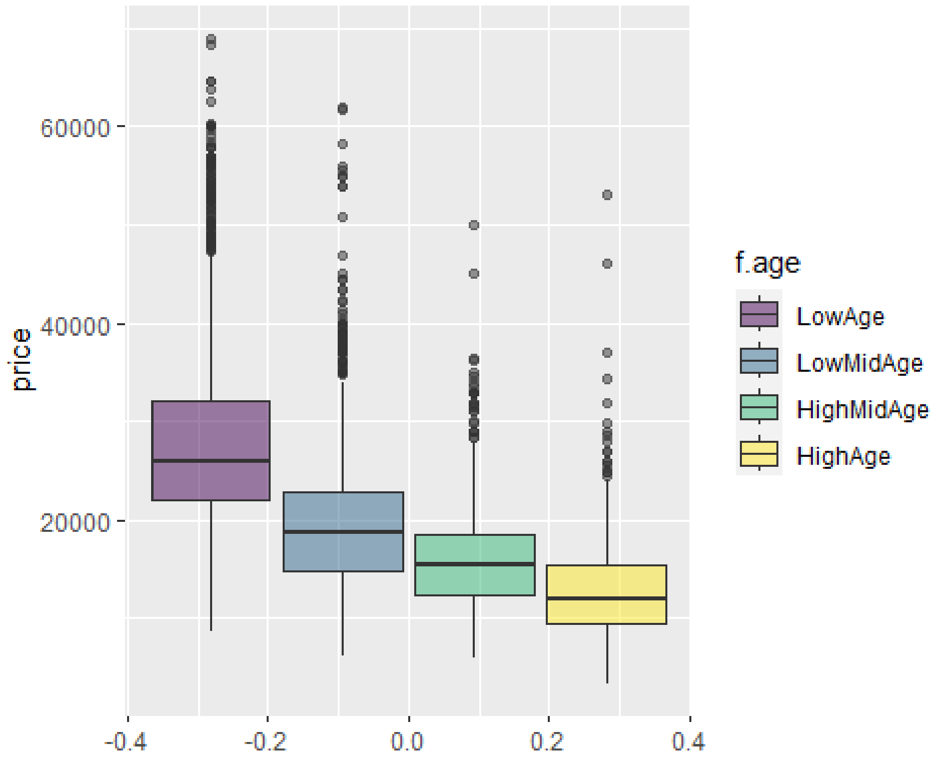
Define a polytomic factor f.age for the covariate car age according to its quartiles and argue if the average price depends on the level of age. Statistically justify the answer.

Given the evidence below, we argue that the price is dependent on the level of age. Firstly, in the boxplots it can be observed that the older the car (Q1) the lower the average price of the car. This is also seen clearly in the distribution plot below. Secondly, the wilcoxon test shows that the means of these levels are not equal meaning that some relation exists between the factor and the response variable.

```
ggplot(df, aes(x=price, fill=f.age)) +
  geom_density(alpha=.5)
```



```
ggplot(df, aes(y=price, fill=f.age)) +
  geom_boxplot(alpha=0.5)
```



We perform pairwise wilcoxon test to check for similar means. Looking at the boxplot and distribution plot we can already expect that these are not going to be similar. The result of the wilcoxon test indicates our hypothesis was right that there is a clear difference between the means of the different quartiles.

```
pairwise.wilcox.test(df$price, df$f.age)

## 
##  Pairwise comparisons using Wilcoxon rank sum test with continuity cor
rection 
## 
## data:  df$price and df$f.age 
## 
##             LowAge LowMidAge HighMidAge
## LowMidAge  <2e-16 -         -         
## HighMidAge <2e-16 <2e-16    -         
## HighAge    <2e-16 <2e-16    <2e-16    
## 
## P value adjustment method: holm
```

Calculate and interpret the anova model that explains car price according to the age factor and the fuel type.

The calculation and interpretation of the anova model that explains the car price according to the age factor and fuel type was done in 4 steps. First, box plots of the fuel type and age factor are presented to show that there is variability in the means of both factors, meaning that they both could have significant influence on the car price. Next to the box plots, wilcoxon and oneway tests are performed to assess if the means of the categories in the factors are statistically different and if the variances between the categories in the factors are statistically different.

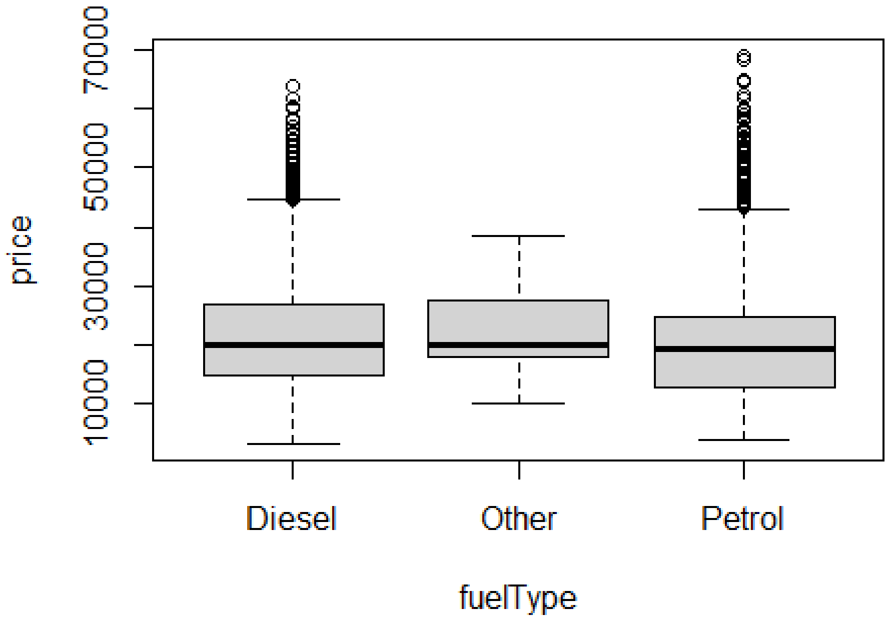
Step 2 involves around inspecting the models for only fuel type and age. From the QQ plots of both models, it can be observed that a transformation might be needed in order to properly assess the quality of the models. Therefore, both the anova before and after transformation is considered, to see if this transformation has any effect on the interaction between the two factors.

Step 3 thus implies considering the interaction and the anova model without the transformation. From the interaction plots, it can be observed that the age factor has a large influence on the price, and that although smaller, the fuel types also influence the price. Next to this, it can be observed that the “Other” category (which mostly consisted of Hybrid and Electric vehicles) deviates from the pattern of the other fuel types in age quartile 4. It seems that newer hybrid cars might be more expensive than older ones. Five linear models are created in order to assess the interaction between the factors. These models are: - m0: Null model, which models the price vs a constant. - m1: Interaction model, which models the price vs the age factor times the fuel type factor. - m2: Interaction model, which models the price vs the age factor plus the fuel type factor. - m3: standard model, which models the price vs the age factor. - m4: standard model, which models the price vs the fuel type factor.

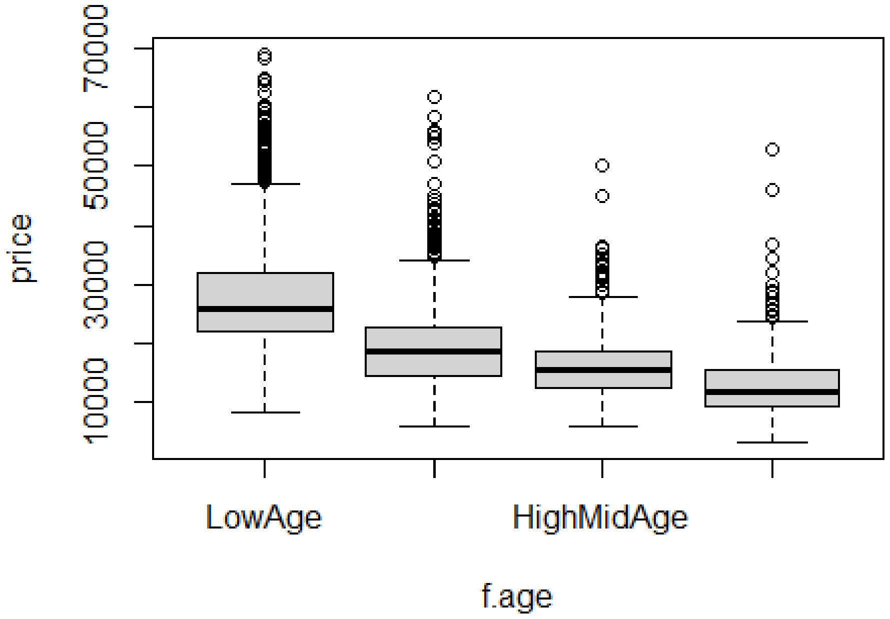
Next, the summaries of the interaction and standard models are considered to see how much of the variation in the price variable these are able to explain. Interestingly, both interaction models explain nearly the same amount of variation, which might indicate that no significant interaction is present. Also, the standard model for the age factor (m3) has an explainability of 31.8%, whereas m4 for the fuel type factor only explains 0.3%. ANOVA models checking the null model versus all other models show that all models are statistically different from the null model, indicating that some relation with the response variable is present. Next to this, additive ANOVA models are considered to see if the addition of one of the two factors influences the model, which was found to be true. Lastly, the interaction models are compared using an ANOVA model, and are found to be significantly different (p = 0.003229).

Now that the factors are analyzed before a transformation on the response variable, interaction between the factors will be considered using a transformation, which is step 4. Using BoxCox, it can be observed that a transformation can be performed. Since lambda is close to 0, a logarithmic transformation on the response variable seems suitable. Using this transformation the interaction plot can be observed, which shows that the influence seems to be mostly the same as the influence on the non-transformed price, except for the severity of the interaction between the “Other” fuel type factor and the “AgeQ4” factor. Using the same approach as in step 3, 5 models were created. After the transformation, both interaction models explain more variation then the once before transformation. However, they almost explain the same amount of variation compared to each other. The model on the age factor now explains 42% and the model the fuel type factor 0.7%, which for both is improvement compared to the non-transformed models. All models are significantly different from the null model. Addition of both factors to the other model (comparing m2 to m3 and m4) is also found to be significant. However, when comparing the interaction models with each other, a p-value of 0.1186 was found, implying that after the transformation, the interaction is not significant anymore. This could indicate that the interaction found in the non-transformed price is explaining a sub-group that is not distinguishable after the transformation.

```r
summary(df$fuelType)

## Diesel  Other Petrol 
##   2728     46   2085

boxplot(price~fuelType, data=df)
```



```
boxplot(price~f.age, data=df)
```



```
pairwise.wilcox.test(df$price, df$f.age)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity cor
rection
##
## data:  df$price and df$f.age
##
##              LowAge LowMidAge HighMidAge
## LowMidAge  <2e-16 -         -
## HighMidAge <2e-16 <2e-16    -
## HighAge    <2e-16 <2e-16    <2e-16
##
## P value adjustment method: holm
```

```
pairwise.wilcox.test(df$price, df$fuelType)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity cor
rection
##
## data:  df$price and df$fuelType
##
##        Diesel  Other
## Other  0.318   -
## Petrol 1.6e-07 0.092
##
## P value adjustment method: holm
```

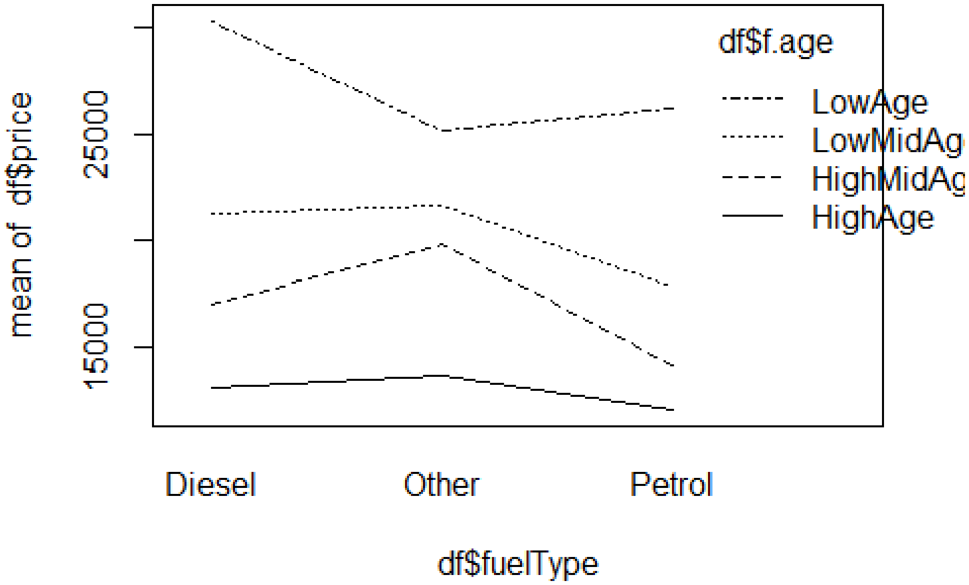
```
oneway.test(df$price~df$f.age)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  df$price and df$f.age
## F = 1090.1, num df = 3.0, denom df = 2396.5, p-value < 2.2e-16
```

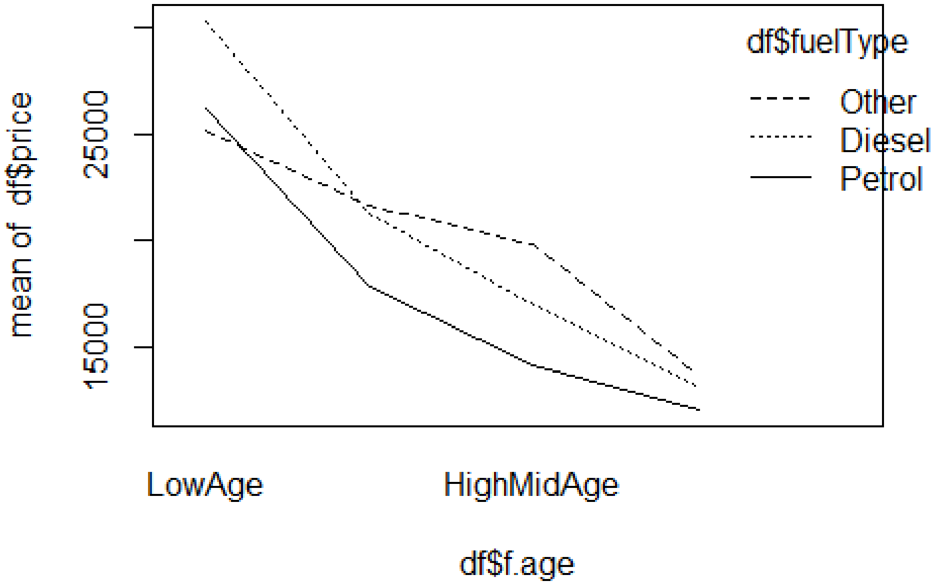
```
oneway.test(df$price~df$fuelType)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  df$price and df$fuelType
## F = 16.234, num df = 2.00, denom df = 123.45, p-value = 5.505e-07
```

```
m_fuel <- lm(price ~ fuelType, data=df)
m_age <- lm(price ~ f.age, data=df)

par(mfrow=c(1,1))
interaction.plot(df$fuelType, df$f.age, df$price)
```



```
par(mfrow=c(1,1))
interaction.plot(df$f.age, df$fuelType, df$price)
```



```
options(contrasts = c("contr.treatment", "contr.treatment"))
m0 <- lm(price ~ 1, data=df)
m1 <- lm(price ~ f.age*fuelType, data=df)
m2 <- lm(price ~ f.age + fuelType, data=df)
m3 <- lm(price ~ f.age, data=df)
m4 <- lm(price ~ fuelType, data=df)

summary(m1)
```

```
##
## Call:
## lm(formula = price ~ f.age * fuelType, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -17817  -4693  -1225   3118  43990
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     30312.4      243.5 124.470  < 2e-16 **
*
## f.ageLowMidAge                  -9017.1      370.0 -24.373  < 2e-16 **
*
## f.ageHighMidAge                -13318.5      415.3 -32.069  < 2e-16 **
*
## f.ageHighAge                   -17255.9      405.3 -42.571  < 2e-16 **
*
## fuelTypeOther                   -5121.4     1744.5  -2.936  0.00334 **
## fuelTypePetrol                  -4135.9      346.6 -11.932  < 2e-16 **
*
## f.ageLowMidAge:fuelTypeOther     5417.2     2677.8   2.023  0.04313 *
## f.ageHighMidAge:fuelTypeOther    7950.8     2970.9   2.676  0.00747 **
## f.ageHighAge:fuelTypeOther       5676.5     4695.5   1.209  0.22675
## f.ageLowMidAge:fuelTypePetrol     642.6      533.3   1.205  0.22831
## f.ageHighMidAge:fuelTypePetrol   1284.0      664.7   1.932  0.05343 .
## f.ageHighAge:fuelTypePetrol      3104.7      689.1   4.505 6.79e-06 **
*
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7530 on 4847 degrees of freedom
## Multiple R-squared:  0.4059, Adjusted R-squared:  0.4045
## F-statistic:   301 on 11 and 4847 DF,  p-value: < 2.2e-16
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = price ~ f.age + fuelType, data = df)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max 
## -17970  -4727  -1231   3059  43878 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      29870.9      204.9 145.748   <2e-16 ***
## f.ageLowMidAge   -8645.2      265.7 -32.535   <2e-16 ***
## f.ageHighMidAge -12665.1      321.9 -39.349   <2e-16 ***
## f.ageHighAge    -16135.2      325.3 -49.605   <2e-16 ***
## fuelTypeOther    -1260.8     1122.8  -1.123    0.262    
## fuelTypePetrol   -3311.3      222.2 -14.900   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7547 on 4853 degrees of freedom
## Multiple R-squared:  0.4024, Adjusted R-squared:  0.4018 
## F-statistic: 653.5 on 5 and 4853 DF,  p-value: < 2.2e-16
```

```
summary(m3)
```

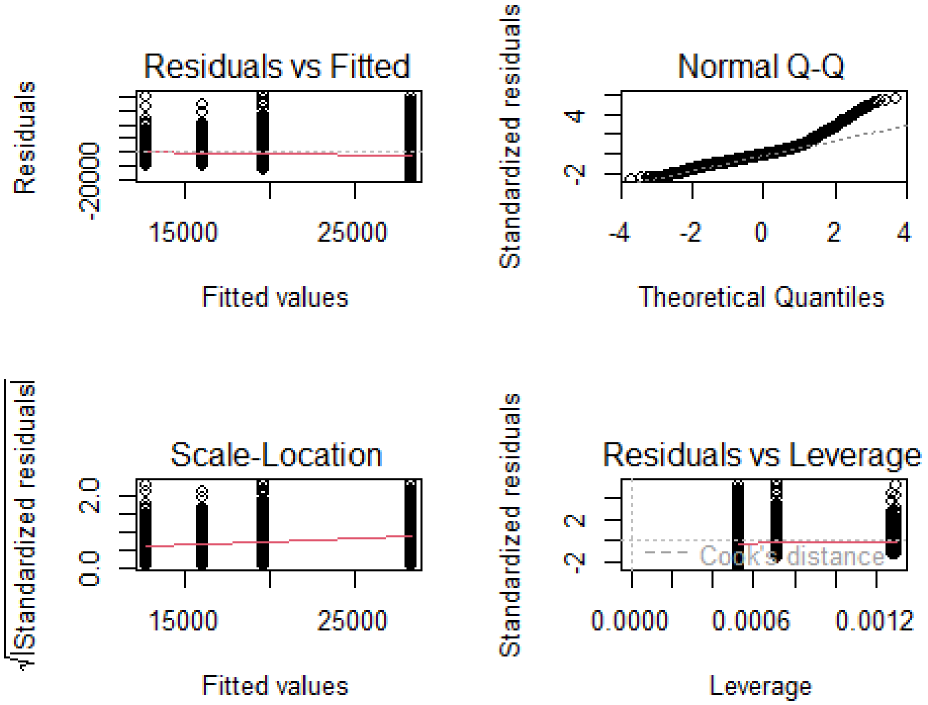
```
## 
## Call:
## lm(formula = price ~ f.age, data = df)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -19650  -4787  -1174   3198  42235 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      28240.0      176.7  159.82   <2e-16 ***
## f.ageLowMidAge   -8574.9      271.6  -31.57   <2e-16 ***
## f.ageHighMidAge -12200.7      327.5  -37.25   <2e-16 ***
## f.ageHighAge    -15485.4      329.5  -47.00   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7716 on 4855 degrees of freedom
## Multiple R-squared:  0.375,  Adjusted R-squared:  0.3746 
## F-statistic: 971.1 on 3 and 4855 DF,  p-value: < 2.2e-16
```

```
summary(m4)
```

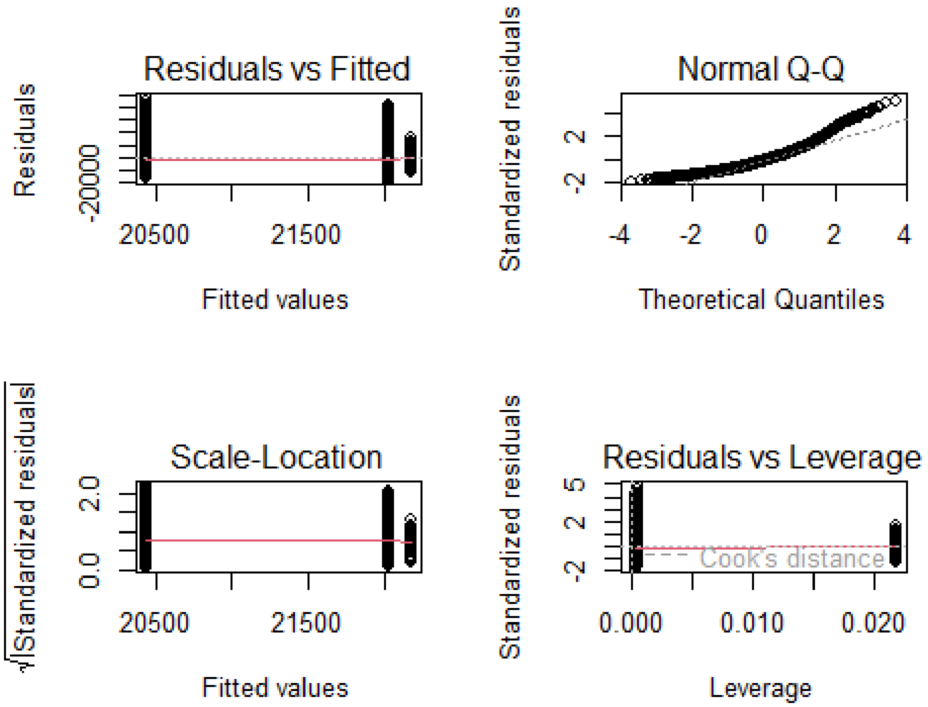
```
## 
## Call:
## lm(formula = price ~ fuelType, data = df)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -18684  -7046  -1741   4793  48556 
## 
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     22034.4      186.2 118.308  < 2e-16 ***
## fuelTypeOther     138.9     1446.3   0.096    0.923
## fuelTypePetrol  -1595.4      283.0  -5.638 1.82e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9728 on 4856 degrees of freedom
## Multiple R-squared:  0.006571,   Adjusted R-squared:  0.006162
## F-statistic: 16.06 on 2 and 4856 DF,  p-value: 1.117e-07
```

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



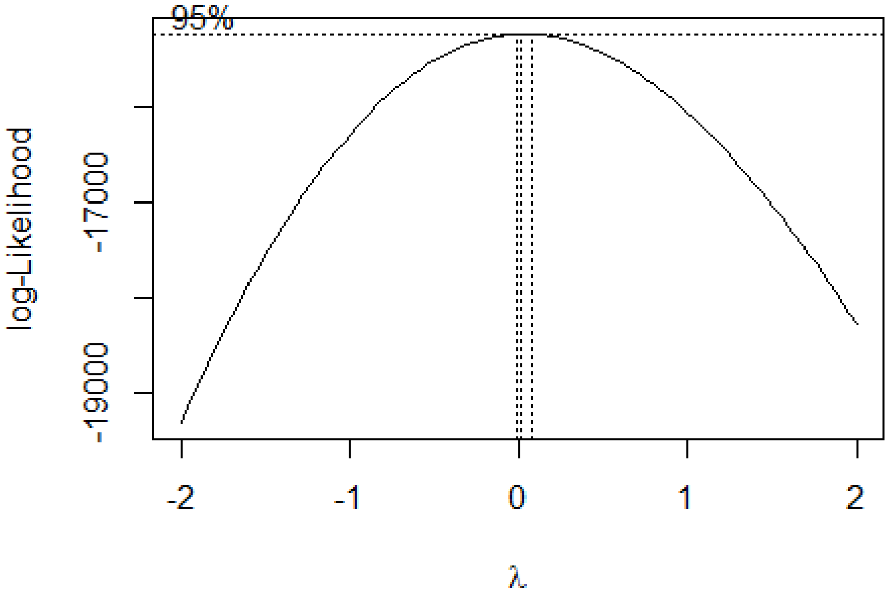
```
par(mfrow=c(2,2))
plot(m4, id.n = 0)
```



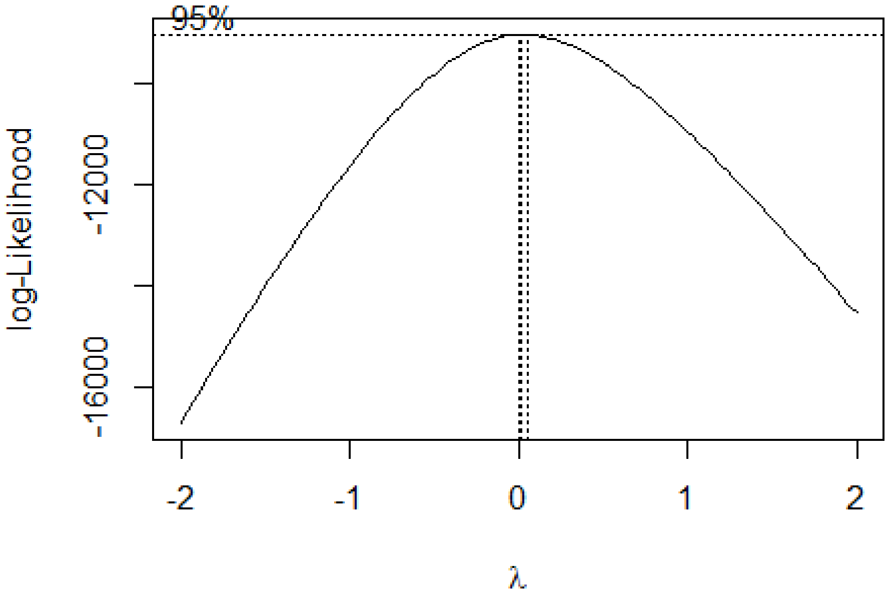
```
anova(m2, m1)

## Analysis of Variance Table
##
## Model 1: price ~ f.age + fuelType
## Model 2: price ~ f.age * fuelType
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1   4853 2.7644e+11
## 2   4847 2.7482e+11  6 1618419380 4.7574 7.654e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

rm(m0, m1, m2, m3, m4)
```

From here the transformation is applied. Moreover, applying a formula using all variables in the dataset to predict the price in boxcox, still indicates that a logarithmic transformation is feasable such that from now one only the logarithmically transformed price variable will be used in all linear models.

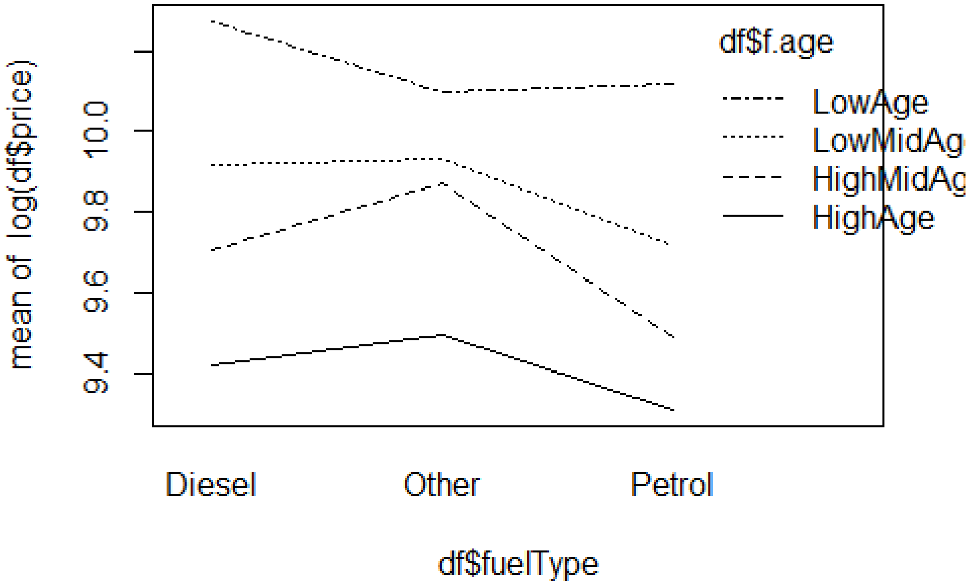
```
library(MASS)
par(mfrow=c(1,1))
boxcox(price ~ f.age+fuelType, data=df)
```



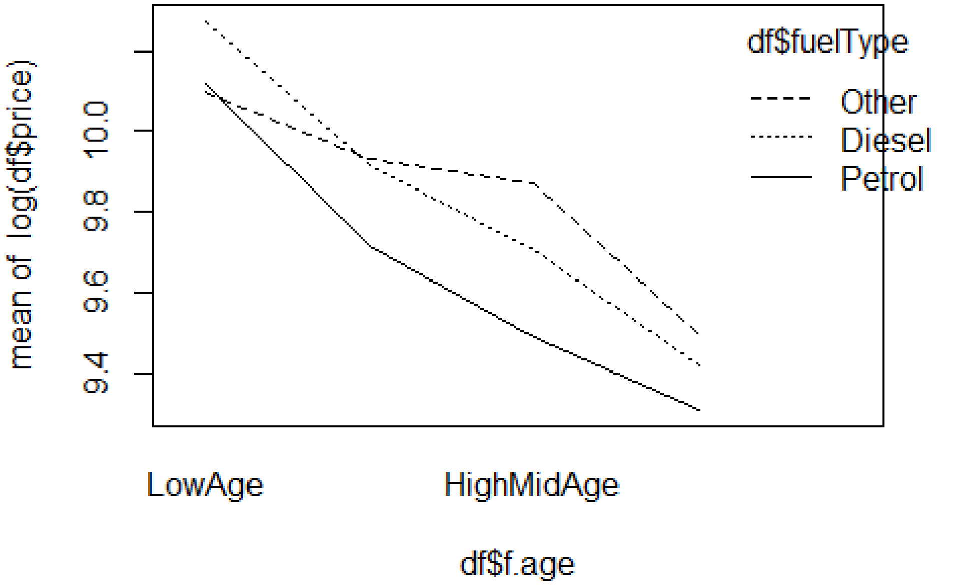
```
boxcox(price ~. , data=df)
```



```
par(mfrow=c(1,1))
interaction.plot(df$fuelType, df$f.age, log(df$price))
```



```
par(mfrow=c(1,1))
interaction.plot(df$f.age, df$fuelType, log(df$price))
```



```
m0l <- lm(log(price) ~ 1, data=df)
m1l <- lm(log(price) ~ f.age*fuelType, data=df)
m2l <- lm(log(price) ~ f.age + fuelType, data=df)
m3l <- lm(log(price) ~ f.age, data=df)
m4l <- lm(log(price) ~ fuelType, data=df)


anova(m2l, m1l)

## Analysis of Variance Table
##
## Model 1: log(price) ~ f.age + fuelType
## Model 2: log(price) ~ f.age * fuelType
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1   4853 526.84
## 2   4847 524.51  6    2.3285 3.5863 0.001504 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m3l, m2l)

## Analysis of Variance Table
##
## Model 1: log(price) ~ f.age
## Model 2: log(price) ~ f.age + fuelType
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   4855 561.84
## 2   4853 526.84  2    34.994 161.17 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m4l, m2l)

## Analysis of Variance Table
##
## Model 1: log(price) ~ fuelType
## Model 2: log(price) ~ f.age + fuelType
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   4856 981.68
## 2   4853 526.84  3    454.84 1396.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Do you think that the variability of the price depends on both factors? Does the relation between price and age factor depend on fuel type?

As shown above (in #4) an interaction is present when the data is not transformed using the logarithm. However, after the transformation, the interaction seems to be weaker and not significant. Both models that describe the price (non-transformed and transformed) only using the fuel type report an incredibly small explainability. At the same time, from the ANOVA models we learn that the addition to the model of the fuel type factor is significant, which is also supported by the interaction plots. Thus it seems that if the price is only modeled using these two factors, it makes sense to use both factors. Looking at Aikakes information criterion, it can also be observed that the interaction models perform best, and that there is little difference between adding and multiplying the factors. Thus, from this perspective it could be argued that a multiplying interaciton is useful. However, this does not guarantee that in a model with better explanatory variables, the fuel type factor will become less important for explaining the variability.

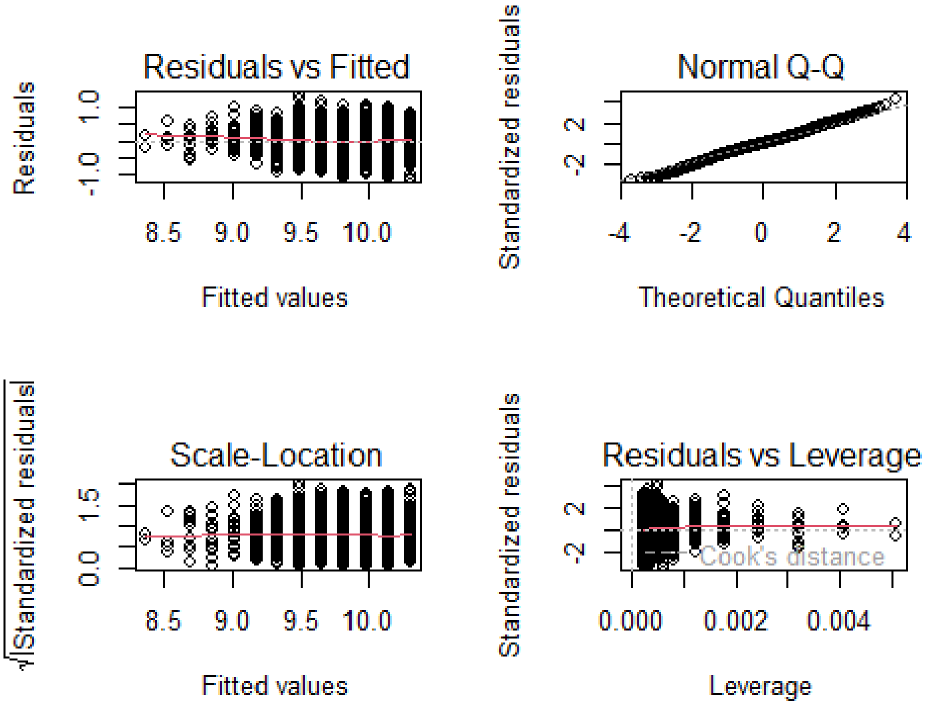
```
AIC(m0l, m1l, m2l, m3l, m4l)

##     df      AIC
## m0l  2 6065.928
## m1l 13 2998.550
## m2l  7 3008.073
## m3l  5 3316.548
## m4l  4 6026.158

rm(m0l, m1l, m2l, m3l, m4l)
```

Calculate the linear regression model that explains the price from the age: interpret the regression line and assess its quality.

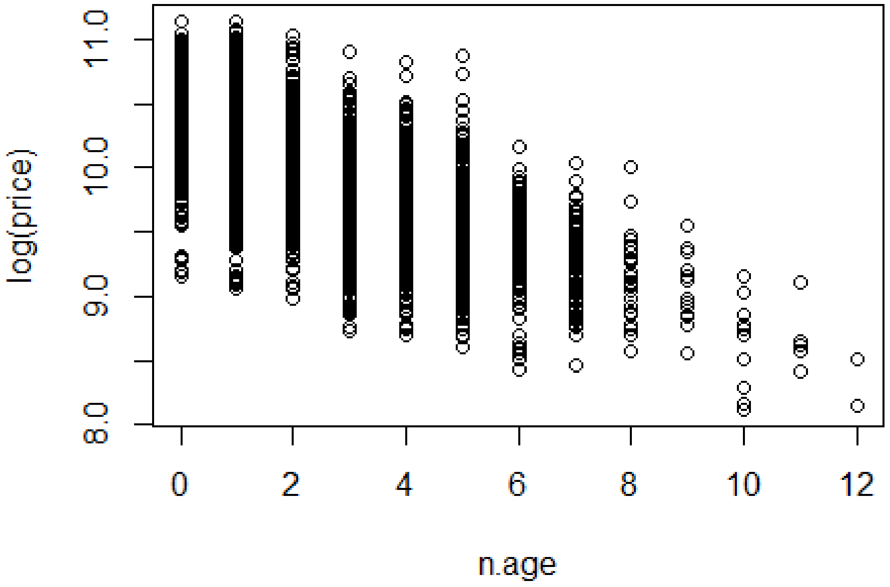
A linear model using the logarithmic price and n.age variable is constructed. It yields an R-sq of 49%, which is insufficient for accurate predictions. However, all diagnostic plots show solid results. The residuals vs fitted plot yields a more or less straight line which confirms linearity. The Q-Q plot indicates non normally distributed residuals in the model. The scale-location plot gives a straight line which indicates that homoscedastisity is satisfied. The Residuals vs. leverage plot shows that there are some high leverage points in the model. An influence plot confirms this behaviour: residual outliers are present (2173 and some more) and influent data (2078, 2169 and 1861).

```
require(MASS)

lmAgeLog = lm(log(price)~n.age, data = df)
par(mfrow=c(2,2))
summary(lmAgeLog)

##
## Call:
## lm(formula = log(price) ~ n.age, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.14197 -0.20079 -0.00035  0.19895  1.38208
##
## Coefficients:
```

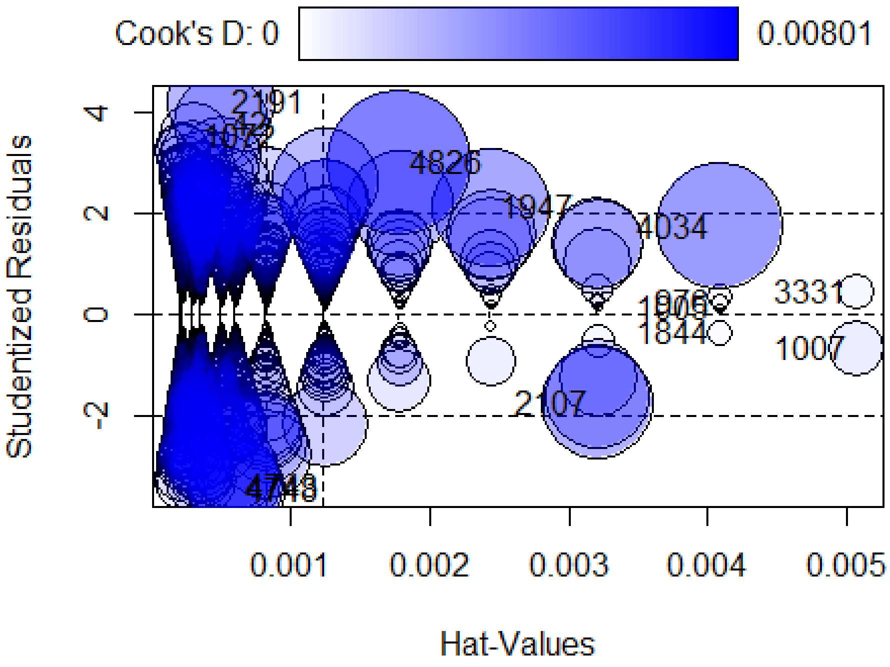
```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.299962   0.008128 1267.20   <2e-16 ***
## n.age       -0.160818   0.002467  -65.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3298 on 4857 degrees of freedom
## Multiple R-squared:  0.4666, Adjusted R-squared:  0.4665
## F-statistic:  4249 on 1 and 4857 DF,  p-value: < 2.2e-16

plot(lmAgeLog,id.n=0)
```



```
par(mfrow=c(1,1))
plot(log(price) ~ n.age, data = df)
```



```
influencePlot(lmAgeLog, id=list(n=5))
```



```
##           StudRes          Hat        CookD
## 4743 -3.4007071 0.0006073170 3.506260e-03
## 3331  0.4438897 0.0050675003 5.018711e-04
## 1007 -0.6808879 0.0050675003 1.180781e-03
## 976   0.2449390 0.0040803137 1.229247e-04
## 1844 -0.3619023 0.0040803137 2.683488e-04
## 1005  0.1323094 0.0040803137 3.586809e-05
## 2107 -1.7467189 0.0032050370 4.902978e-03
## 42    3.7672200 0.0005073028 3.591878e-03
## 2191  4.1985774 0.0005073028 4.458386e-03
## 1947  2.1081117 0.0024416703 5.434982e-03
## 4034  1.7426028 0.0040803137 6.218047e-03
## 4748 -3.4673226 0.0006073170 3.644629e-03
## 1072  3.5304678 0.0003034857 1.887473e-03
## 4826  2.9920174 0.0017902135 8.014396e-03
```

What is the percentage of the price variability that is explained by the age of the car?

As explained in question 6: for the log(price) ~ n.age model about 49% of the price variability is explained. This is insufficient for accurate prediction, such that additional explanatory variables are expected to be useful.

##. Do you think it is necessary to introduce a quadratic term in the equation that relates the price to its age?

The polynomial terms do not significantly improve the R-sq value (~0.005 improvement). The anova test indicates that the models are different p-value very low.

Lastly, looking at AIC, the addition of this polynomial term improves the criterion significantly.

Concluding, as the benefit of adding a polynomial term to the model, in terms of explainability, is very small, but based on inferential Fisher test it is necessary to add up to a cubic term.

```
boxTidwell(log(price)~I(n.age+0.001), data = df)

##  MLE of lambda Score Statistic (z)  Pr(>|z|)
##        0.87001              4.9186 8.716e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## iterations =  4

lmAgeLog2 = lm(log(price)~poly(n.age,3), data=df)

summary(lmAgeLog)

##
## Call:
## lm(formula = log(price) ~ n.age, data = df)
```

```
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.14197 -0.20079 -0.00035  0.19895  1.38208 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 10.299962   0.008128 1267.20   <2e-16 ***
## n.age       -0.160818   0.002467  -65.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3298 on 4857 degrees of freedom
## Multiple R-squared:  0.4666, Adjusted R-squared:  0.4665 
## F-statistic:  4249 on 1 and 4857 DF,  p-value: < 2.2e-16
```

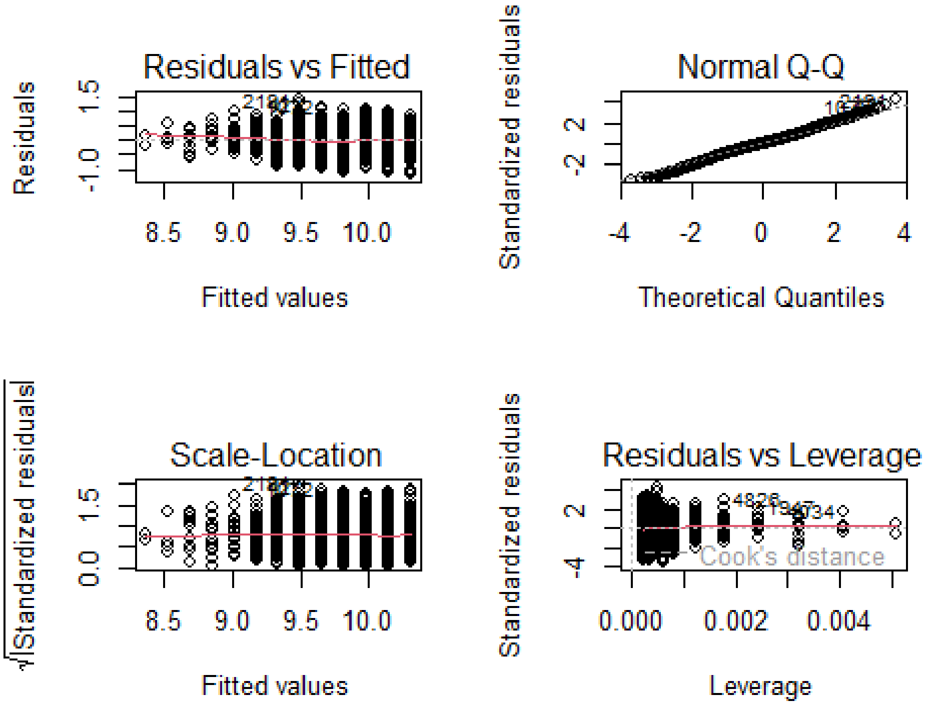
```
summary(lmAgeLog2)
```

```
## 
## Call:
## lm(formula = log(price) ~ poly(n.age, 3), data = df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.20620 -0.19838 -0.00209  0.19752  1.38374 
## 
## Coefficients:
##                   Estimate Std. Error  t value Pr(>|t|)    
## (Intercept)       9.869172   0.004713 2094.174   <2e-16 ***
## poly(n.age, 3)1 -21.498849   0.328505  -65.445   <2e-16 ***
## poly(n.age, 3)2   1.952131   0.328505    5.942    3e-09 ***
## poly(n.age, 3)3  -0.792071   0.328505   -2.411   0.0159 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3285 on 4855 degrees of freedom
## Multiple R-squared:  0.4711, Adjusted R-squared:  0.4708 
## F-statistic:  1441 on 3 and 4855 DF,  p-value: < 2.2e-16
```

```
anova(lmAgeLog, lmAgeLog2)
```

```
## Analysis of Variance Table
## 
## Model 1: log(price) ~ n.age
## Model 2: log(price) ~ poly(n.age, 3)
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)    
## 1   4857 528.37                                  
## 2   4855 523.93  2    4.4382 20.563 1.28e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

plot(lmAgeLog)
```



```
AIC(lmAgeLog, lmAgeLog2)

##            df      AIC
## lmAgeLog    3 3014.109
## lmAgeLog2   5 2977.122
```

Are there any additional explanatory numeric variables needed to the car price? Study collinearity effects.

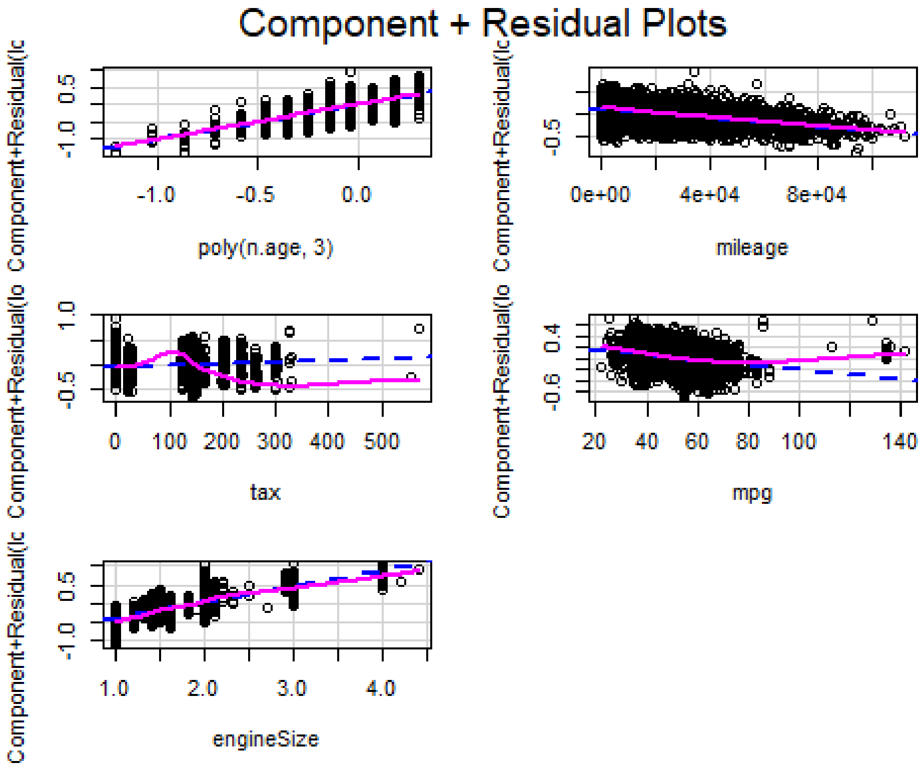
First, all remaining numeric variables are naively additively added to the model. This yields a signficantly better prediction model with an R-sq of about 83%. However, to simplify our model, collinearity is investigated to see if there are variables that are redundant in our model.

The variance inflation factor is calculated for the numeric variables. This indicates whether or not a variable correlates too much with other predictors such that it becomes redundant in the model. In general, a VIF-value larger than 1/(1-R_sq) is considered as showing too much collinear behaviour. The result for every variable is always significantly below this threshold such that no severe collinearity is detected in the model. To further confirm this hypothesis, models are build by alternately removing the highly correlated variables from the logarithmic model. Then, ANOVA is applied to test whether or not the models are significantly predicting something else and AIC to see what model is considered the best. These tests show that the model with all numeric variables performs the best and that no severe collinearity is present in our model.

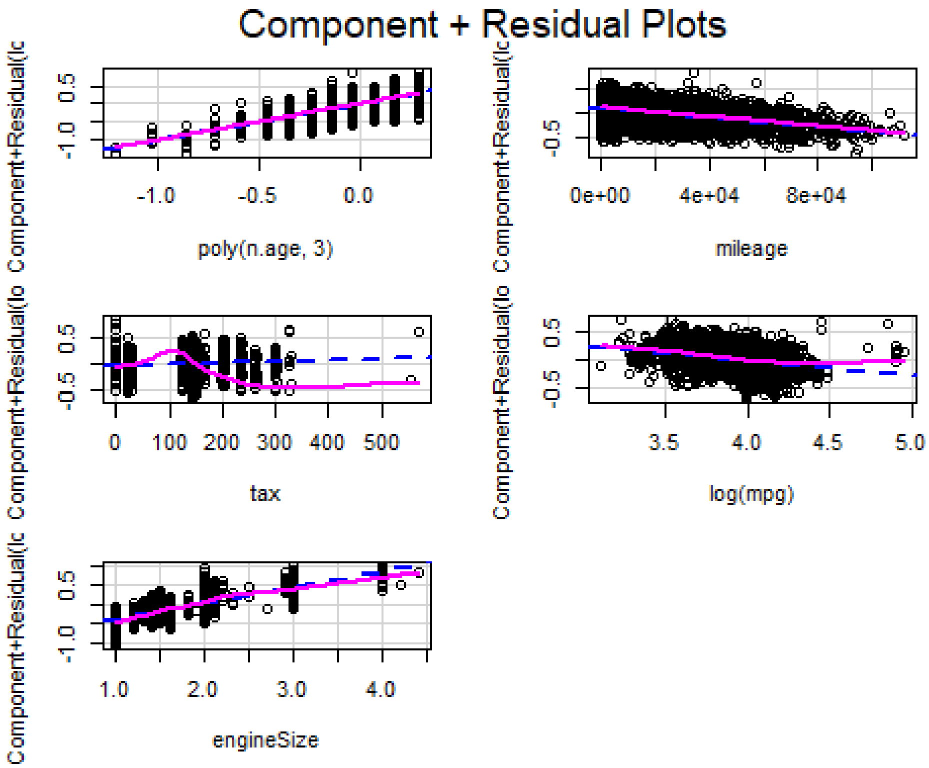
Therefore, the model of choice for the continuation of the project will be the one with all numeric variables. Transformation for mpg is addressed and the model is remarkably improved accounting for AIC statistics. Missfit can be still seen for tax and engineSize.

```
lmNumLog = lm(log(price) ~ poly(n.age,3)+mileage+tax+mpg+engineSize, data
=df)
t = summary(lmNumLog)
vif(lmNumLog)

##                     GVIF Df GVIF^(1/(2*Df))
## poly(n.age, 3) 3.130119  3        1.209465
## mileage        2.764583  1        1.662703
## tax            1.743501  1        1.320417
## mpg            1.844796  1        1.358233
## engineSize     1.215555  1        1.102522

1/(1-t$r.squared)

## [1] 5.774163

lmNumLog2 = lm(log(price) ~ poly(n.age,3)+tax+mpg+engineSize, data=df)
lmNumLog3 = lm(log(price) ~ poly(n.age,3)+mileage+mpg+engineSize, data=df
)
lmNumLog4 = lm(log(price) ~ poly(n.age,3)+tax+mileage+engineSize, data=df
)

anova(lmNumLog2, lmNumLog)

## Analysis of Variance Table
##
## Model 1: log(price) ~ poly(n.age, 3) + tax + mpg + engineSize
## Model 2: log(price) ~ poly(n.age, 3) + mileage + tax + mpg + engineSiz
e
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   4852 189.38
## 2   4851 171.55  1    17.832 504.23 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(lmNumLog3, lmNumLog)

## Analysis of Variance Table
##
## Model 1: log(price) ~ poly(n.age, 3) + mileage + mpg + engineSize
## Model 2: log(price) ~ poly(n.age, 3) + mileage + tax + mpg + engineSiz
e
```

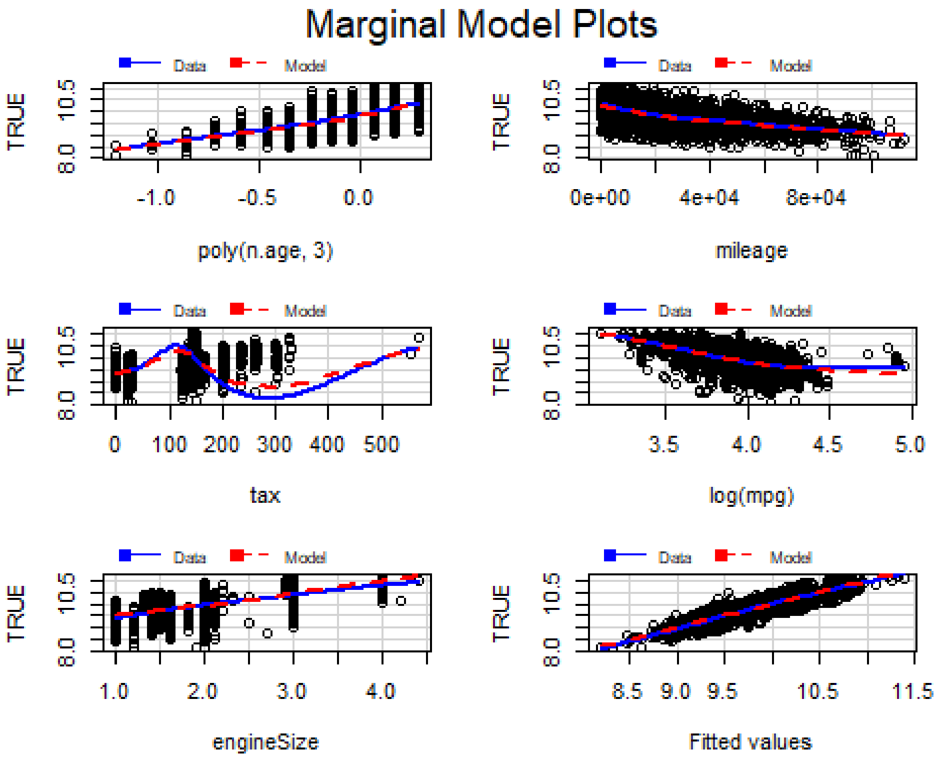
```
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
## 1   4852 172.51                                  
## 2   4851 171.55  1   0.95824 27.096 2.015e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(lmNumLog4, lmNumLog)

## Analysis of Variance Table
## 
## Model 1: log(price) ~ poly(n.age, 3) + tax + mileage + engineSize
## Model 2: log(price) ~ poly(n.age, 3) + mileage + tax + mpg + engineSiz
e
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
## 1   4852 178.94                                  
## 2   4851 171.55  1    7.3924 209.04 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

AIC(lmNumLog, lmNumLog2, lmNumLog3, lmNumLog4)

##           df       AIC
## lmNumLog   9 -2439.811
## lmNumLog2  8 -1961.308
## lmNumLog3  8 -2414.745
## lmNumLog4  8 -2236.816

crPlots( lmNumLog)
```



```
# boxTidwell(log(price) ~ mileage+mpg, ~ poly(n.age,3)+tax+engineSize,dat
a=df)
#
lmNumLogt = lm(log(price) ~ poly(n.age,3)+mileage+tax+log(mpg)+engineSize
, data=df)
AIC(lmNumLogt,lmNumLog)

##           df       AIC
## lmNumLogt  9 -2528.583
## lmNumLog   9 -2439.811

crPlots( lmNumLogt)
```



```
marginalModelPlots(lmNumLogt)

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



After controlling by numerical variables, indicate whether the additive effect of the available factors on the price are statistically significant.

All the remaining available factors (except the ones defined on the numeric variables like fe. f.age) are now added one by one to the current model. This yields an improvement of R-sq of about 6%. An anova test indicates significant difference in prediction. The vif function indicates no significant collinearity by introducing the factors. Finally, the AIC function shows that the model with the factors is significantly better than the one without. Therefore, it can be concluded that the effect of the available factors is statistically significant and positive in its prediction capabilities.

```
lmNumLog = lm(log(price) ~ poly(n.age,3)+mileage+tax+log(mpg)+engineSize,
data=df)
lmFactLog = lm(log(price) ~ poly(n.age,3)+mileage+tax+log(mpg)+engineSize
+transmission+fuelType+manufacturer, data=df)

summary(lmNumLog)

##
## Call:
## lm(formula = log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) +
##     engineSize, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69937 -0.11714  0.00726  0.12705  0.90504
```

```
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.024e+01  6.977e-02 146.790  < 2e-16 ***
## poly(n.age, 3)1 -1.444e+01  3.085e-01 -46.806  < 2e-16 ***
## poly(n.age, 3)2 -3.513e-01  1.989e-01  -1.766   0.0775 .
## poly(n.age, 3)3 -4.726e-01  1.890e-01  -2.501   0.0124 *
## mileage         -4.848e-06  2.185e-07 -22.190  < 2e-16 ***
## tax              2.716e-04  5.861e-05   4.634 3.68e-06 ***
## log(mpg)        -2.802e-01  1.612e-02 -17.388  < 2e-16 ***
## engineSize       4.243e-01  5.513e-03  76.961  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1863 on 4851 degrees of freedom
## Multiple R-squared:   0.83,  Adjusted R-squared:  0.8297
## F-statistic:  3382 on 7 and 4851 DF,  p-value: < 2.2e-16
```

```
summary(lmFactLog)
```

```
##
## Call:
## lm(formula = log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) +
##     engineSize + transmission + fuelType + manufacturer, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54424 -0.09169  0.00372  0.09775  0.79548
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.196e+01  8.312e-02 143.849  < 2e-16 ***
## poly(n.age, 3)1       -1.283e+01  2.585e-01 -49.630  < 2e-16 ***
## poly(n.age, 3)2       -1.606e+00  1.682e-01  -9.548  < 2e-16 ***
## poly(n.age, 3)3       -3.416e-02  1.548e-01  -0.221  0.82538
## mileage               -4.700e-06  1.797e-07 -26.157  < 2e-16 ***
## tax                    2.689e-05  4.829e-05   0.557  0.57775
## log(mpg)              -5.908e-01  1.817e-02 -32.507  < 2e-16 ***
## engineSize             2.574e-01  6.706e-03  38.388  < 2e-16 ***
## transmissionManual    -9.372e-02  6.628e-03 -14.140  < 2e-16 ***
## transmissionSemi-Auto  1.568e-02  5.628e-03   2.787  0.00535 **
## fuelTypeOther          2.687e-01  2.312e-02  11.624  < 2e-16 ***
## fuelTypePetrol        -1.233e-01  6.702e-03 -18.394  < 2e-16 ***
## manufacturerBMW       -7.561e-02  6.989e-03 -10.818  < 2e-16 ***
## manufacturerMercedes   1.853e-02  6.757e-03   2.743  0.00611 **
## manufacturerVW        -1.797e-01  6.277e-03 -28.622  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1519 on 4844 degrees of freedom
## Multiple R-squared:  0.8872, Adjusted R-squared:  0.8869
## F-statistic:  2721 on 14 and 4844 DF,  p-value: < 2.2e-16
```

```
anova( lmNumLog, lmFactLog)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engi
neSize
## Model 2: log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engi
neSize +
##     transmission + fuelType + manufacturer
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   4851 168.45
## 2   4844 111.74  7    56.709 351.21 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

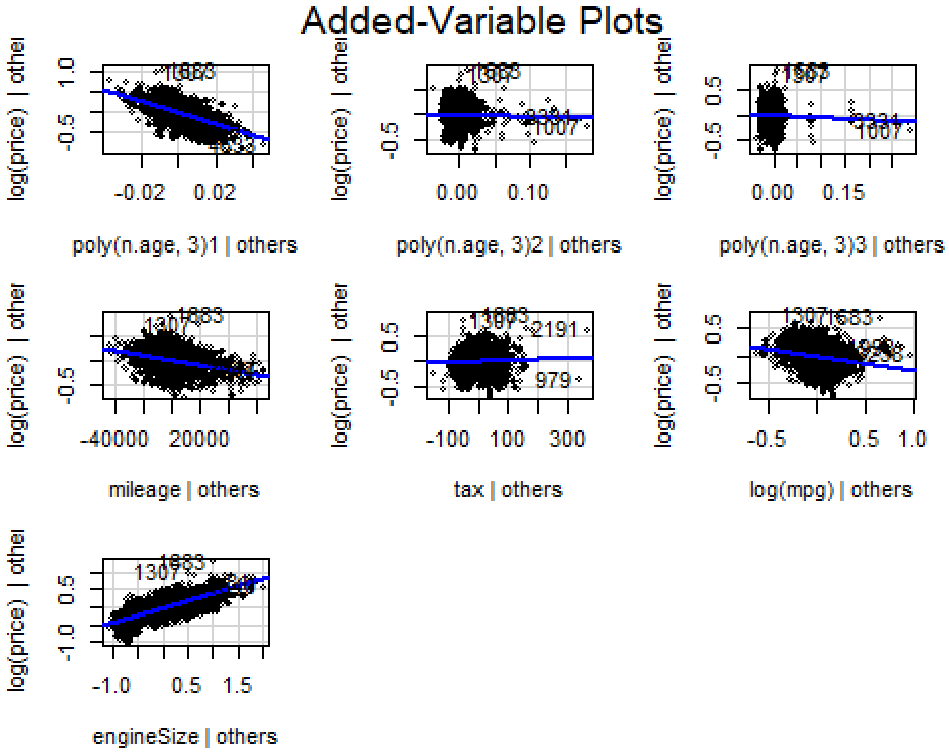
```
vif(lmFactLog)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## poly(n.age, 3) 3.587475  3        1.237271
## mileage        2.817324  1        1.678489
## tax            1.730159  1        1.315355
## log(mpg)       3.657772  1        1.912530
## engineSize     2.808177  1        1.675762
## transmission   1.606442  2        1.125813
## fuelType       2.425412  2        1.247948
## manufacturer   1.531257  3        1.073597
```

```
AIC(lmFactLog, lmNumLog)
```

```
##           df       AIC
## lmFactLog 16 -4509.050
## lmNumLog   9 -2528.583
```

```
avPlots(lmNumLog)
```



```
summary(lmFactLog)

##
## Call:
## lm(formula = log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) +
##     engineSize + transmission + fuelType + manufacturer, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54424 -0.09169  0.00372  0.09775  0.79548
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.196e+01  8.312e-02 143.849  < 2e-16 ***
## poly(n.age, 3)1        -1.283e+01  2.585e-01 -49.630  < 2e-16 ***
## poly(n.age, 3)2        -1.606e+00  1.682e-01  -9.548  < 2e-16 ***
## poly(n.age, 3)3        -3.416e-02  1.548e-01  -0.221  0.82538
## mileage                -4.700e-06  1.797e-07 -26.157  < 2e-16 ***
## tax                     2.689e-05  4.829e-05   0.557  0.57775
## log(mpg)               -5.908e-01  1.817e-02 -32.507  < 2e-16 ***
## engineSize              2.574e-01  6.706e-03  38.388  < 2e-16 ***
## transmissionManual     -9.372e-02  6.628e-03 -14.140  < 2e-16 ***
## transmissionSemi-Auto   1.568e-02  5.628e-03   2.787  0.00535 **
## fuelTypeOther           2.687e-01  2.312e-02  11.624  < 2e-16 ***
## fuelTypePetrol         -1.233e-01  6.702e-03 -18.394  < 2e-16 ***
## manufacturerBMW        -7.561e-02  6.989e-03 -10.818  < 2e-16 ***
## manufacturerMercedes    1.853e-02  6.757e-03   2.743  0.00611 **
## manufacturerVW        -1.797e-01  6.277e-03 -28.622  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1519 on 4844 degrees of freedom
## Multiple R-squared:  0.8872, Adjusted R-squared:  0.8869
## F-statistic:  2721 on 14 and 4844 DF,  p-value: < 2.2e-16
```

Interactions

```
lmFactLogI = lm(log(price) ~( poly(n.age,3)+mileage+tax+log(mpg)+engineSi
ze)*(transmission+fuelType+manufacturer), data=df)

lmFactLogIr <- step( lmFactLogI, k=log(nrow(df)))

## Start:  AIC=-18266.52
## log(price) ~ (poly(n.age, 3) + mileage + tax + log(mpg) + engineSize)
*
##     (transmission + fuelType + manufacturer)
##
##                               Df Sum of Sq    RSS    AIC
## - poly(n.age, 3):fuelType      6   0.15084 101.39 -18310
## - poly(n.age, 3):transmission  6   0.28385 101.52 -18304
## - mileage:manufacturer         3   0.07340 101.31 -18289
## - tax:transmission             2   0.01920 101.26 -18283
## - tax:manufacturer             3   0.20027 101.44 -18282
## - mileage:transmission         2   0.02787 101.27 -18282
## - log(mpg):transmission        2   0.02801 101.27 -18282
## - engineSize:fuelType          2   0.05575 101.29 -18281
## - mileage:fuelType             2   0.07205 101.31 -18280
## - engineSize:transmission      2   0.13762 101.38 -18277
## - log(mpg):fuelType            2   0.22302 101.46 -18273
## <none>                                     101.24 -18267
## - tax:fuelType                 2   0.49859 101.74 -18260
## - engineSize:manufacturer      3   1.04392 102.28 -18242
## - poly(n.age, 3):manufacturer  9   2.36306 103.60 -18231
## - log(mpg):manufacturer        3   1.28286 102.52 -18231
##
## Step:  AIC=-18310.22
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):transmissi
on +
##     poly(n.age, 3):manufacturer + mileage:transmission + mileage:fuelT
ype +
##     mileage:manufacturer + tax:transmission + tax:fuelType +
##     tax:manufacturer + log(mpg):transmission + log(mpg):fuelType +
##     log(mpg):manufacturer + engineSize:transmission + engineSize:fuelT
ype +
##     engineSize:manufacturer
##
##                               Df Sum of Sq    RSS    AIC
## - poly(n.age, 3):transmission  6   0.27926 101.67 -18348
## - mileage:manufacturer         3   0.07296 101.46 -18332
## - tax:manufacturer             3   0.19456 101.58 -18326
## - tax:transmission             2   0.01871 101.41 -18326
## - log(mpg):transmission        2   0.02257 101.41 -18326
## - mileage:transmission         2   0.02910 101.42 -18326
## - mileage:fuelType             2   0.03300 101.42 -18326
## - engineSize:fuelType          2   0.07415 101.46 -18324
## - engineSize:transmission      2   0.14370 101.53 -18320
## <none>                                     101.39 -18310
## - tax:fuelType                 2   0.46197 101.85 -18305
## - log(mpg):fuelType            2   0.52938 101.92 -18302
## - engineSize:manufacturer      3   1.03605 102.42 -18286
## - log(mpg):manufacturer        3   1.29049 102.68 -18274
## - poly(n.age, 3):manufacturer  9   2.38921 103.78 -18273
## 
## Step:  AIC=-18347.78
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize + 
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er + 
##     mileage:transmission + mileage:fuelType + mileage:manufacturer + 
##     tax:transmission + tax:fuelType + tax:manufacturer + log(mpg):tran
smission + 
##     log(mpg):fuelType + log(mpg):manufacturer + engineSize:transmissio
n + 
##     engineSize:fuelType + engineSize:manufacturer
## 
##                                Df Sum of Sq    RSS    AIC
## - mileage:manufacturer         3   0.08342 101.75 -18369
## - log(mpg):transmission        2   0.02021 101.69 -18364
## - tax:transmission             2   0.02082 101.69 -18364
## - tax:manufacturer             3   0.20430 101.87 -18364
## - mileage:fuelType             2   0.03679 101.70 -18363
## - engineSize:fuelType          2   0.06485 101.73 -18362
## - mileage:transmission         2   0.09451 101.76 -18360
## - engineSize:transmission      2   0.17962 101.85 -18356
## <none>                                     101.67 -18348
## - tax:fuelType                 2   0.44422 102.11 -18344
## - log(mpg):fuelType            2   0.56832 102.23 -18338
## - engineSize:manufacturer      3   1.03511 102.70 -18324
## - log(mpg):manufacturer        3   1.42050 103.09 -18306
## - poly(n.age, 3):manufacturer  9   2.74301 104.41 -18295
## 
## Step:  AIC=-18369.26
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize + 
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er + 
##     mileage:transmission + mileage:fuelType + tax:transmission + 
##     tax:fuelType + tax:manufacturer + log(mpg):transmission + 
##     log(mpg):fuelType + log(mpg):manufacturer + engineSize:transmissio
```

```
n +
##     engineSize:fuelType + engineSize:manufacturer
##
##                               Df Sum of Sq    RSS    AIC
## - tax:transmission             2    0.0167 101.77 -18385
## - log(mpg):transmission        2    0.0194 101.77 -18385
## - tax:manufacturer             3    0.2106 101.96 -18385
## - mileage:fuelType             2    0.0358 101.79 -18385
## - engineSize:fuelType          2    0.0667 101.82 -18383
## - mileage:transmission         2    0.0850 101.83 -18382
## - engineSize:transmission      2    0.1932 101.94 -18377
## <none>                                    101.75 -18369
## - tax:fuelType                 2    0.4381 102.19 -18365
## - log(mpg):fuelType            2    0.5673 102.32 -18359
## - engineSize:manufacturer      3    0.9640 102.72 -18349
## - log(mpg):manufacturer        3    1.4700 103.22 -18325
## - poly(n.age, 3):manufacturer  9    3.4025 105.15 -18286
##
## Step:  AIC=-18385.44
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     mileage:transmission + mileage:fuelType + tax:fuelType +
##     tax:manufacturer + log(mpg):transmission + log(mpg):fuelType +
##     log(mpg):manufacturer + engineSize:transmission + engineSize:fuelT
ype +
##     engineSize:manufacturer
##
##                               Df Sum of Sq    RSS    AIC
## - log(mpg):transmission        2    0.0099 101.78 -18402
## - mileage:fuelType             2    0.0365 101.80 -18401
## - engineSize:fuelType          2    0.0681 101.83 -18399
## - mileage:transmission         2    0.0815 101.85 -18399
## - tax:manufacturer             3    0.2619 102.03 -18398
## - engineSize:transmission      2    0.1923 101.96 -18393
## <none>                                    101.77 -18385
## - tax:fuelType                 2    0.4516 102.22 -18381
## - log(mpg):fuelType            2    0.5550 102.32 -18376
## - engineSize:manufacturer      3    0.9682 102.73 -18365
## - log(mpg):manufacturer        3    1.5314 103.30 -18338
## - poly(n.age, 3):manufacturer  9    3.4039 105.17 -18302
##
## Step:  AIC=-18401.95
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     mileage:transmission + mileage:fuelType + tax:fuelType +
##     tax:manufacturer + log(mpg):fuelType + log(mpg):manufacturer +
##     engineSize:transmission + engineSize:fuelType + engineSize:manufac
turer
```

```
##
##                                Df Sum of Sq    RSS    AIC
## - mileage:fuelType             2    0.0418 101.82 -18417
## - engineSize:fuelType          2    0.0740 101.85 -18415
## - tax:manufacturer             3    0.2597 102.04 -18415
## - mileage:transmission         2    0.1242 101.90 -18413
## - engineSize:transmission      2    0.2315 102.01 -18408
## <none>                                     101.78 -18402
## - tax:fuelType                 2    0.4541 102.23 -18397
## - log(mpg):fuelType            2    0.5571 102.33 -18392
## - engineSize:manufacturer      3    0.9660 102.74 -18382
## - log(mpg):manufacturer        3    1.6084 103.39 -18351
## - poly(n.age, 3):manufacturer  9    3.4273 105.20 -18317
##
## Step:  AIC=-18416.93
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     mileage:transmission + tax:fuelType + tax:manufacturer +
##     log(mpg):fuelType + log(mpg):manufacturer + engineSize:transmissio
n +
##     engineSize:fuelType + engineSize:manufacturer
##
##                                Df Sum of Sq    RSS    AIC
## - engineSize:fuelType          2    0.0663 101.89 -18431
## - tax:manufacturer             3    0.2781 102.10 -18429
## - mileage:transmission         2    0.1182 101.94 -18428
## - engineSize:transmission      2    0.2261 102.05 -18423
## <none>                                     101.82 -18417
## - tax:fuelType                 2    0.4847 102.30 -18411
## - log(mpg):fuelType            2    0.6047 102.42 -18405
## - engineSize:manufacturer      3    0.9766 102.80 -18396
## - log(mpg):manufacturer        3    1.7256 103.54 -18361
## - poly(n.age, 3):manufacturer  9    3.4946 105.31 -18329
##
## Step:  AIC=-18430.75
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     mileage:transmission + tax:fuelType + tax:manufacturer +
##     log(mpg):fuelType + log(mpg):manufacturer + engineSize:transmissio
n +
##     engineSize:manufacturer
##
##                                Df Sum of Sq    RSS    AIC
## - tax:manufacturer             3    0.2766 102.16 -18443
## - mileage:transmission         2    0.1207 102.01 -18442
## - engineSize:transmission      2    0.2198 102.11 -18437
## <none>                                     101.89 -18431
## - tax:fuelType                 2    0.4933 102.38 -18424
## - log(mpg):fuelType           2    0.8007 102.69 -18410
## - engineSize:manufacturer     3    0.9925 102.88 -18409
## - log(mpg):manufacturer       3    1.7909 103.68 -18372
## - poly(n.age, 3):manufacturer  9    3.4980 105.38 -18343
##
## Step:  AIC=-18443.04
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     mileage:transmission + tax:fuelType + log(mpg):fuelType +
##     log(mpg):manufacturer + engineSize:transmission + engineSize:manuf
acturer
##
##                               Df Sum of Sq    RSS    AIC
## - mileage:transmission         2    0.1191 102.28 -18454
## - engineSize:transmission      2    0.2645 102.43 -18447
## - tax:fuelType                 2    0.3124 102.47 -18445
## <none>                                     102.16 -18443
## - log(mpg):fuelType            2    0.7591 102.92 -18424
## - engineSize:manufacturer      3    0.9548 103.12 -18423
## - log(mpg):manufacturer        3    1.6279 103.79 -18392
## - poly(n.age, 3):manufacturer  9    3.4664 105.63 -18357
##
## Step:  AIC=-18454.35
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     tax:fuelType + log(mpg):fuelType + log(mpg):manufacturer +
##     engineSize:transmission + engineSize:manufacturer
##
##                               Df Sum of Sq    RSS    AIC
## - engineSize:transmission      2    0.2563 102.54 -18459
## - tax:fuelType                 2    0.3100 102.59 -18457
## <none>                                     102.28 -18454
## - log(mpg):fuelType            2    0.7693 103.05 -18435
## - engineSize:manufacturer      3    0.9586 103.24 -18435
## - log(mpg):manufacturer        3    1.6047 103.89 -18404
## - poly(n.age, 3):manufacturer  9    3.4741 105.75 -18368
## - mileage                      1   15.0319 117.31 -17797
##
## Step:  AIC=-18459.17
## log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     tax:fuelType + log(mpg):fuelType + log(mpg):manufacturer +
##     engineSize:manufacturer
##
##                               Df Sum of Sq    RSS    AIC
## <none>                                     102.54 -18459
## - tax:fuelType                 2    0.3812 102.92 -18458
## - log(mpg):fuelType            2    0.7042 103.24 -18443
## - engineSize:manufacturer      3    1.2610 103.80 -18425
## - log(mpg):manufacturer        3    1.5349 104.07 -18412
## - poly(n.age, 3):manufacturer  9    3.4915 106.03 -18373
## - transmission                 2    6.0349 108.57 -18198
## - mileage                      1   15.0444 117.58 -17802
```

```
summary(lmFactLogIr)
```

```
## 
## Call:
## lm(formula = log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + 
##     engineSize + transmission + fuelType + manufacturer + poly(n.age, 
##     3):manufacturer + tax:fuelType + log(mpg):fuelType + log(mpg):manu
facturer + 
##     engineSize:manufacturer, data = df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.48108 -0.09332  0.00241  0.09160  0.59502 
## 
## Coefficients:
##                                         Estimate Std. Error t value Pr(
>|t|)    
## (Intercept)                            1.293e+01  1.572e-01  82.251  < 
2e-16 ***
## poly(n.age, 3)1                       -1.065e+01  4.222e-01 -25.227  < 
2e-16 ***
## poly(n.age, 3)2                       -3.533e+00  3.197e-01 -11.050  < 
2e-16 ***
## poly(n.age, 3)3                        4.307e-01  2.777e-01   1.551 0.1
21006    
## mileage                               -4.653e-06  1.749e-07 -26.607  < 
2e-16 ***
## tax                                   -1.462e-04  6.179e-05  -2.366 0.0
18015 *  
## log(mpg)                              -8.055e-01  3.617e-02 -22.271  < 
2e-16 ***
## engineSize                             1.960e-01  1.108e-02  17.687  < 
2e-16 ***
## transmissionManual                    -8.642e-02  6.536e-03 -13.222  < 
2e-16 ***
## transmissionSemi-Auto                  1.226e-02  5.429e-03   2.258 0.0
24011 *  
## fuelTypeOther                         -1.416e+00  3.447e-01  -4.106 4.0
9e-05 ***
## fuelTypePetrol                        -4.313e-01  1.143e-01  -3.775 0.0
00162 ***
## manufacturerBMW                       -1.049e+00  1.681e-01  -6.237 4.8
4e-10 ***
```

```
## manufacturerMercedes                -8.502e-01  1.555e-01  -5.469 4.7
7e-08 ***
## manufacturerVW                      -3.065e-01  1.636e-01  -1.873 0.0
61131 .
## poly(n.age, 3)1:manufacturerBMW     -4.087e+00  4.995e-01  -8.181 3.5
6e-16 ***
## poly(n.age, 3)2:manufacturerBMW      3.723e+00  4.405e-01   8.452  <
2e-16 ***
## poly(n.age, 3)3:manufacturerBMW     -1.403e+00  4.173e-01  -3.363 0.0
00778 ***
## poly(n.age, 3)1:manufacturerMercedes -3.312e+00  5.032e-01  -6.582 5.1
4e-11 ***
## poly(n.age, 3)2:manufacturerMercedes  1.878e+00  4.779e-01   3.931 8.5
9e-05 ***
## poly(n.age, 3)3:manufacturerMercedes -8.483e-01  4.056e-01  -2.091 0.0
36538 *
## poly(n.age, 3)1:manufacturerVW      -3.471e-01  4.985e-01  -0.696 0.4
86340
## poly(n.age, 3)2:manufacturerVW       1.230e+00  4.492e-01   2.739 0.0
06187 **
## poly(n.age, 3)3:manufacturerVW       3.877e-01  4.178e-01   0.928 0.3
53479
## tax:fuelTypeOther                   -8.391e-04  4.963e-04  -1.691 0.0
90986 .
## tax:fuelTypePetrol                   3.642e-04  9.714e-05   3.750 0.0
00179 ***
## log(mpg):fuelTypeOther               4.024e-01  7.292e-02   5.519 3.6
0e-08 ***
## log(mpg):fuelTypePetrol              6.430e-02  2.749e-02   2.339 0.0
19373 *
## log(mpg):manufacturerBMW             2.326e-01  3.858e-02   6.029 1.7
7e-09 ***
## log(mpg):manufacturerMercedes        1.949e-01  3.609e-02   5.401 6.9
5e-08 ***
## log(mpg):manufacturerVW             -1.002e-02  3.857e-02  -0.260 0.7
95106
## engineSize:manufacturerBMW           3.652e-02  1.433e-02   2.549 0.0
10825 *
## engineSize:manufacturerMercedes      5.780e-02  1.362e-02   4.244 2.2
4e-05 ***
## engineSize:manufacturerVW            1.006e-01  1.332e-02   7.551 5.1
2e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1458 on 4825 degrees of freedom
## Multiple R-squared:  0.8965, Adjusted R-squared:  0.8958
## F-statistic:  1266 on 33 and 4825 DF,  p-value: < 2.2e-16
```

```
vif( lmFactLogIr )
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
##                                     GVIF Df GVIF^(1/(2*Df))
## poly(n.age, 3)              1.212549e+02  3        2.224760
## mileage                     2.897426e+00  1        1.702183
## tax                         3.074039e+00  1        1.753294
## log(mpg)                    1.572466e+01  1        3.965434
## engineSize                  8.324695e+00  1        2.885255
## transmission                1.723998e+00  2        1.145867
## fuelType                    1.818523e+05  2       20.650458
## manufacturer                2.934133e+08  3       25.777847
## poly(n.age, 3):manufacturer 1.818244e+02  9        1.335169
## tax:fuelType                6.741208e+01  2        2.865395
## log(mpg):fuelType           1.325318e+05  2       19.080074
## log(mpg):manufacturer       1.707402e+08  3       23.553528
## engineSize:manufacturer     1.214166e+04  3        4.794166
```

```
summary(lmFactLogI)
```

```
##
## Call:
## lm(formula = log(price) ~ (poly(n.age, 3) + mileage + tax + log(mpg) +
##     engineSize) * (transmission + fuelType + manufacturer), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46885 -0.09245  0.00102  0.09112  0.61274
##
## Coefficients:
##                                          Estimate Std. Error t value Pr
(>|t|)
## (Intercept)                             1.294e+01  2.088e-01  61.954  <
2e-16
## poly(n.age, 3)1                        -1.220e+01  7.447e-01 -16.378  <
2e-16
## poly(n.age, 3)2                        -2.603e+00  5.028e-01  -5.177 2.
35e-07
## poly(n.age, 3)3                         3.896e-02  5.137e-01   0.076 0.
939550
## mileage                                -4.082e-06  4.777e-07  -8.545  <
2e-16
## tax                                    -3.009e-04  1.390e-04  -2.166 0.
030393
## log(mpg)                               -8.039e-01  4.743e-02 -16.950  <
2e-16
## engineSize                              1.952e-01  1.449e-02  13.468  <
2e-16
## transmissionManual                      4.328e-02  1.770e-01   0.244 0.
806871
## transmissionSemi-Auto                   5.727e-02  1.343e-01   0.427 0.
```

```
669672
## fuelTypeOther                          -1.096e+00  4.035e-01  -2.716 0.
006634
## fuelTypePetrol                         -4.638e-01  1.712e-01  -2.709 0.
006764
## manufacturerBMW                        -1.156e+00  2.046e-01  -5.649 1.
71e-08
## manufacturerMercedes                   -1.001e+00  1.928e-01  -5.192 2.
16e-07
## manufacturerVW                         -2.374e-01  1.952e-01  -1.216 0.
223940
## poly(n.age, 3)1:transmissionManual      1.324e+00  6.810e-01   1.945 0.
051831
## poly(n.age, 3)2:transmissionManual     -1.127e+00  4.675e-01  -2.411 0.
015944
## poly(n.age, 3)3:transmissionManual      2.730e-01  4.940e-01   0.553 0.
580524
## poly(n.age, 3)1:transmissionSemi-Auto  1.115e+00  6.735e-01   1.655 0.
097890
## poly(n.age, 3)2:transmissionSemi-Auto -1.037e+00  5.255e-01  -1.974 0.
048410
## poly(n.age, 3)3:transmissionSemi-Auto  7.191e-01  5.048e-01   1.424 0.
154383
## poly(n.age, 3)1:fuelTypeOther          -2.375e+01  1.361e+01  -1.745 0.
081080
## poly(n.age, 3)2:fuelTypeOther          -2.441e+01  1.992e+01  -1.225 0.
220660
## poly(n.age, 3)3:fuelTypeOther          -7.853e+00  1.032e+01  -0.761 0.
446777
## poly(n.age, 3)1:fuelTypePetrol         -2.805e-01  5.435e-01  -0.516 0.
605728
## poly(n.age, 3)2:fuelTypePetrol          1.718e-01  3.608e-01   0.476 0.
633997
## poly(n.age, 3)3:fuelTypePetrol          2.133e-01  3.315e-01   0.643 0.
520006
## poly(n.age, 3)1:manufacturerBMW        -3.667e+00  7.507e-01  -4.885 1.
07e-06
## poly(n.age, 3)2:manufacturerBMW         3.721e+00  4.602e-01   8.085 7.
80e-16
## poly(n.age, 3)3:manufacturerBMW        -1.599e+00  4.463e-01  -3.582 0.
000344
## poly(n.age, 3)1:manufacturerMercedes   -2.310e+00  7.677e-01  -3.009 0.
002636
## poly(n.age, 3)2:manufacturerMercedes    1.690e+00  5.451e-01   3.100 0.
001944
## poly(n.age, 3)3:manufacturerMercedes   -1.048e+00  5.028e-01  -2.084 0.
037250
## poly(n.age, 3)1:manufacturerVW          2.554e-01  7.063e-01   0.362 0.
717639
## poly(n.age, 3)2:manufacturerVW          1.278e+00  4.669e-01   2.737 0.
```

```
006232
## poly(n.age, 3)3:manufacturerVW         4.137e-01  4.421e-01   0.936 0.
349435
## mileage:transmissionManual            -3.666e-07  4.500e-07  -0.815 0.
415260
## mileage:transmissionSemi-Auto          1.185e-07  4.756e-07   0.249 0.
803327
## mileage:fuelTypeOther                  2.946e-06  1.984e-06   1.484 0.
137770
## mileage:fuelTypePetrol                 4.663e-07  4.037e-07   1.155 0.
248177
## mileage:manufacturerBMW               -4.531e-07  5.029e-07  -0.901 0.
367674
## mileage:manufacturerMercedes          -7.570e-07  5.430e-07  -1.394 0.
163343
## mileage:manufacturerVW                -8.520e-07  4.955e-07  -1.719 0.
085619
## tax:transmissionManual                 4.088e-05  1.333e-04   0.307 0.
759051
## tax:transmissionSemi-Auto              1.168e-04  1.254e-04   0.931 0.
351647
## tax:fuelTypeOther                     -1.546e-03  7.415e-04  -2.085 0.
037110
## tax:fuelTypePetrol                     4.679e-04  1.080e-04   4.331 1.
51e-05
## tax:manufacturerBMW                    1.644e-04  1.520e-04   1.081 0.
279588
## tax:manufacturerMercedes               2.590e-04  1.452e-04   1.784 0.
074456
## tax:manufacturerVW                    -1.651e-04  1.317e-04  -1.254 0.
209936
## log(mpg):transmissionManual           -4.682e-02  4.069e-02  -1.150 0.
250001
## log(mpg):transmissionSemi-Auto        -1.423e-02  3.021e-02  -0.471 0.
637525
## log(mpg):fuelTypeOther                 2.619e-01  9.108e-02   2.876 0.
004052
## log(mpg):fuelTypePetrol                7.214e-02  3.854e-02   1.872 0.
061283
## log(mpg):manufacturerBMW               2.554e-01  4.652e-02   5.489 4.
26e-08
## log(mpg):manufacturerMercedes          2.260e-01  4.416e-02   5.118 3.
21e-07
## log(mpg):manufacturerVW               -1.587e-02  4.567e-02  -0.347 0.
728250
## engineSize:transmissionManual          3.456e-02  1.606e-02   2.153 0.
031408
## engineSize:transmissionSemi-Auto      -3.376e-03  1.162e-02  -0.291 0.
771331
## engineSize:fuelTypeOther               9.506e-02  7.122e-02   1.335 0.
```

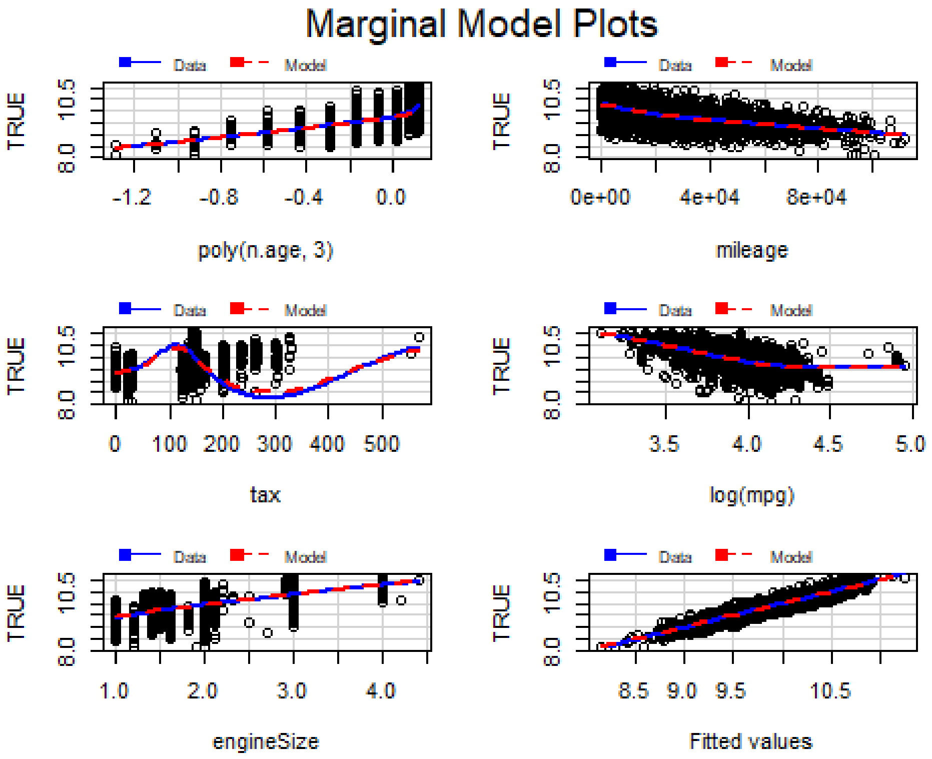
```
182045
## engineSize:fuelTypePetrol             -1.029e-02  1.322e-02  -0.778 0.
436334
## engineSize:manufacturerBMW             4.059e-02  1.467e-02   2.767 0.
005685
## engineSize:manufacturerMercedes        6.416e-02  1.399e-02   4.585 4.
66e-06
## engineSize:manufacturerVW              9.687e-02  1.448e-02   6.688 2.
52e-11
##
## (Intercept)                           ***
## poly(n.age, 3)1                       ***
## poly(n.age, 3)2                       ***
## poly(n.age, 3)3
## mileage                               ***
## tax                                   *
## log(mpg)                              ***
## engineSize                            ***
## transmissionManual
## transmissionSemi-Auto
## fuelTypeOther                         **
## fuelTypePetrol                        **
## manufacturerBMW                       ***
## manufacturerMercedes                  ***
## manufacturerVW
## poly(n.age, 3)1:transmissionManual    .
## poly(n.age, 3)2:transmissionManual    *
## poly(n.age, 3)3:transmissionManual
## poly(n.age, 3)1:transmissionSemi-Auto .
## poly(n.age, 3)2:transmissionSemi-Auto *
## poly(n.age, 3)3:transmissionSemi-Auto
## poly(n.age, 3)1:fuelTypeOther         .
## poly(n.age, 3)2:fuelTypeOther
## poly(n.age, 3)3:fuelTypeOther
## poly(n.age, 3)1:fuelTypePetrol
## poly(n.age, 3)2:fuelTypePetrol
## poly(n.age, 3)3:fuelTypePetrol
## poly(n.age, 3)1:manufacturerBMW       ***
## poly(n.age, 3)2:manufacturerBMW       ***
## poly(n.age, 3)3:manufacturerBMW       ***
## poly(n.age, 3)1:manufacturerMercedes  **
## poly(n.age, 3)2:manufacturerMercedes  **
## poly(n.age, 3)3:manufacturerMercedes  *
## poly(n.age, 3)1:manufacturerVW
## poly(n.age, 3)2:manufacturerVW        **
## poly(n.age, 3)3:manufacturerVW
## mileage:transmissionManual
## mileage:transmissionSemi-Auto
## mileage:fuelTypeOther
## mileage:fuelTypePetrol
## mileage:manufacturerBMW
## mileage:manufacturerMercedes
## mileage:manufacturerVW                .
## tax:transmissionManual
## tax:transmissionSemi-Auto
## tax:fuelTypeOther                     *
## tax:fuelTypePetrol                    ***
## tax:manufacturerBMW
## tax:manufacturerMercedes              .
## tax:manufacturerVW
## log(mpg):transmissionManual
## log(mpg):transmissionSemi-Auto
## log(mpg):fuelTypeOther                **
## log(mpg):fuelTypePetrol               .
## log(mpg):manufacturerBMW              ***
## log(mpg):manufacturerMercedes         ***
## log(mpg):manufacturerVW
## engineSize:transmissionManual         *
## engineSize:transmissionSemi-Auto
## engineSize:fuelTypeOther
## engineSize:fuelTypePetrol
## engineSize:manufacturerBMW            **
## engineSize:manufacturerMercedes       ***
## engineSize:manufacturerVW             ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1453 on 4795 degrees of freedom
## Multiple R-squared:  0.8978, Adjusted R-squared:  0.8965
## F-statistic: 668.6 on 63 and 4795 DF,  p-value: < 2.2e-16
```

```
anova( lmFactLogIr, lmFactLogI)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ poly(n.age, 3) + mileage + tax + log(mpg) + engi
neSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     tax:fuelType + log(mpg):fuelType + log(mpg):manufacturer +
##     engineSize:manufacturer
## Model 2: log(price) ~ (poly(n.age, 3) + mileage + tax + log(mpg) + eng
ineSize) *
##     (transmission + fuelType + manufacturer)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   4825 102.54
## 2   4795 101.24 30    1.3003 2.0529 0.0006261 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
marginalModelPlots( lmFactLogIr )
```

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

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



```
AIC(lmFactLog,lmFactLogI,lmFactLogIr)

##             df       AIC
## lmFactLog   16 -4509.050
## lmFactLogI  65 -4890.545
## lmFactLogIr 35 -4888.535
```

Select the best model available so far. Interpret the equations that relate the explanatory variables to the answer (price).

From the AIC we find that the model with all factors, all numeric variables and some logarithmic transformations have the best criterion.

In order to interpret the model for each of the variables a plot effects is undertaken.

Now, some graphical representations of the behaviour of the variables in the model are investigated and looked in to. First, an added variable plot is obtained using the avPlots-function. This plot represents for each predictor in the model, the actual behaviour of the response variable by keeping the influence of the other explanatory variables constant. This scatterplot matrix clearly represents the linear relationships between the explanatory variables and response variable. However, to further look for monotone linear relationsships that might benefit from a transformation, the crPlots funtion is used, which is a Partial-Residual plot. These are partial regressions and are used to distinguish between monotone linearity (which might benefit from a transformation) and non-monotone linearity (which don't). From these plots however, no clear cases of monotone linearity are found. However, from the av-plot showing mpg vs the prices and petrol vs prices, we see that this variable is heavily influenced by a few values with high leverage.

```
# avPlots( lmFactLogIr )
lmBest= lm(log(price) ~ poly(n.age, 3) + mileage + log(mpg) +
    engineSize + transmission + fuelType + manufacturer + poly(n.age,
    3):fuelType + poly(n.age, 3):manufacturer + log(mpg):fuelType +
    engineSize:manufacturer, data = df)
summary(lmBest)

##
## Call:
## lm(formula = log(price) ~ poly(n.age, 3) + mileage + log(mpg) +
##     engineSize + transmission + fuelType + manufacturer + poly(n.age,
##     3):fuelType + poly(n.age, 3):manufacturer + log(mpg):fuelType +
##     engineSize:manufacturer, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48672 -0.09225  0.00297  0.09506  0.54598
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(
>|t|)
## (Intercept)                             1.227e+01  8.712e-02 140.854  <
2e-16 ***
## poly(n.age, 3)1                        -1.184e+01  4.191e-01 -28.240  <
2e-16 ***
## poly(n.age, 3)2                        -2.958e+00  3.367e-01  -8.785  <
2e-16 ***
## poly(n.age, 3)3                         1.571e-01  3.171e-01   0.495  0.
62030
## mileage                                -4.654e-06  1.788e-07 -26.035  <
2e-16 ***
## log(mpg)                               -6.510e-01  1.952e-02 -33.346  <
2e-16 ***
## engineSize                              2.180e-01  1.005e-02  21.681  <
2e-16 ***
## transmissionManual                     -9.233e-02  6.490e-03 -14.226  <
2e-16 ***
## transmissionSemi-Auto                   1.097e-02  5.471e-03   2.006  0.
04492 *
## fuelTypeOther                          -1.678e+00  3.024e-01  -5.551 2.9
9e-08 ***
## fuelTypePetrol                         -1.008e-01  9.811e-02  -1.028  0.
```

```
30408
## manufacturerBMW                        -3.624e-02  2.601e-02  -1.393  0.
16368
## manufacturerMercedes                   -1.040e-02  2.490e-02  -0.417  0.
67640
## manufacturerVW                         -3.449e-01  2.160e-02 -15.967  <
2e-16 ***
## poly(n.age, 3)1:fuelTypeOther          -1.850e+01  1.228e+01  -1.507  0.
13195
## poly(n.age, 3)2:fuelTypeOther          -3.146e+01  1.841e+01  -1.709  0.
08754 .
## poly(n.age, 3)3:fuelTypeOther          -1.332e+01  9.429e+00  -1.413  0.
15786
## poly(n.age, 3)1:fuelTypePetrol          6.839e-01  3.460e-01   1.977  0.
04812 *
## poly(n.age, 3)2:fuelTypePetrol          1.310e-01  3.414e-01   0.384  0.
70118
## poly(n.age, 3)3:fuelTypePetrol          4.058e-01  3.262e-01   1.244  0.
21356
## poly(n.age, 3)1:manufacturerBMW        -2.367e+00  4.342e-01  -5.452 5.2
4e-08 ***
## poly(n.age, 3)2:manufacturerBMW         2.796e+00  4.211e-01   6.640 3.4
7e-11 ***
## poly(n.age, 3)3:manufacturerBMW        -1.202e+00  4.273e-01  -2.814  0.
00491 **
## poly(n.age, 3)1:manufacturerMercedes -1.791e+00  4.420e-01  -4.053 5.1
3e-05 ***
## poly(n.age, 3)2:manufacturerMercedes  8.792e-01  4.490e-01   1.958  0.
05025 .
## poly(n.age, 3)3:manufacturerMercedes -7.027e-01  4.147e-01  -1.695  0.
09022 .
## poly(n.age, 3)1:manufacturerVW         -4.047e-01  4.131e-01  -0.980  0.
32735
## poly(n.age, 3)2:manufacturerVW          1.244e+00  4.135e-01   3.009  0.
00263 **
## poly(n.age, 3)3:manufacturerVW          5.046e-01  4.218e-01   1.196  0.
23165
## log(mpg):fuelTypeOther                  4.295e-01  7.653e-02   5.612 2.1
1e-08 ***
## log(mpg):fuelTypePetrol                -7.641e-03  2.534e-02  -0.301  0.
76307
## engineSize:manufacturerBMW             -1.304e-02  1.226e-02  -1.063  0.
28764
## engineSize:manufacturerMercedes         1.862e-02  1.207e-02   1.542  0.
12309
## engineSize:manufacturerVW               9.946e-02  1.181e-02   8.421  <
2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1469 on 4825 degrees of freedom
## Multiple R-squared:  0.8949, Adjusted R-squared:  0.8942
## F-statistic:  1245 on 33 and 4825 DF,  p-value: < 2.2e-16
```

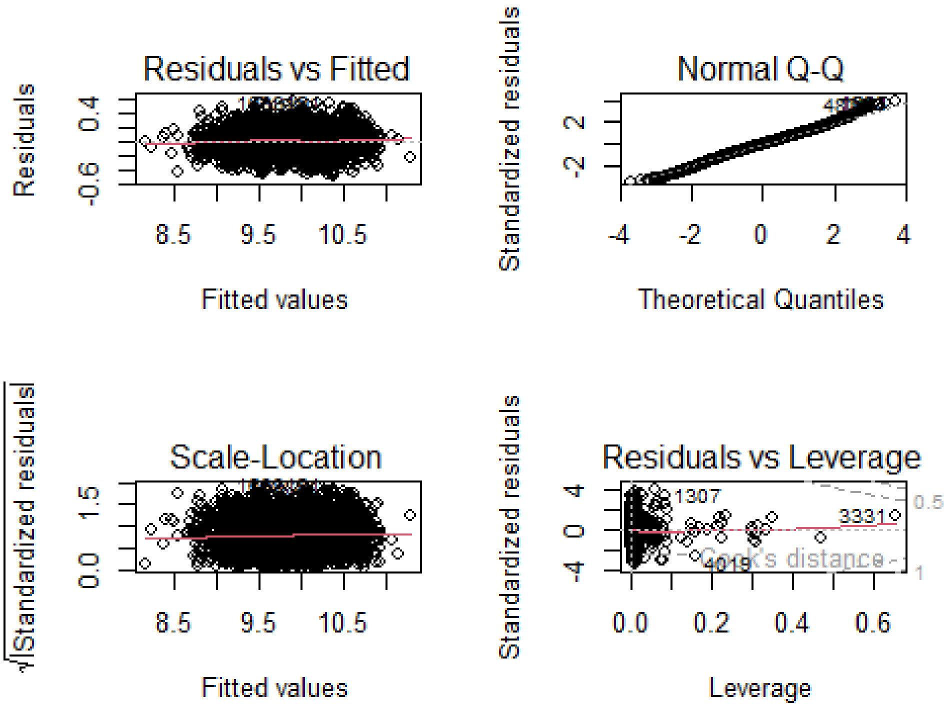
Graphically assess the best model obtained so far.

The residuals vs Fitted plot shows that the residuals follow a good linear pattern, which meets the regression assumptions very well. The Normal Q-Q plot shows that the standard errors are mostly normally distributed with a small amount of deviating prediction at the upper end of the tail. The scale-location plot shows that homoscedasticity is satisfied as a straight line is obtained. The residuals vs leverage plot shows that our model includes some high-leverage (so highly a priori influential in our model) points that deviate significantly (4 standardized residuals away) and asymmetrically from the prediction. An influence plot further confirms this believe.

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

##
## Call:
## lm(formula = log(price) ~ poly(n.age, 3) + mileage + log(mpg) +
##     engineSize + transmission + fuelType + manufacturer + poly(n.age,
##     3):fuelType + poly(n.age, 3):manufacturer + log(mpg):fuelType +
##     engineSize:manufacturer, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48672 -0.09225  0.00297  0.09506  0.54598
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(
>|t|)
## (Intercept)                             1.227e+01  8.712e-02 140.854  <
2e-16 ***
## poly(n.age, 3)1                        -1.184e+01  4.191e-01 -28.240  <
2e-16 ***
## poly(n.age, 3)2                        -2.958e+00  3.367e-01  -8.785  <
2e-16 ***
## poly(n.age, 3)3                         1.571e-01  3.171e-01   0.495  0.
62030
## mileage                                -4.654e-06  1.788e-07 -26.035  <
2e-16 ***
## log(mpg)                               -6.510e-01  1.952e-02 -33.346  <
2e-16 ***
## engineSize                              2.180e-01  1.005e-02  21.681  <
2e-16 ***
## transmissionManual                     -9.233e-02  6.490e-03 -14.226  <
2e-16 ***
## transmissionSemi-Auto                   1.097e-02  5.471e-03   2.006  0.
04492 *
```

```
## fuelTypeOther                         -1.678e+00  3.024e-01  -5.551 2.9
9e-08 ***
## fuelTypePetrol                        -1.008e-01  9.811e-02  -1.028  0.
30408
## manufacturerBMW                       -3.624e-02  2.601e-02  -1.393  0.
16368
## manufacturerMercedes                  -1.040e-02  2.490e-02  -0.417  0.
67640
## manufacturerVW                        -3.449e-01  2.160e-02 -15.967  <
2e-16 ***
## poly(n.age, 3)1:fuelTypeOther         -1.850e+01  1.228e+01  -1.507  0.
13195
## poly(n.age, 3)2:fuelTypeOther         -3.146e+01  1.841e+01  -1.709  0.
08754 .
## poly(n.age, 3)3:fuelTypeOther         -1.332e+01  9.429e+00  -1.413  0.
15786
## poly(n.age, 3)1:fuelTypePetrol         6.839e-01  3.460e-01   1.977  0.
04812 *
## poly(n.age, 3)2:fuelTypePetrol         1.310e-01  3.414e-01   0.384  0.
70118
## poly(n.age, 3)3:fuelTypePetrol         4.058e-01  3.262e-01   1.244  0.
21356
## poly(n.age, 3)1:manufacturerBMW       -2.367e+00  4.342e-01  -5.452 5.2
4e-08 ***
## poly(n.age, 3)2:manufacturerBMW        2.796e+00  4.211e-01   6.640 3.4
7e-11 ***
## poly(n.age, 3)3:manufacturerBMW       -1.202e+00  4.273e-01  -2.814  0.
00491 **
## poly(n.age, 3)1:manufacturerMercedes -1.791e+00  4.420e-01  -4.053 5.1
3e-05 ***
## poly(n.age, 3)2:manufacturerMercedes  8.792e-01  4.490e-01   1.958  0.
05025 .
## poly(n.age, 3)3:manufacturerMercedes -7.027e-01  4.147e-01  -1.695  0.
09022 .
## poly(n.age, 3)1:manufacturerVW        -4.047e-01  4.131e-01  -0.980  0.
32735
## poly(n.age, 3)2:manufacturerVW         1.244e+00  4.135e-01   3.009  0.
00263 **
## poly(n.age, 3)3:manufacturerVW         5.046e-01  4.218e-01   1.196  0.
23165
## log(mpg):fuelTypeOther                 4.295e-01  7.653e-02   5.612 2.1
1e-08 ***
## log(mpg):fuelTypePetrol               -7.641e-03  2.534e-02  -0.301  0.
76307
## engineSize:manufacturerBMW            -1.304e-02  1.226e-02  -1.063  0.
28764
## engineSize:manufacturerMercedes        1.862e-02  1.207e-02   1.542  0.
12309
## engineSize:manufacturerVW              9.946e-02  1.181e-02   8.421  <
2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1469 on 4825 degrees of freedom
## Multiple R-squared:  0.8949, Adjusted R-squared:  0.8942
## F-statistic:  1245 on 33 and 4825 DF,  p-value: < 2.2e-16

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



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

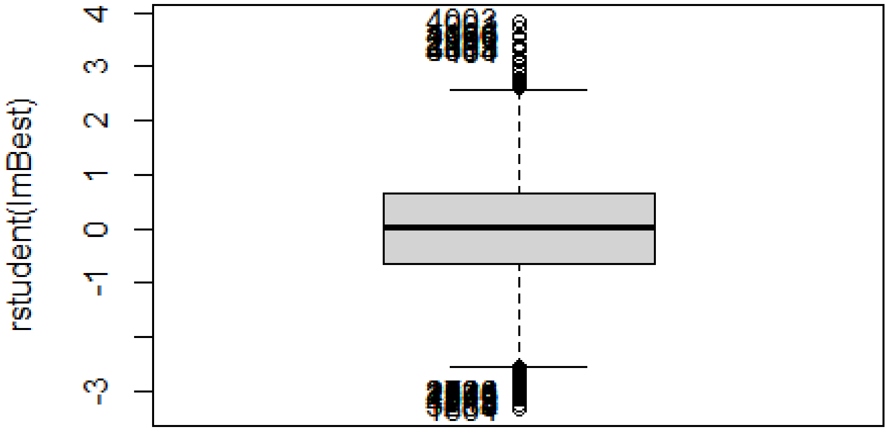
Assess the presence of outliers in the studentized residuals at a 99% confidence level. Indicate what those observations are.

First, a histogram is plotted to make sure the residuals follow a nice and smooth normal distribution, which they do. Then the studentized residuals at the 99% CI are calculated, 72 residual outliers are found. The boxplot and plot of the residuals show which values are considered outliers. Interestingly, the plot of residuals shows many residuals grouped tightly together in the right bottom of the plot. Out of curiosity, these will be inspected more in depth a little later. First, the cooks distance is plotted, together with the outliers crossed out.

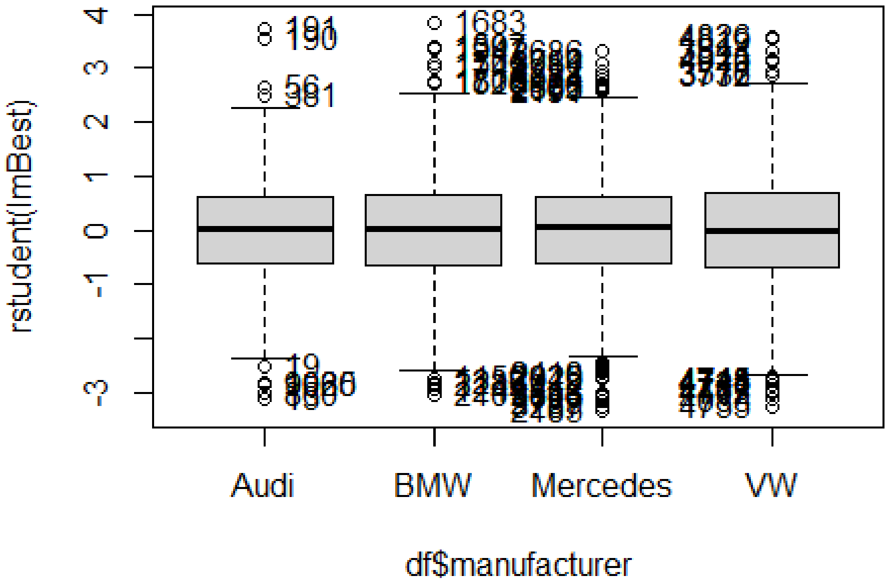
From the summary, it can be observed that many outliers are Volkswagens. Using a boxplot the prices of the residual outliers vs the manufacturer are plotted to see if Volkswagen deviates from the others, which it is not the case.

Lastly, all prices are plotted, together with the prices of the outliers per manufacturer. From this it seems that the earlier observed clustered group are all volkswagens. When manually looking at these observations it appears that these many belong to a specific model line named “Up”. Weirdly, it appears that not all “Up” models are found to be outliers, although many of them have a similar price.

```
# Residual outliers
qnorm(0.995)

## [1] 2.575829

llres <- which( abs(rstudent(lmBest))>qnorm(0.995));length(llres)

## [1] 72

Boxplot(rstudent(lmBest))
```

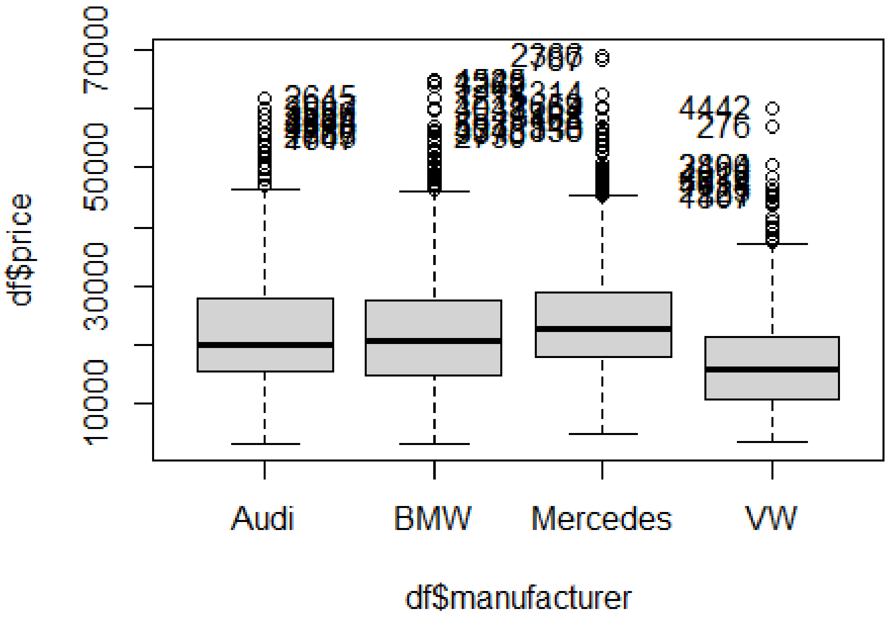


```
##   [1] 1664 1285 3234 1813 1213 2872 4544 1706 2044  223 4003  901 4166
1030 2999
## [16] 1155 2802 3013 4331  464

table(df$manufacturer[llres])

##
##     Audi      BMW Mercedes       VW
##        8       15       21       28

Boxplot(rstudent(lmBest)~df$manufacturer)
```



```
##  [1] "860"  "19"   "906"  "13"   "1030" "1025" "191"  "190"  "56"   "3
81"
## [11] "2107" "1280" "1244" "1159" "2147" "1597" "1818" "1501" "1307" "1
947"
## [21] "1749" "1683" "1526" "1705" "2185" "3287" "2789" "3006" "2844" "3
418"
## [31] "2945" "2622" "2329" "2418" "3196" "2710" "2686" "2662" "3354" "2
191"
## [41] "2565" "3054" "2974" "2291" "4735" "4083" "4705" "4472" "4717" "4
721"
## [51] "4743" "4715" "4744" "4748" "3777" "4825" "3732" "4819" "3516" "4
910"
## [61] "4826" "3544"
```

```
Boxplot(df$price~df$manufacturer)
```



```
##  [1] "2645" "3007" "3550" "4284" "3143" "166"  "4573" "4850" "2765" "1
647"
## [11] "1525" "4348" "3262" "1215" "4041" "3038" "593"  "4043" "3378" "2
730"
## [21] "2366" "787"  "1314" "2659" "64"   "3188" "464"  "345"  "130"  "8
36"
## [31] "4442" "276"  "3490" "2864" "4010" "4718" "614"  "2225" "151"  "1
307"
```

Study the presence of a priori influential data observations, indicating their number according to the criteria studied in class.

Therefore, this is shown in the following segment. In this case, 88 a priori values where found, assuming that the dataset is large enough to use hat>4*mean(hat) instead of multiplying with 2 or 3.

```
mean_hat <- mean(hatvalues(lmBest));mean_hat

## [1] 0.006997325

priori <- which(hatvalues(lmBest)>4*mean_hat)
length(priori)

## [1] 88
```

Study the presence of a posteriori influential values, indicating the criteria studied in class and the actual atypical observations.

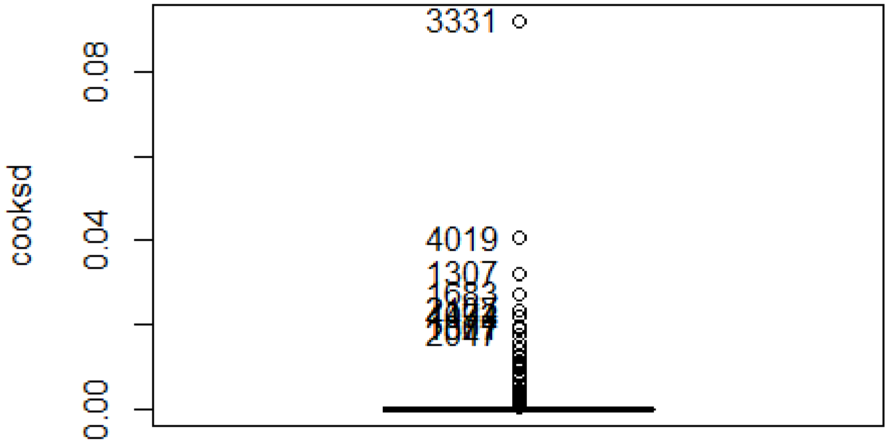
A posteriori influential values are found using the dfbetas function. These are tried to be firstly plotted against their cut-off value, which is given by 2/sqrt(n). Then for each variable in the model, it is tested which observations are considered as a posteriori influential values.

Taking a look to cook's distance outliers, 3 influent data are found and are removed from the sample.

The best model is then reconstructed on this new data set and compared with its original to see the difference. From the summary, it can be observed that the change mostly affected the coefficients of the tax, tranmission levels and manufacturer levels. The R-squared of the model has increased up to 0.91.

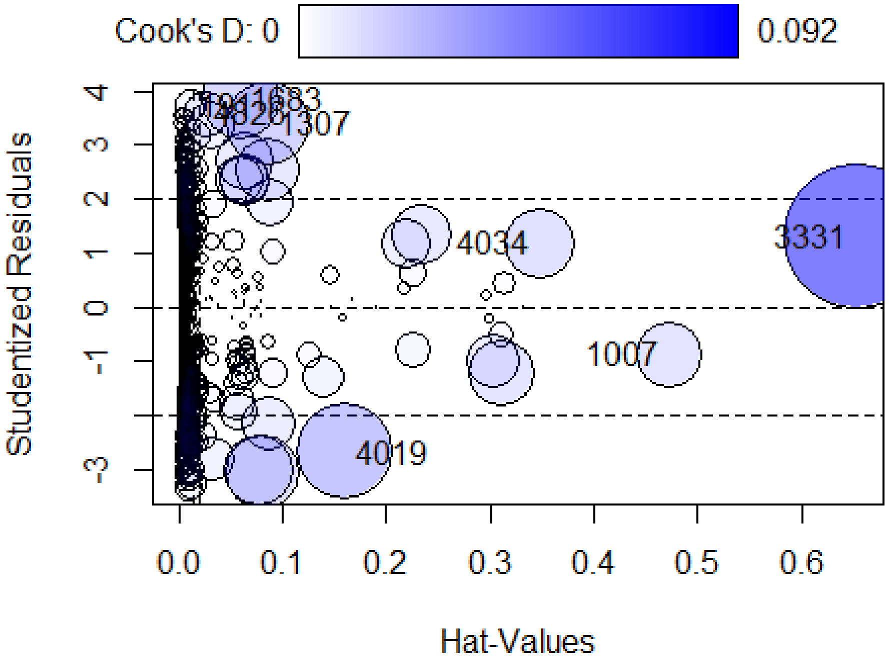
Graphically, it can be observed that indeed several high leverage observations where taken out, which improves the model. The process should be repeated again, since residual outliers and influent data are still present.

```
betas <- as.data.frame(dfbetas(lmBest))
betas_cutoff = 2 / sqrt(dim(df)[1])
betas_cutoff

## [1] 0.02869172

# par(mfrow=c(1,1))
# matplot(betas, type="l", lwd=2, col= rainbow(ncol(betas)))
# lines(sqrt(cooks.distance(lmBest)), col=4, lwd=3)
# abline(h=betas_cutoff, lty=3,lwd=1,col=1)
# abline(h=-betas_cutoff[1], lty=3,lwd=1,col=1)
# legend("bottomleft", legend=names(coef(lmBest)),
#        col=rainbow(ncol(betas)), lty=1:2, cex=0.8, ncol=2) # Nothing ca
n be seen

cooksd <- cooks.distance(lmBest)
Boxplot(cooksd, id=list(labels=rownames(df)))
```



```
##  [1] "3331" "4019" "1307" "1683" "2107" "4472" "4034" "1007" "1371" "2
047"
```

```
influencePlot(lmBest, id=list(n=3, method="noteworthy"))
```



```
##          StudRes        Hat       CookD
## 3331  1.2896943 0.65278648 0.091962226
## 1007 -0.8603855 0.47125530 0.019406216
## 4019 -2.6792135 0.16115085 0.040506874
## 191   3.7248431 0.01144269 0.004710922
## 1307  3.3923830 0.08661370 0.032027135
## 4034  1.1868242 0.34697372 0.022010150
## 1683  3.8382102 0.05953618 0.027351633
## 4826  3.5757731 0.02532999 0.009749420
```

```
llcoo <-c("3331","1007","4034")
rownames(df)[llres]
```

```
##  [1] "4706" "3196" "4743" "860"  "2547" "2710" "4760" "4737" "2686" "4
598"
## [11] "4019" "4052" "2622" "191"  "2662" "190"  "4695" "4705" "1113" "1
597"
## [21] "4083" "3777" "3287" "2329" "3354" "4744" "2418" "2185" "2844" "2
789"
## [31] "4825" "906"  "2107" "4723" "2306" "1280" "3732" "1818" "2565" "3
054"
## [41] "3418" "1501" "1307" "4721" "4824" "13"   "4819" "1947" "1244" "3
516"
## [51] "2974" "4735" "1749" "4748" "4742" "1030" "4910" "1159" "56"   "1
683"
## [61] "1526" "2291" "2945" "4826" "4472" "3544" "2147" "4715" "3006" "1
705"
## [71] "4717" "1025"
```

```
llrem <- unique(c(rownames(df)[llres],llcoo)); length(llrem)
```

```
## [1] 75
```

```
llremreg<-which(rownames(df)%in%llrem);llremreg
```

```
##  [1]   58  194  202  223  273  281  324  407  447  464  532  543  559
579  667
## [16]  901 1008 1030 1045 1113 1143 1155 1213 1254 1285 1330 1405 1650
1654 1664
## [31] 1706 1813 1843 1928 2044 2045 2170 2185 2351 2511 2590 2659 2664
2678 2802
## [46] 2810 2864 2872 2999 3013 3084 3115 3155 3163 3234 3262 3316 3341
3391 3425
## [61] 3427 3676 4003 4041 4079 4157 4166 4246 4331 4409 4487 4544 4560
4644 4796
```

```
df<-df[-llremreg,]
```

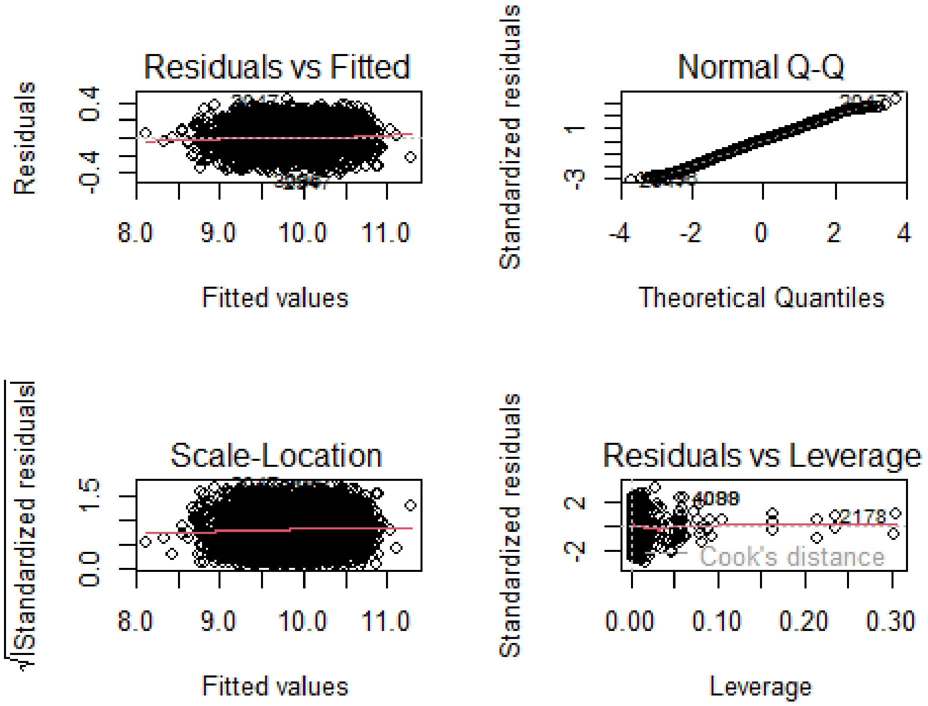
```
lmBestp= lm(log(price) ~ poly(n.age, 3) + mileage + log(mpg) +
    engineSize + transmission + fuelType + manufacturer + poly(n.age,
    3):fuelType + poly(n.age, 3):manufacturer + log(mpg):fuelType +
    engineSize:manufacturer, data = df)
summary(lmBestp)

##
## Call:
## lm(formula = log(price) ~ poly(n.age, 3) + mileage + log(mpg) +
##     engineSize + transmission + fuelType + manufacturer + poly(n.age,
##     3):fuelType + poly(n.age, 3):manufacturer + log(mpg):fuelType +
##     engineSize:manufacturer, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41503 -0.08899  0.00221  0.09223  0.37666
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(
>|t|)
## (Intercept)                        1.237e+01  8.335e-02 148.359  <
2e-16 ***
## poly(n.age, 3)1                   -1.110e+01  3.942e-01 -28.148  <
2e-16 ***
## poly(n.age, 3)2                   -2.905e+00  3.244e-01  -8.955  <
2e-16 ***
## poly(n.age, 3)3                    6.288e-01  3.363e-01   1.869  0.
06162 .
## mileage                           -4.597e-06  1.685e-07 -27.274  <
2e-16 ***
## log(mpg)                          -6.724e-01  1.864e-02 -36.077  <
2e-16 ***
## engineSize                         2.133e-01  9.559e-03  22.315  <
2e-16 ***
## transmissionManual                -9.057e-02  6.134e-03 -14.765  <
2e-16 ***
## transmissionSemi-Auto              1.182e-02  5.182e-03   2.280  0.
02263 *
## fuelTypeOther                     -1.548e+00  3.111e-01  -4.977 6.7
0e-07 ***
## fuelTypePetrol                    -1.571e-02  9.348e-02  -0.168  0.
86653
## manufacturerBMW                   -2.330e-02  2.454e-02  -0.950  0.
34235
## manufacturerMercedes               2.566e-02  2.355e-02   1.090  0.
27597
## manufacturerVW                    -3.127e-01  2.042e-02 -15.318  <
2e-16 ***
## poly(n.age, 3)1:fuelTypeOther     -1.729e+01  1.041e+01  -1.660  0.
09698 .
## poly(n.age, 3)2:fuelTypeOther     -3.093e+01  1.582e+01  -1.955  0.
05059 .
## poly(n.age, 3)3:fuelTypeOther     -1.307e+01  8.184e+00  -1.597  0.
```

```
11028
## poly(n.age, 3)1:fuelTypePetrol        3.877e-01  3.272e-01   1.185  0.
23604
## poly(n.age, 3)2:fuelTypePetrol        8.104e-02  3.192e-01   0.254  0.
79961
## poly(n.age, 3)3:fuelTypePetrol       -6.313e-03  3.084e-01  -0.020  0.
98367
## poly(n.age, 3)1:manufacturerBMW      -2.755e+00  4.083e-01  -6.747 1.6
9e-11 ***
## poly(n.age, 3)2:manufacturerBMW       2.756e+00  3.960e-01   6.961 3.8
4e-12 ***
## poly(n.age, 3)3:manufacturerBMW      -1.227e+00  4.008e-01  -3.062  0.
00221 **
## poly(n.age, 3)1:manufacturerMercedes -2.193e+00  4.213e-01  -5.206 2.0
1e-07 ***
## poly(n.age, 3)2:manufacturerMercedes  7.362e-01  4.559e-01   1.615  0.
10643
## poly(n.age, 3)3:manufacturerMercedes -1.386e+00  4.469e-01  -3.100  0.
00195 **
## poly(n.age, 3)1:manufacturerVW       -1.102e+00  3.887e-01  -2.835  0.
00460 **
## poly(n.age, 3)2:manufacturerVW        7.520e-01  3.892e-01   1.932  0.
05341 .
## poly(n.age, 3)3:manufacturerVW       -3.738e-02  4.013e-01  -0.093  0.
92578
## log(mpg):fuelTypeOther                3.963e-01  7.823e-02   5.066 4.2
2e-07 ***
## log(mpg):fuelTypePetrol              -3.139e-02  2.417e-02  -1.299  0.
19404
## engineSize:manufacturerBMW           -1.952e-02  1.157e-02  -1.687  0.
09168 .
## engineSize:manufacturerMercedes       1.585e-03  1.142e-02   0.139  0.
88964
## engineSize:manufacturerVW             8.057e-02  1.117e-02   7.215 6.2
5e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1378 on 4750 degrees of freedom
## Multiple R-squared:  0.9054, Adjusted R-squared:  0.9047
## F-statistic:  1377 on 33 and 4750 DF,  p-value: < 2.2e-16
```

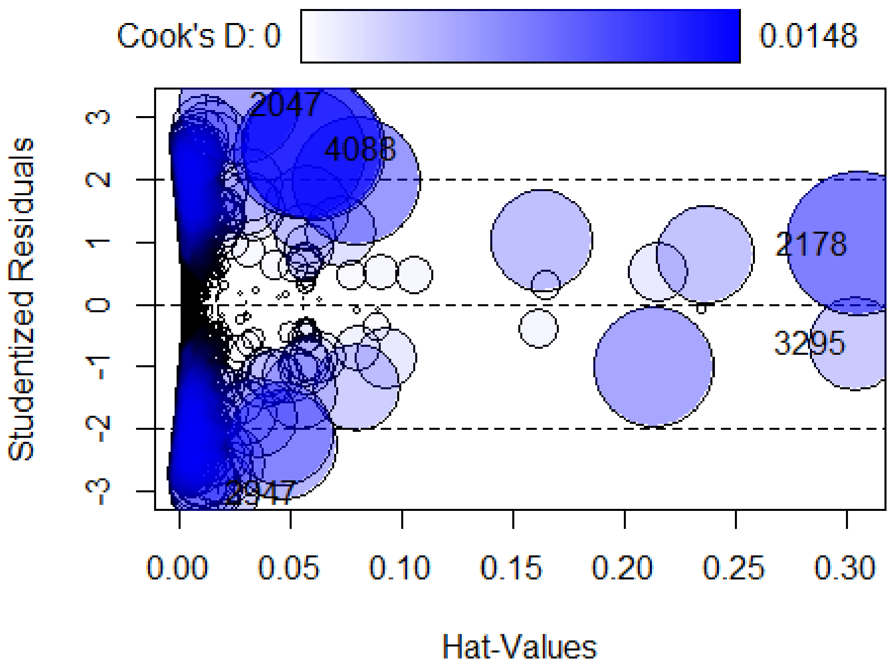
```
lmBestp= step( lmBestp, k=log(nrow(df)))
```

```
## Start:  AIC=-18706.91
## log(price) ~ poly(n.age, 3) + mileage + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):fuelType +
##     poly(n.age, 3):manufacturer + log(mpg):fuelType + engineSize:manuf
acturer
##
```

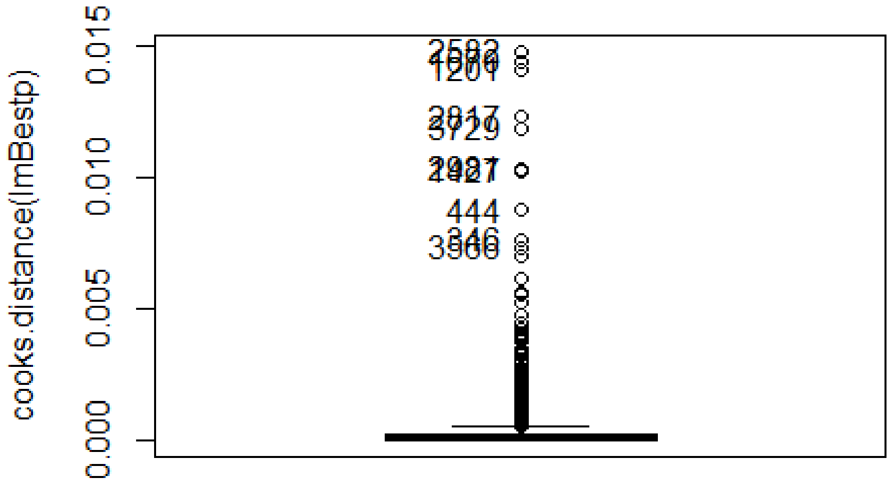
```
##                                 Df Sum of Sq     RSS    AIC
## - poly(n.age, 3):fuelType        6    0.1319  90.375 -18751
## <none>                                        90.243 -18707
## - log(mpg):fuelType              2    0.5433  90.786 -18695
## - poly(n.age, 3):manufacturer    9    2.0590  92.302 -18675
## - engineSize:manufacturer        3    1.5574  91.800 -18651
## - transmission                   2    6.7009  96.944 -18381
## - mileage                        1   14.1322 104.375 -18019
##
## Step:  AIC=-18750.76
## log(price) ~ poly(n.age, 3) + mileage + log(mpg) + engineSize +
##     transmission + fuelType + manufacturer + poly(n.age, 3):manufactur
er +
##     log(mpg):fuelType + engineSize:manufacturer
##
##                                 Df Sum of Sq     RSS    AIC
## <none>                                        90.375 -18751
## - poly(n.age, 3):manufacturer    9    2.0869  92.462 -18718
## - engineSize:manufacturer        3    1.5715  91.946 -18694
## - log(mpg):fuelType              2    2.7151  93.090 -18626
## - transmission                   2    6.7003  97.075 -18426
## - mileage                        1   14.8988 105.274 -18029

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



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

influencePlot( lmBestp )
```

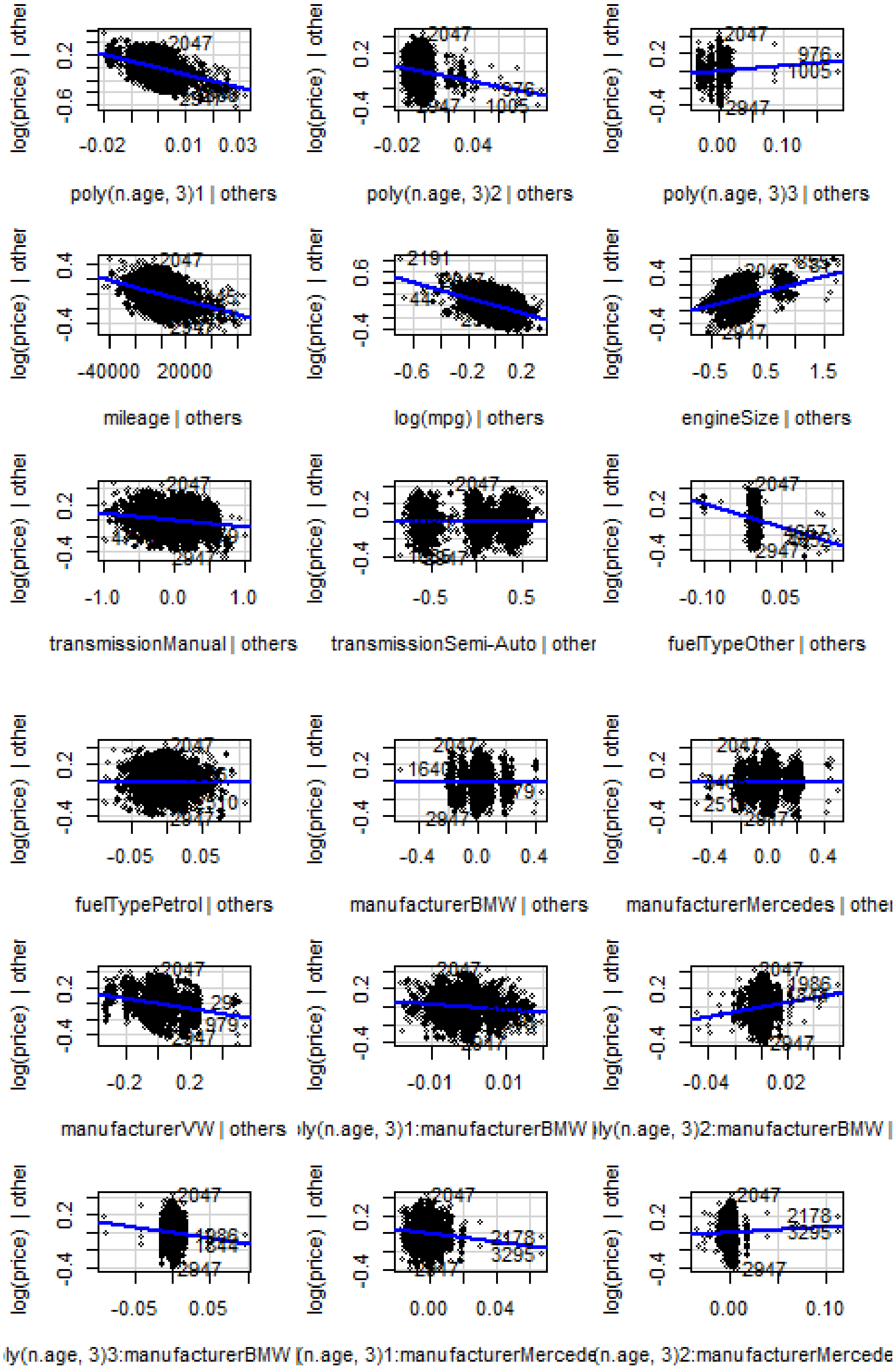


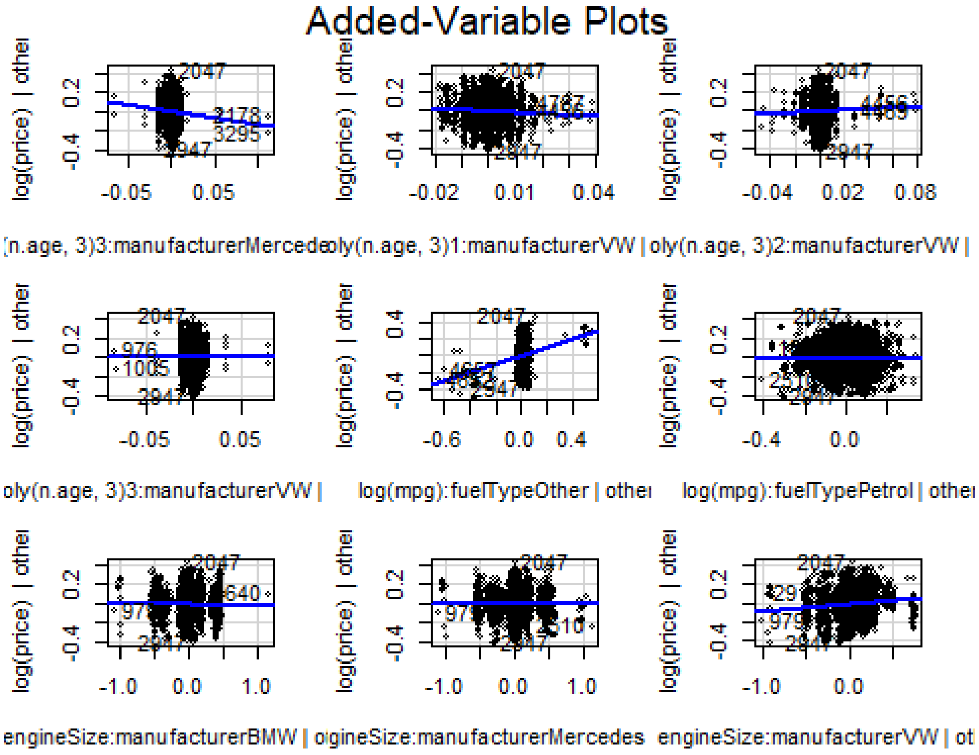
```
##         StudRes        Hat       CookD
## 3295 -0.6290577 0.30368413 0.006164444
## 4088  2.4932366 0.06095096 0.014394121
## 2178  0.9708186 0.30504474 0.014775089
## 2047  3.2026870 0.02735762 0.010283752
## 2947 -3.0325065 0.01648295 0.005494786

Boxplot( cooks.distance(lmBestp) )
```



```
##  [1] 2582 1670 1201 2817 3729 2981 1427  444  346 3566
```

```
avPlots(lmBestp)
```


log(price) | other
poly(n.age, 3)1 | others
-0.02
0.01
0.03
0.2
-0.6
2047
poly(n.age, 3)2 | others
-0.02
0.04
0.2
-0.4
2047
976
1005
poly(n.age, 3)3 | others
0.00
0.10
976
1005
2947
mileage | others
-40000
20000
0.4
-0.4
log(mpg) | others
-0.6
-0.2
0.2
0.6
2191
engineSize | others
-0.5
0.5
1.5
transmissionManual | others
-1.0
0.0
1.0
transmissionSemi-Auto | other
-0.5
0.5
fuelTypeOther | others
-0.10
0.05
fuelTypePetrol | others
-0.05
0.05
manufacturerBMW | others
-0.4
0.0
0.4
1640
manufacturerMercedes | other
manufacturerVW | others
-0.2
0.2
979
ly(n.age, 3)1:manufacturerBMW
-0.01
0.01
ly(n.age, 3)2:manufacturerBMW |
-0.04
0.02
1986
ly(n.age, 3)3:manufacturerBMW |
-0.05
0.05
1844
(n.age, 3)1:manufacturerMercede
0.00
0.04
2178
3295
(n.age, 3)2:manufacturerMercede
0.00
0.10



Given a 5-year old car, the rest of numerical variables on the mean and factors on the reference level, what would be the expected price with a 95% confidence interval?

Reference factors are transmission = Automatic, fuelType = Diesel and manufacturer = Audi. Using the best model, the price is expected to lie in interval [13911.44, 23919.11] with a confidence of 95 %.

```
sample <- data.frame(n.age=5, tax=mean(df$tax), mileage=mean(df$mileage),
engineSize=mean(df$engineSize), mpg=mean(df$mpg), transmission="Automatic
", fuelType="Diesel", manufacturer="Audi")

sam.fit <- predict.lm(lmBestp, sample, se.fit=TRUE, interval="prediction"
, level=0.95)

exp(sam.fit$fit)

##        fit      lwr      upr
## 1 18241.21 13911.67 23918.17
```

Summarize what you have learned by working with this interesting real dataset.

This project is a clear example of how a data set that at first doesn't look that complex requires a lot of careful preprocessing, transformation, analysis and constant re-analysis.

The preprocessing part and exploratory analysis are components we were already a bit more familiar with and thus were carried out rather quickly. However, at first, multivariate outliers weren't accounted for, such that are initial results were quite different and more imbalanced than after these were removed from the data set. This showcased the importance of careful preprocessing in the sense that skipping steps can severely influence later results.

The actual body of the project consisted mostly of building a lot of linear models and its accompanying diagnostic plots. This is an intensive process but step by step improves your best model and at the same time reveals a lot of information in a data set. A simple example of this is the need of the logarithmic transformation which drastically improves the variability coverage of your model but at the same time decreases the influence and interaction of some explanatory variables, thus yielding information about the actual influence of attributes in a data set. Furthermore, we believe that balancing between improving the response variables' variabilty coverage and not overfitting or adding to much complexity and/or degrees of freedom is a ever recurring and crucial reality in most data science projects.

To sum up, working with this data set is an important reference of how to deal with typical difficulties as well as an overview of what to expect in future data science projects.