USED CAR PRICES CASE STUDY

Deliverable 3: General and Binary/Logistic Regression Models

Miquel Parra i Xavier Alaman

May 29, 2022

Contents

1 R libraries imports, useful functions and data loading

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

1.1 Load Required Packages

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

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

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

1.2 Sample load

This sample has a new variable called engineSize_num that has three categories: small_engine (engineSize<=2), medium_engine (2<engineSize<=3) and large_engine (engineSize>3).

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

# Clean workspace
rm(list=ls())

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

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

1.3 Preparing the data

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

```
names(df)
```

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

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

```
## integer(0)
```

```
df$years_sell[ll]<-0.5

ll<-which(df$tax==0);ll
```

```
## integer(0)
```

```
df$tax[ll]<-0.5

ll<-which(df$mileage==0);ll
```

```
## integer(0)
```

```
df$mileage[ll]<-0.5

ll<-which(df$mpg==0);ll
```

```
## integer(0)
```

```
df$mpg[ll]<-0.5
```

2 Linear Models

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

2.1 Only Numeric variables

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

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

```
summary(m1)
```

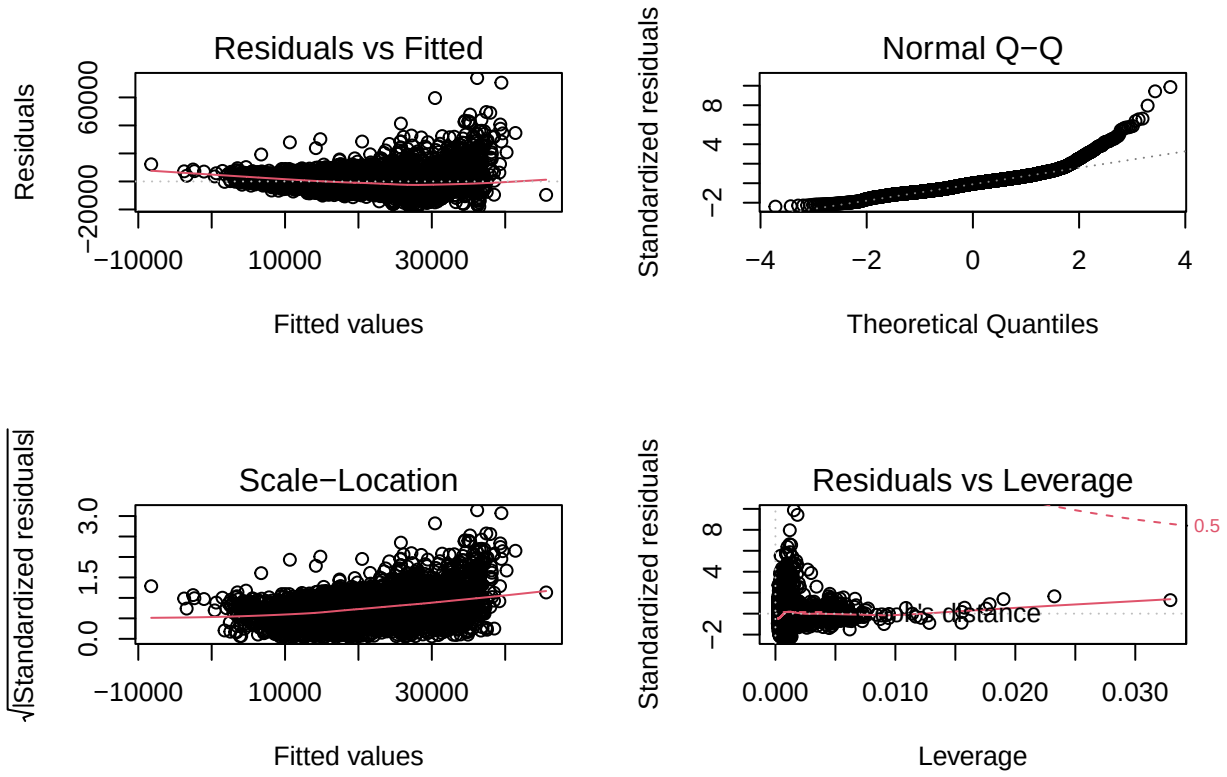
```
##
## Call:
## lm(formula = price ~ mileage + tax + mpg + years_sell, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -17976  -5039   -312   3478  73806
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.831e+04  1.634e+03  35.691  < 2e-16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

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

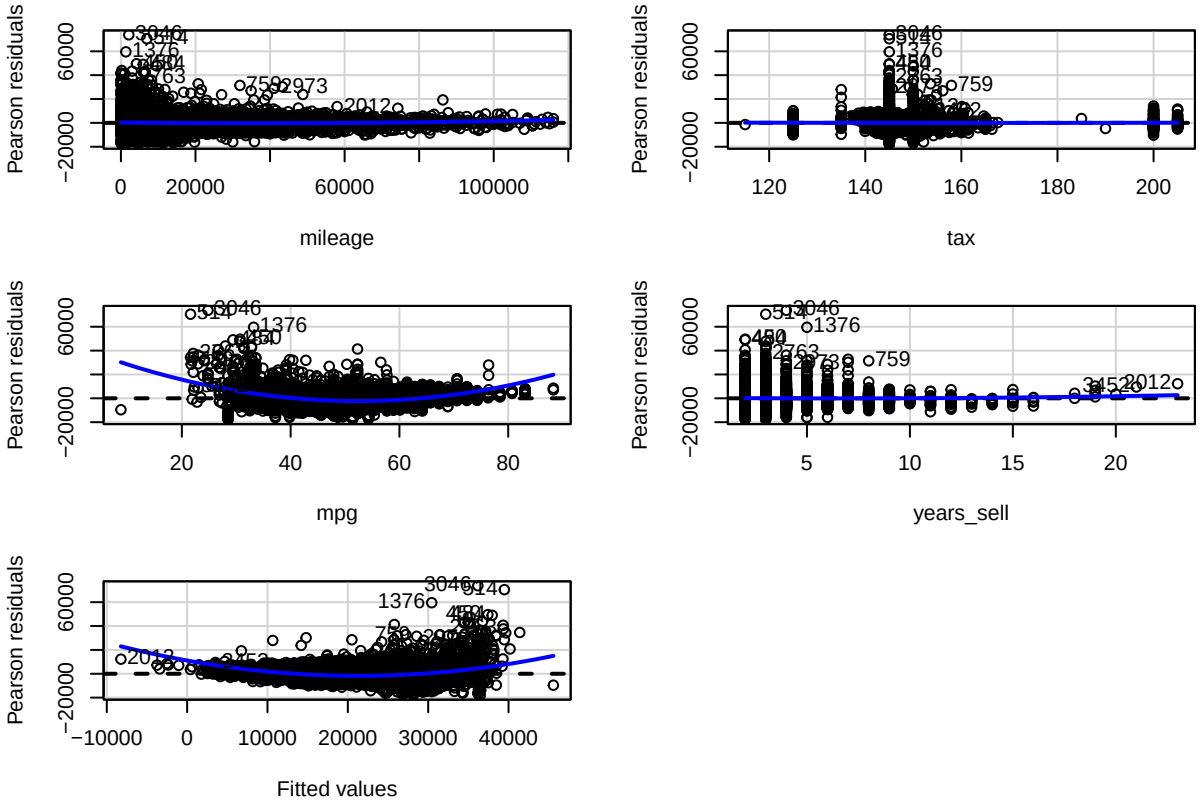
```
vif(m1)
```

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

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




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




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

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

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

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

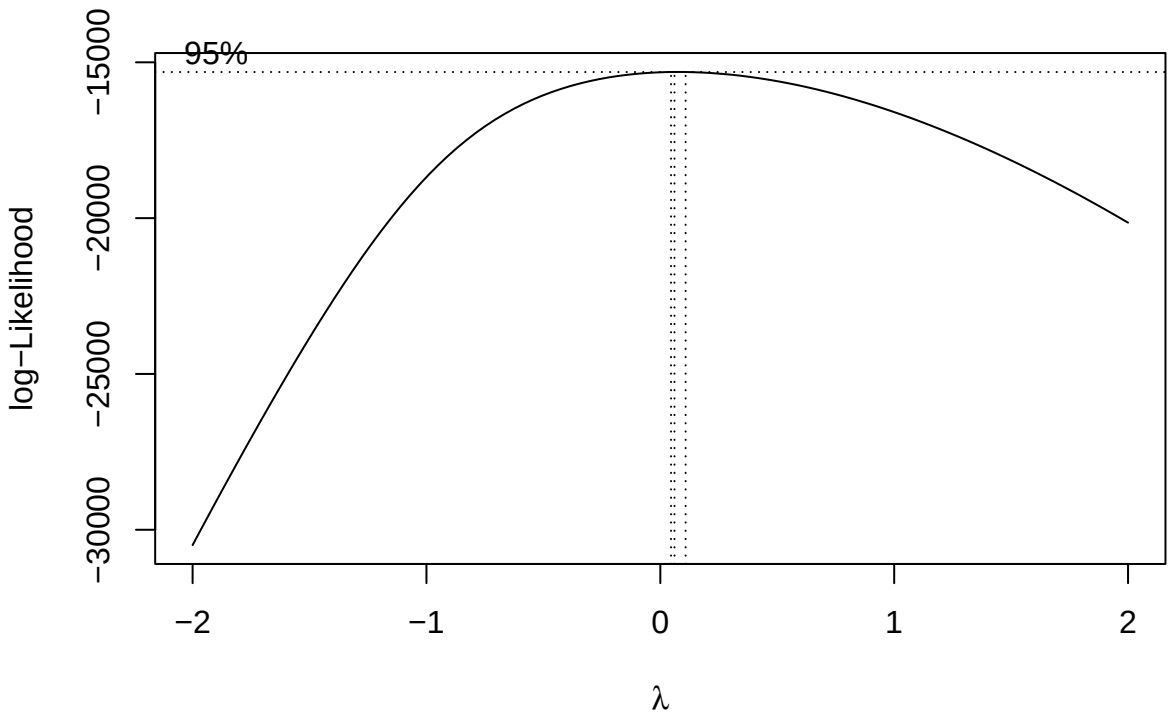
2.1.1 Target transformation

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

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



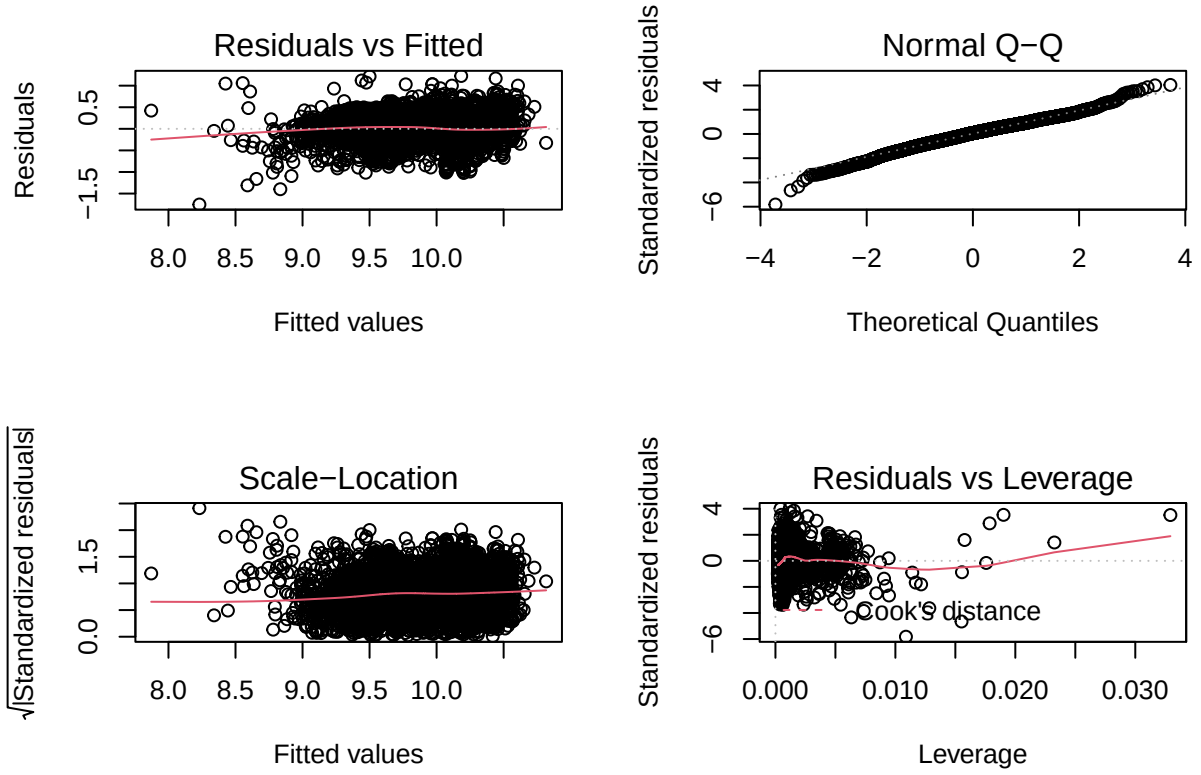

```
# Lambda=0 - log transformation is needed

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

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

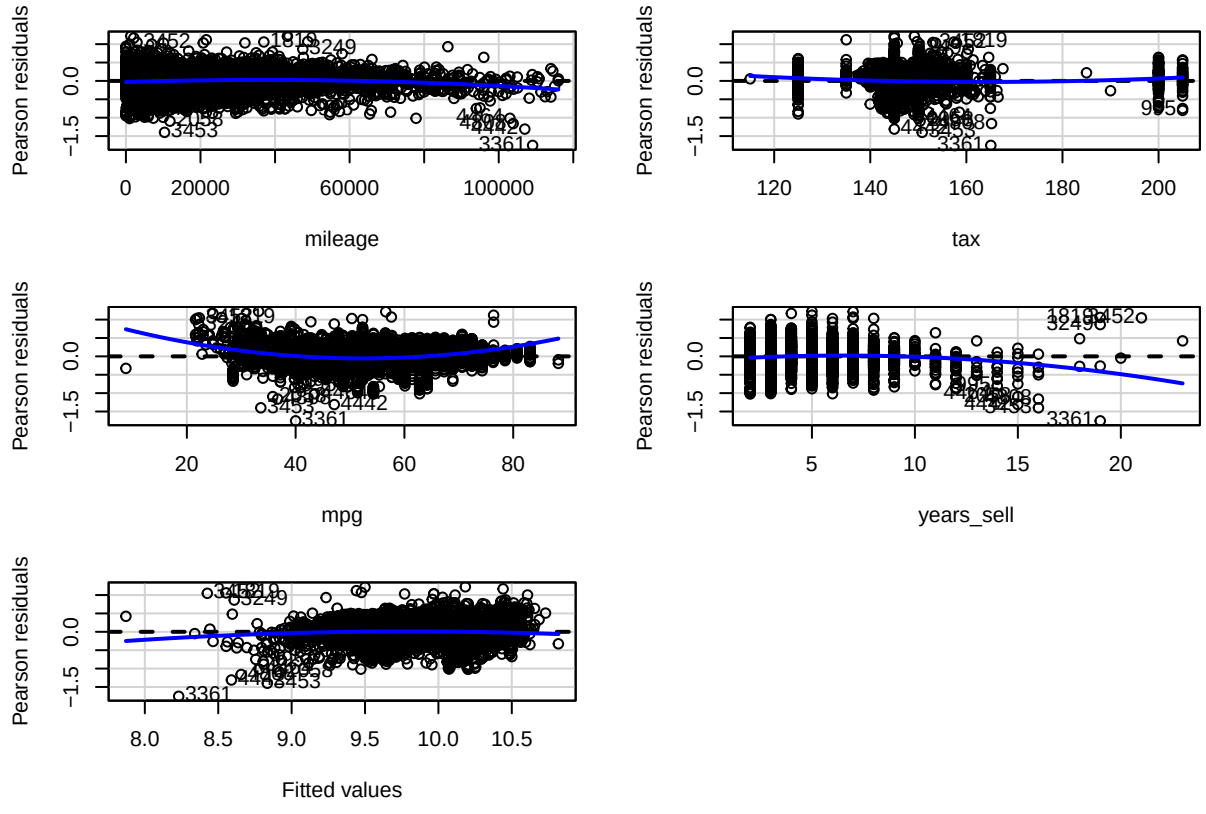
```
vif(m2)
```

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

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




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




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

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

This model is better than the previous one.

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

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

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

2.1.2 Explanatory variables transformation

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

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

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

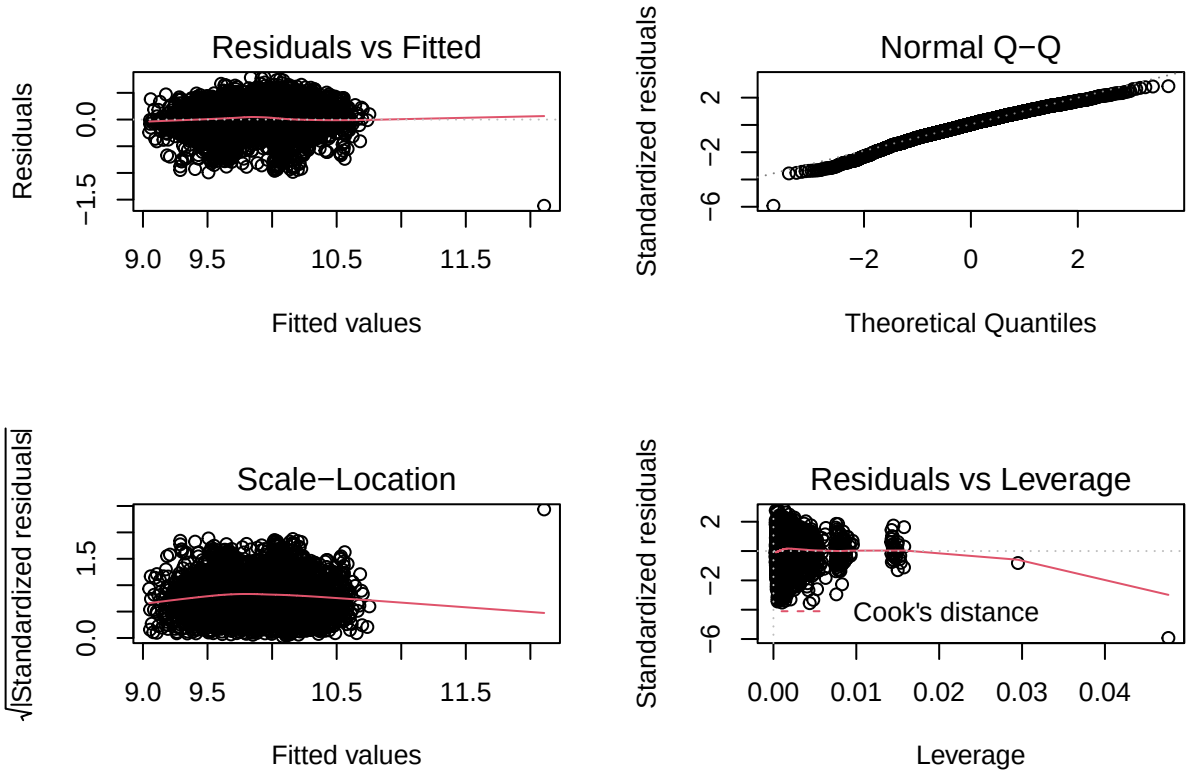
```
m3<-lm(log(price)~sqrt(mileage)+poly(tax,3)+ I(mpg^(-1/2))+years_sell,data=df[!df$mout=="YesMOut",])

summary(m3)
```

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

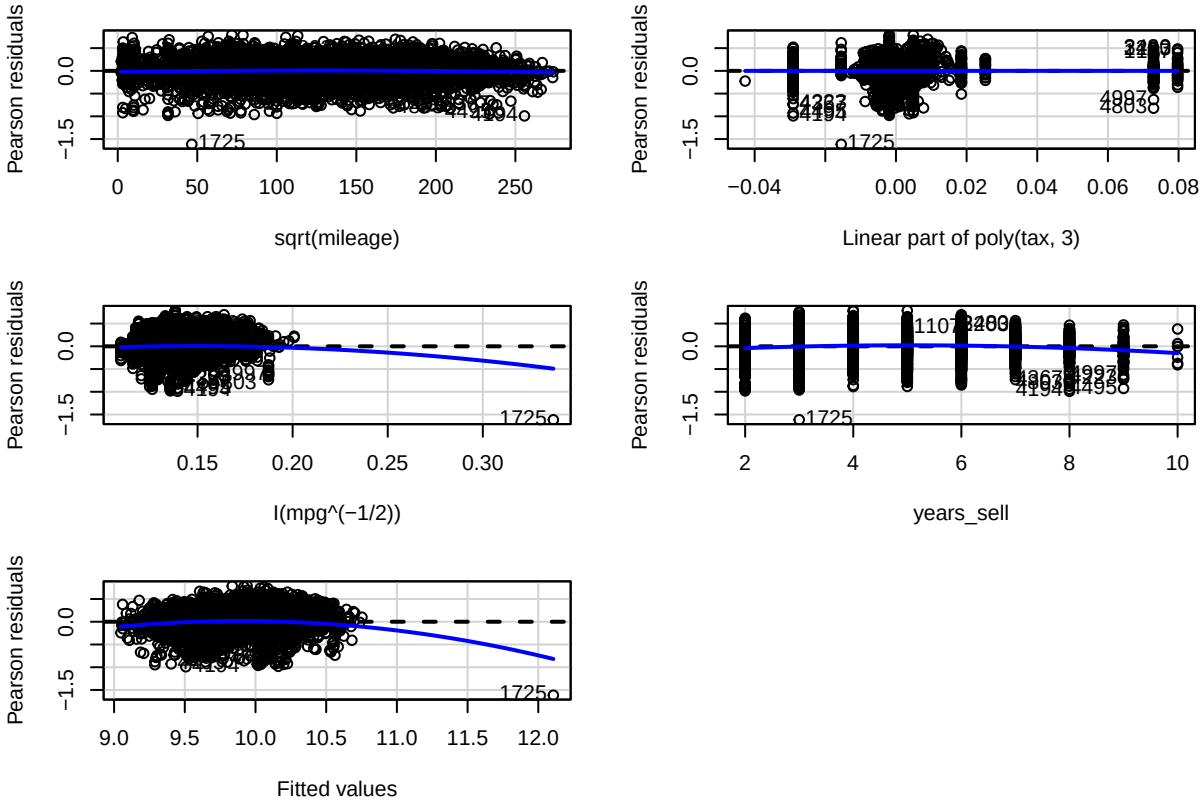
```
vif(m3)
```

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

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




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




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

This model is better than the previous one.

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

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

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

2.2 Including factors

2.2.1 Significant factors

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
summary(m4)
```

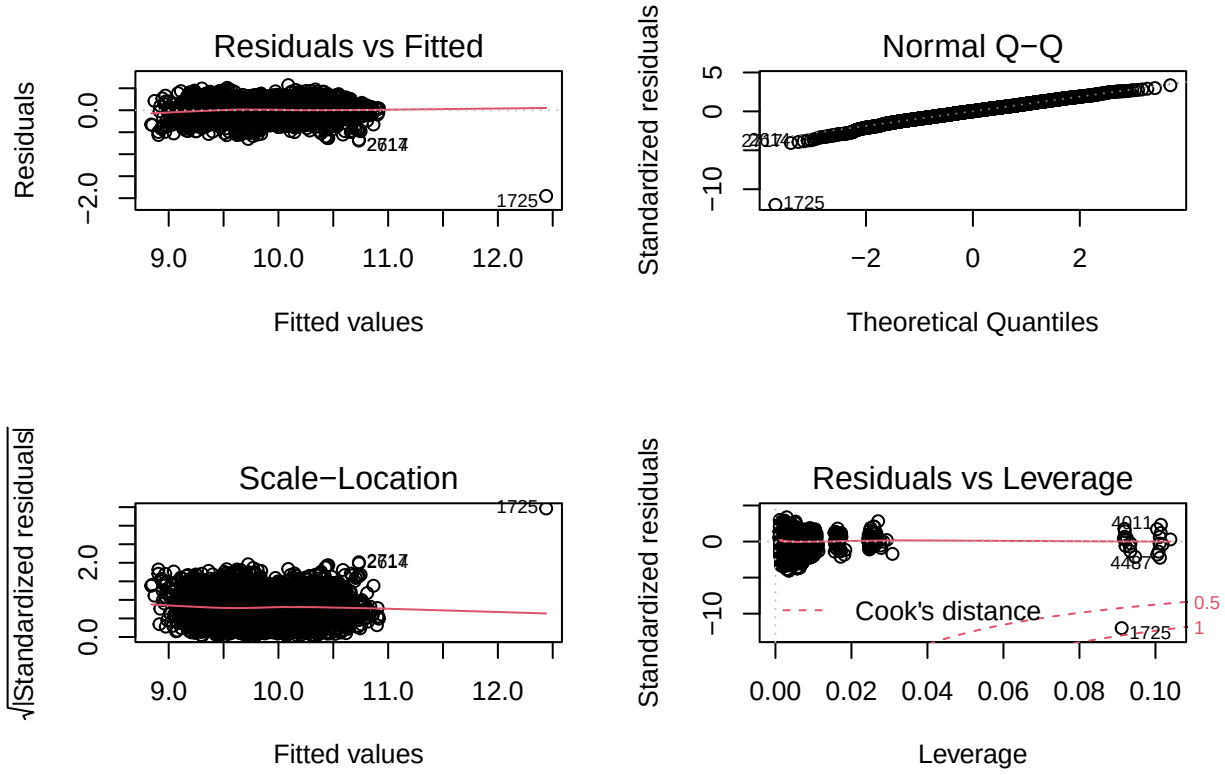
```
##
## Call:
## lm(formula = log(price) ~ sqrt(mileage) + poly(tax, 3) + I(mpg^(-1/2)) +
##     years_sell + transmission + fuelType + manufacturer + engineSize_num,
##     data = df[!df$mout == "YesMOut", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94903 -0.10563  0.00462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

```
vif(m4)
```

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

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

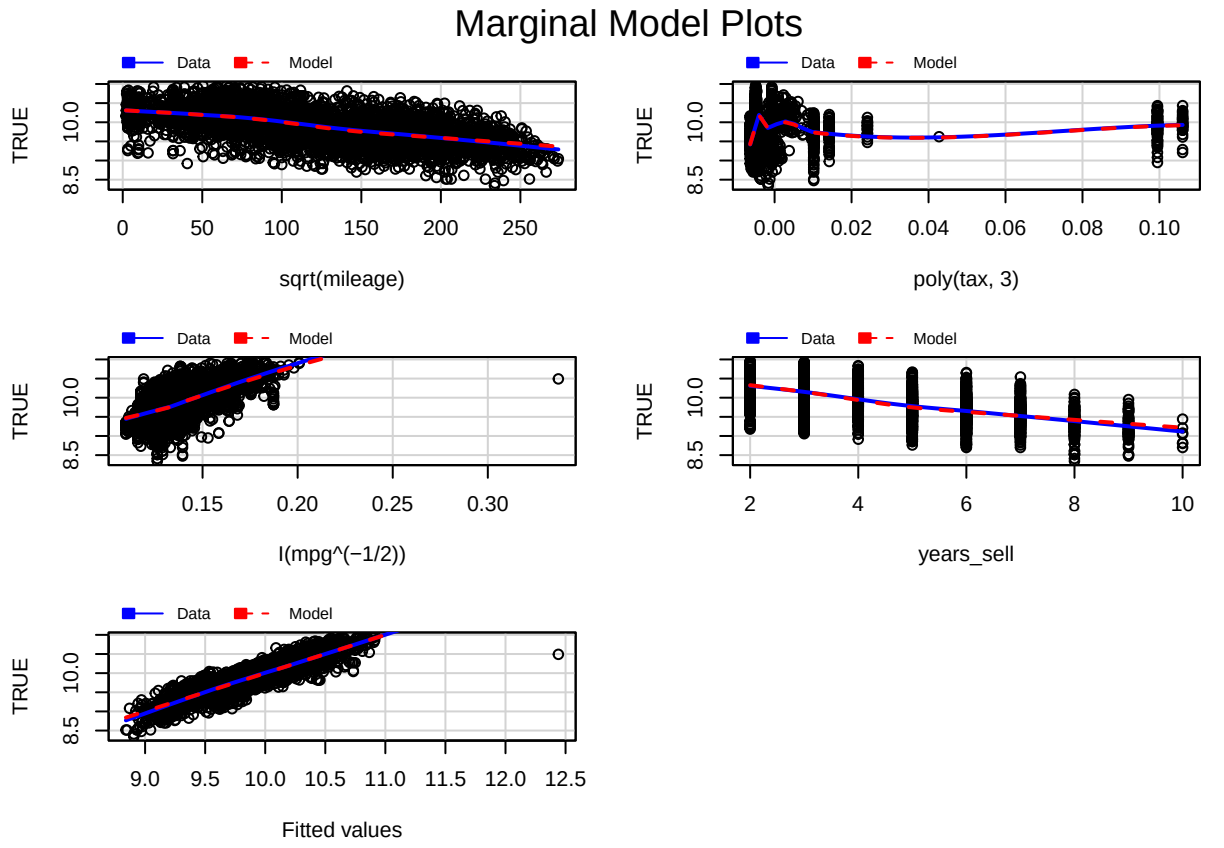
```
par(mfrow=c(2,2))
plot(m4)
```



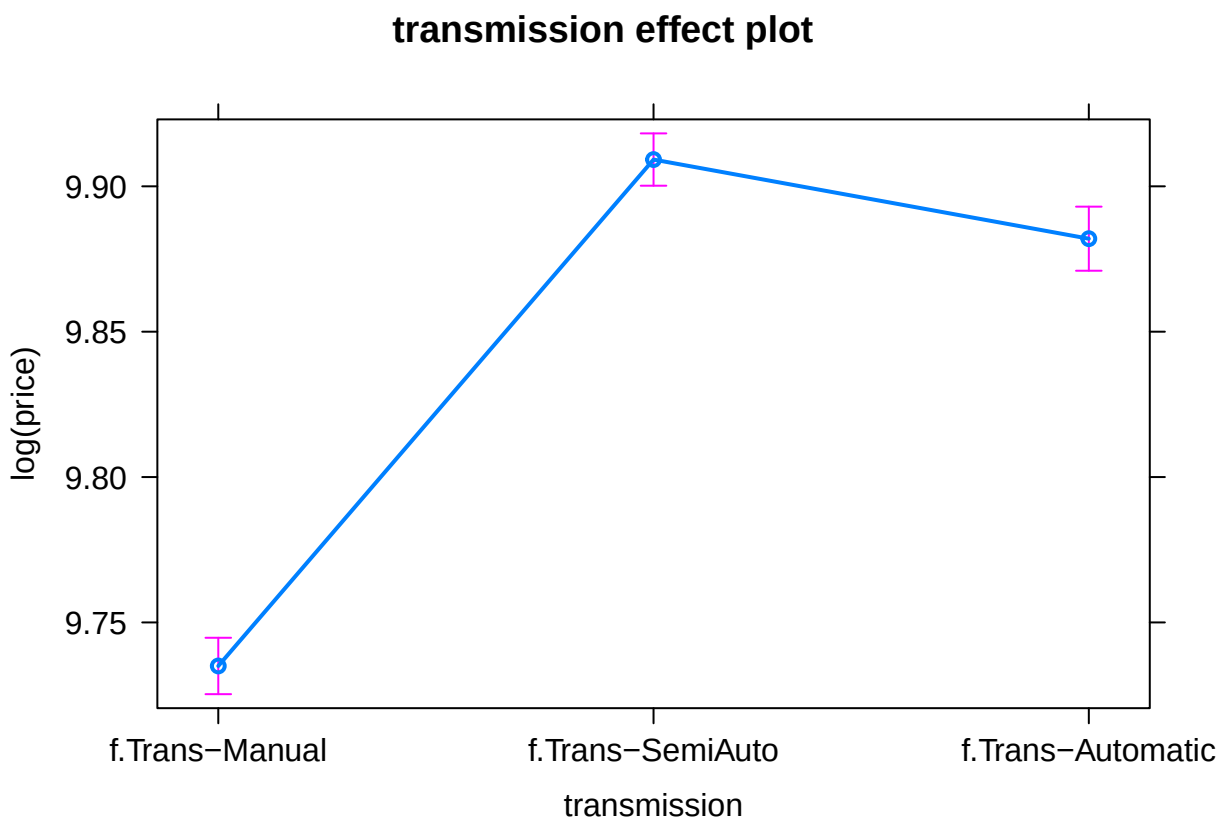

```
marginalModelPlots(m4)
```

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

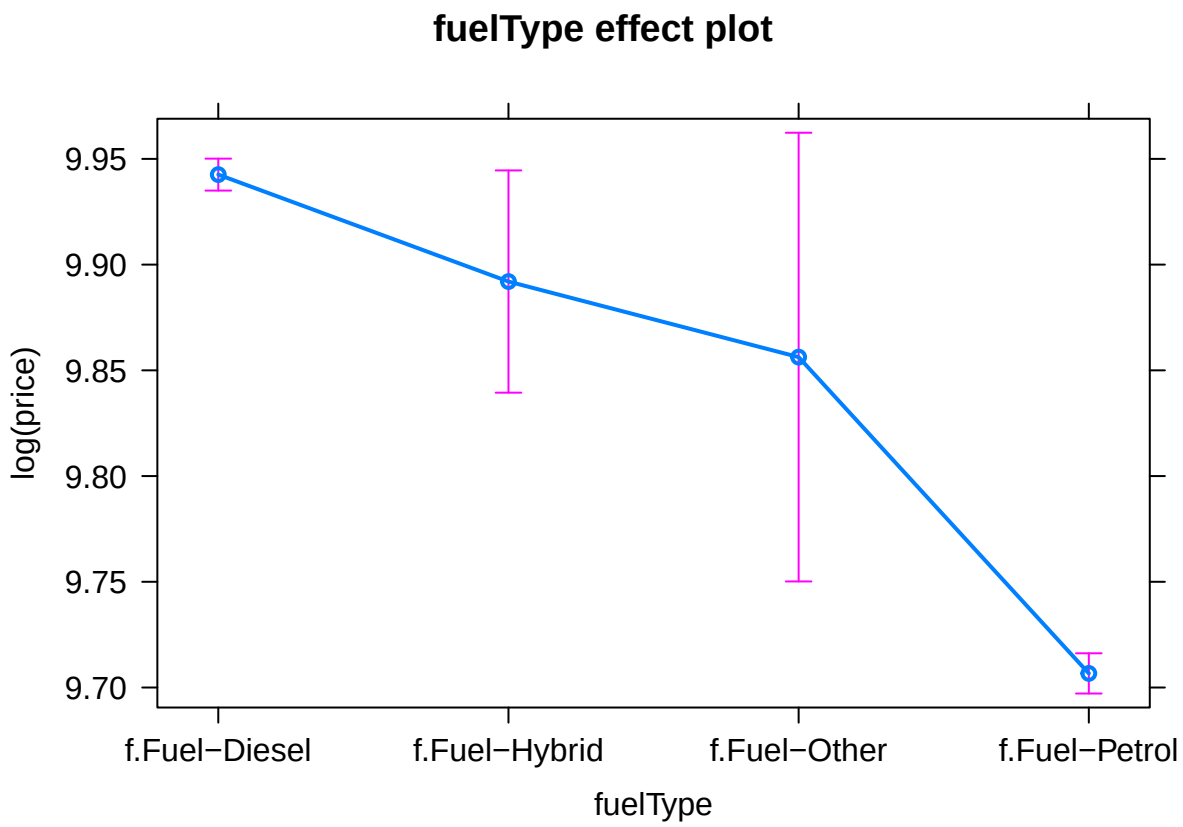
```
## Warning in mmps(...): Interactions and/or factors skipped
```



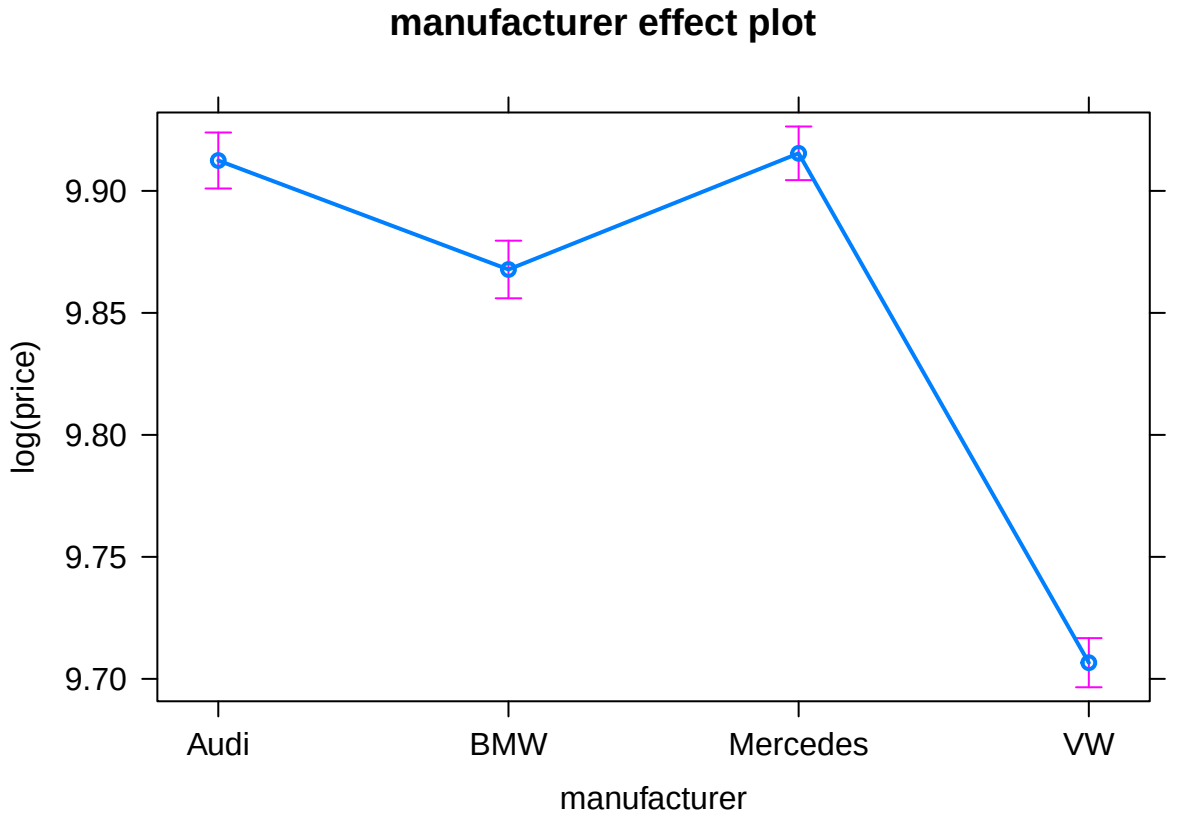

```
#residualPlots(m4,id=list(method=cooks.distance(m4),n=10))
plot(allEffects(m4), selection = 5)
```



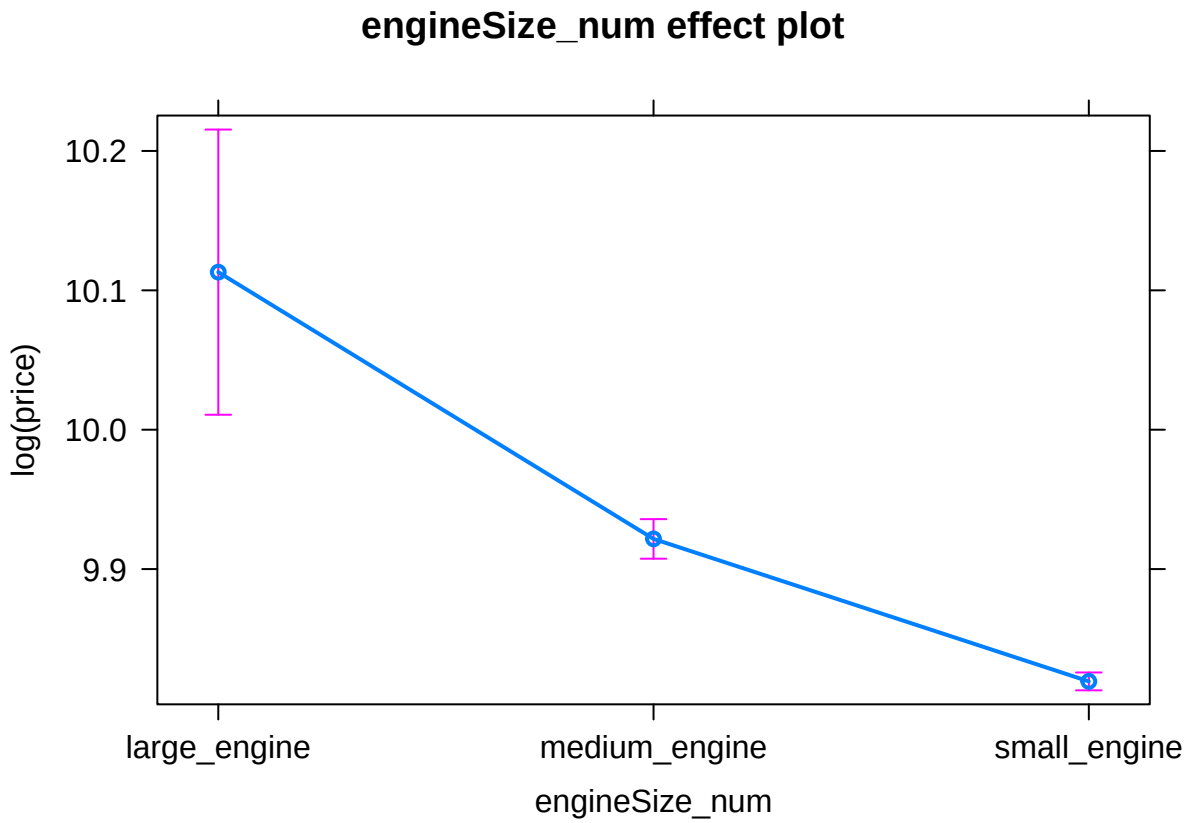

```
plot(allEffects(m4), selection = 6)
```




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



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



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

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

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

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

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

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

2.3 Interactions

2.3.1 Factors interaction

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

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

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

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

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

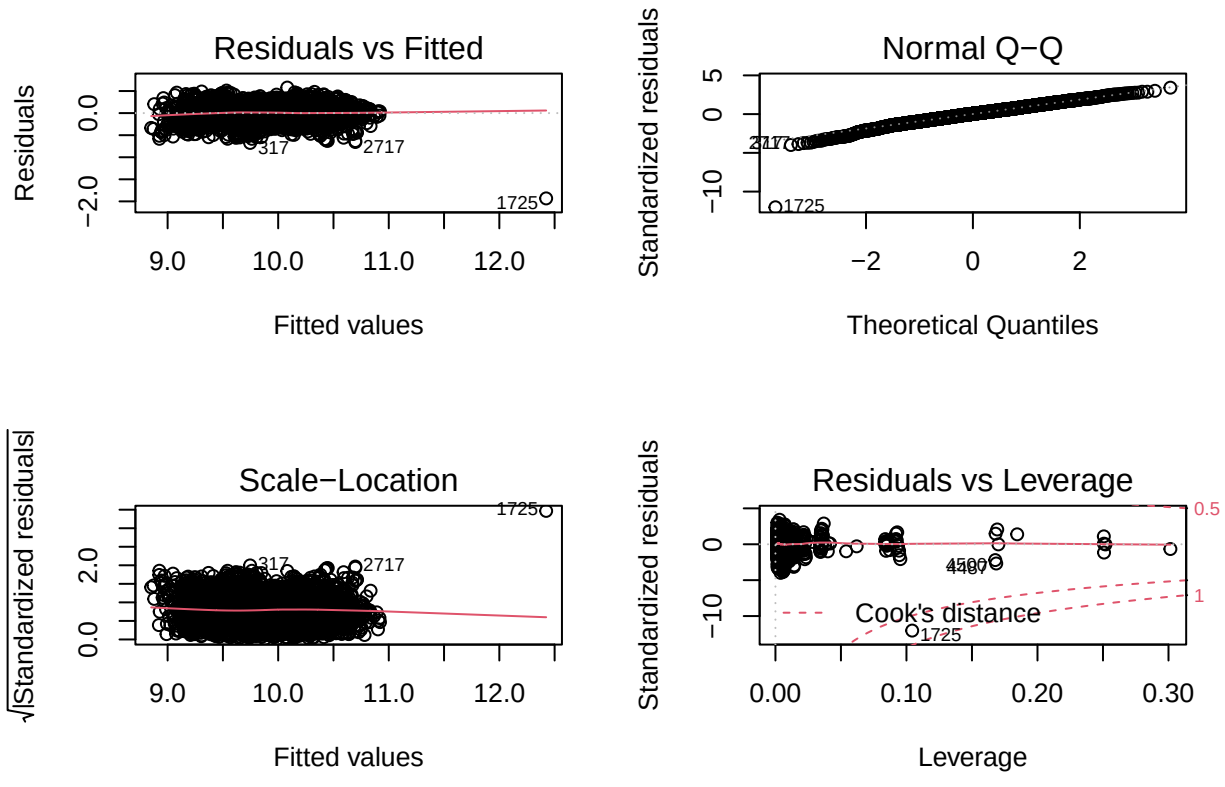
```
##    df       AIC
## m4 18 -3168.681
## m5 24 -3186.836
```

```
summary(m5)
```

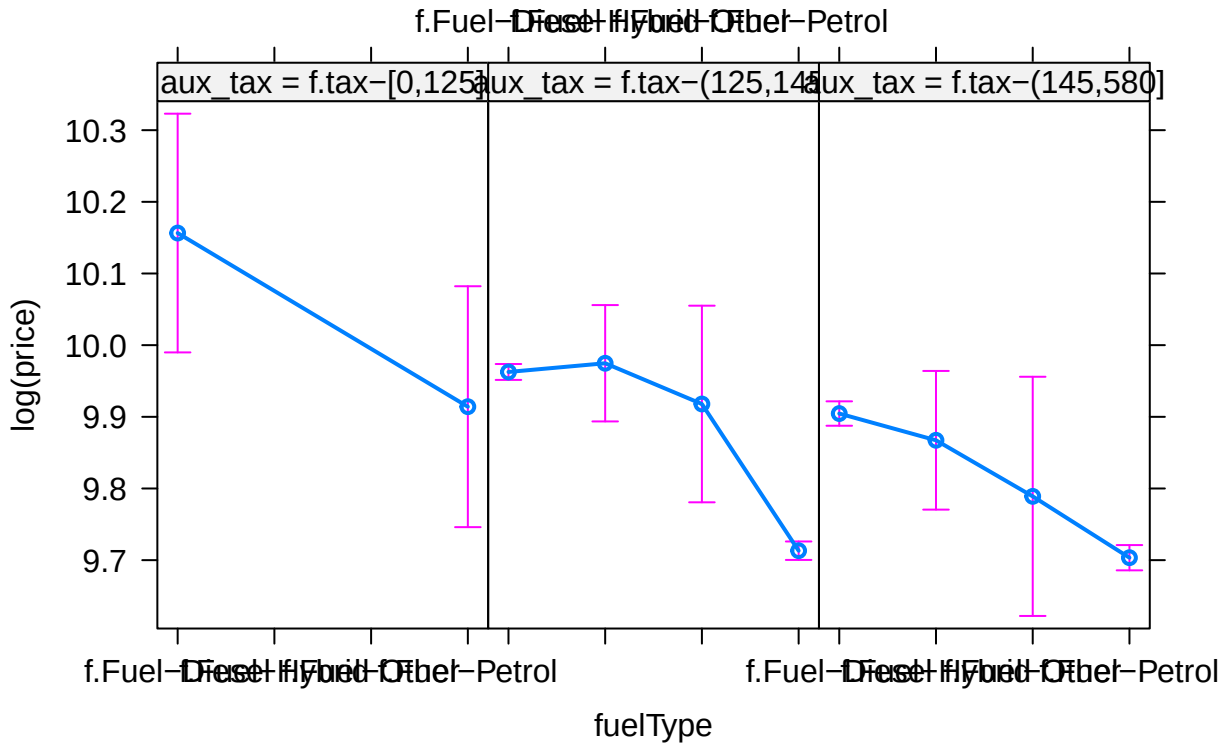
```
##
## Call:
## lm(formula = log(price) ~ sqrt(mileage) + fuelType * aux_tax +
##     poly(tax, 3) + I(mpg^(-1/2)) + years_sell + transmission +
##     manufacturer + engineSize_num, data = df[!df$mout == "YesMOut",
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.93233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

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

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




```
plot(allEffects(m5), selection = 8)
```



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

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

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

2.3.2 Factor and covariate interaction

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

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

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

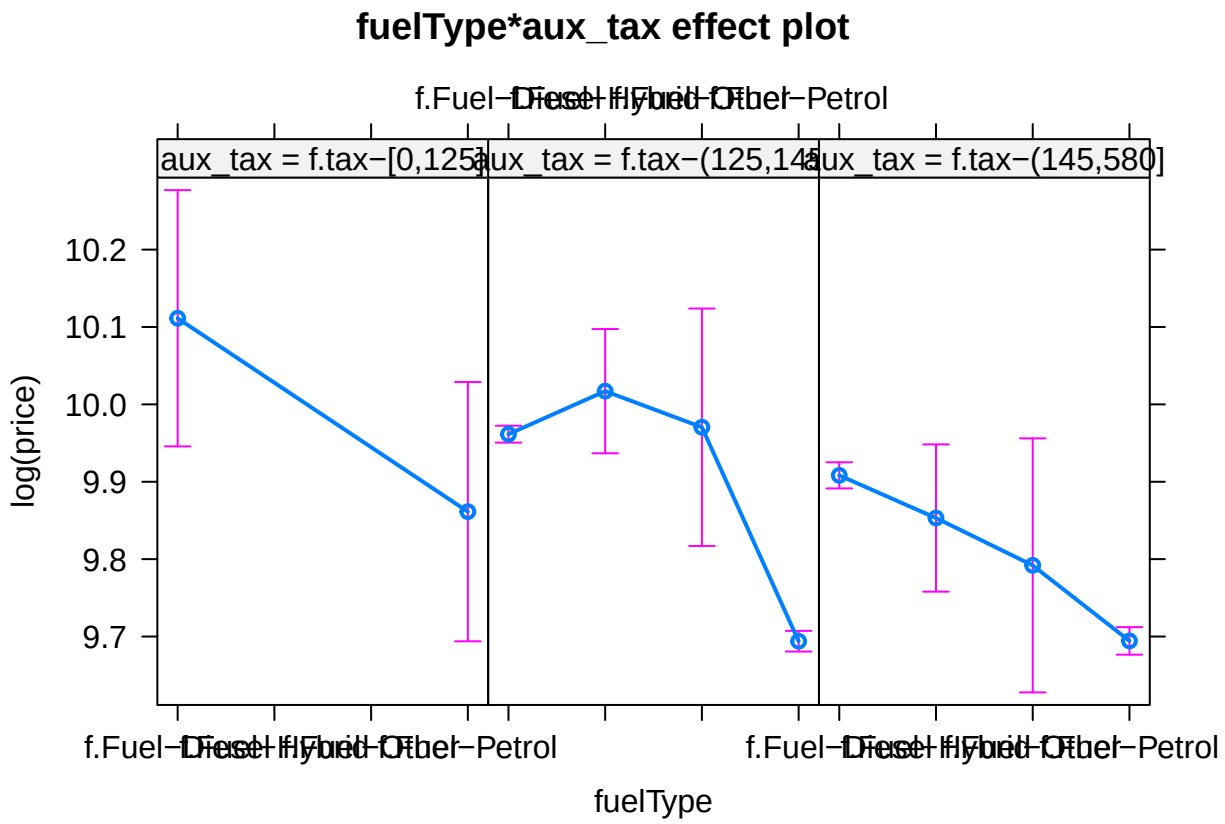
```
summary(m6)
```

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

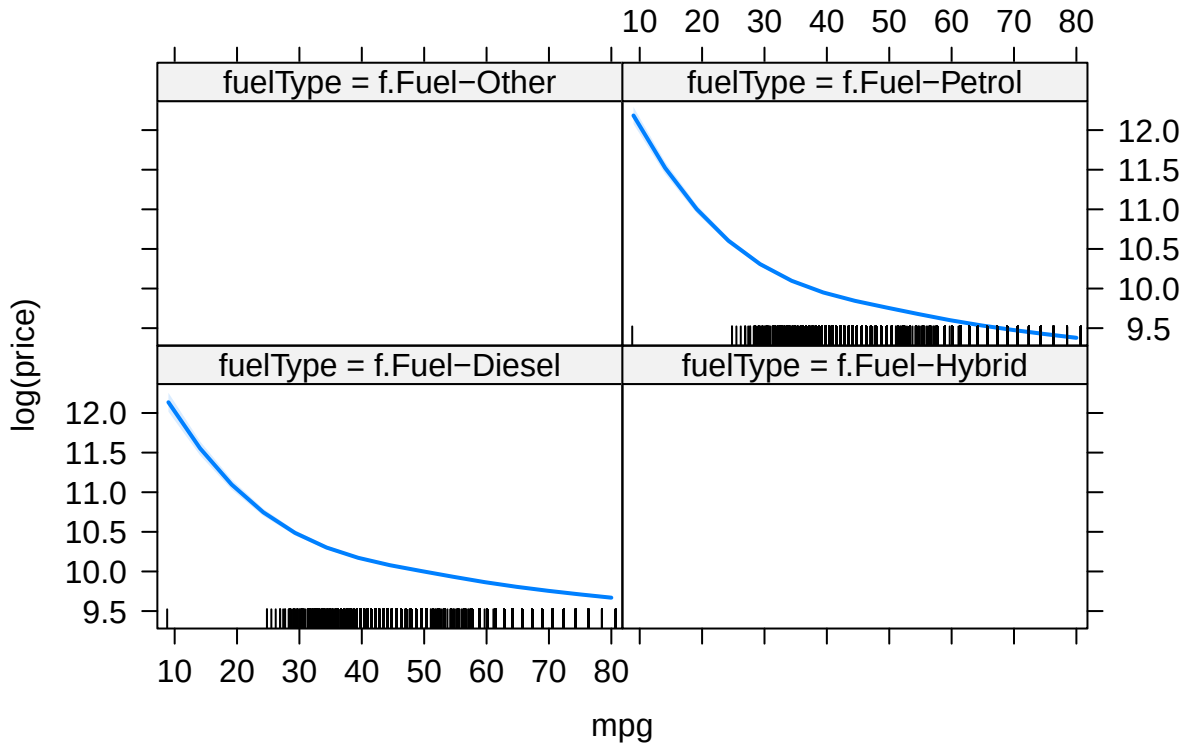
```
##     fuelType * I(mpg^(-1/2)) + poly(tax, 3) + years_sell + transmission +
##     manufacturer + engineSize_num, data = df[!df$mout == "YesMOut",
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69746 -0.10487  0.00618  0.10768  0.59387
##
## Coefficients: (2 not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

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

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




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



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

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

We can see that the only significant interactions are:

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

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

2.4 Best model selection

```
m7 <- step( m6, k=log(nrow(df[!df$mout=="YesMOut",])))
```

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

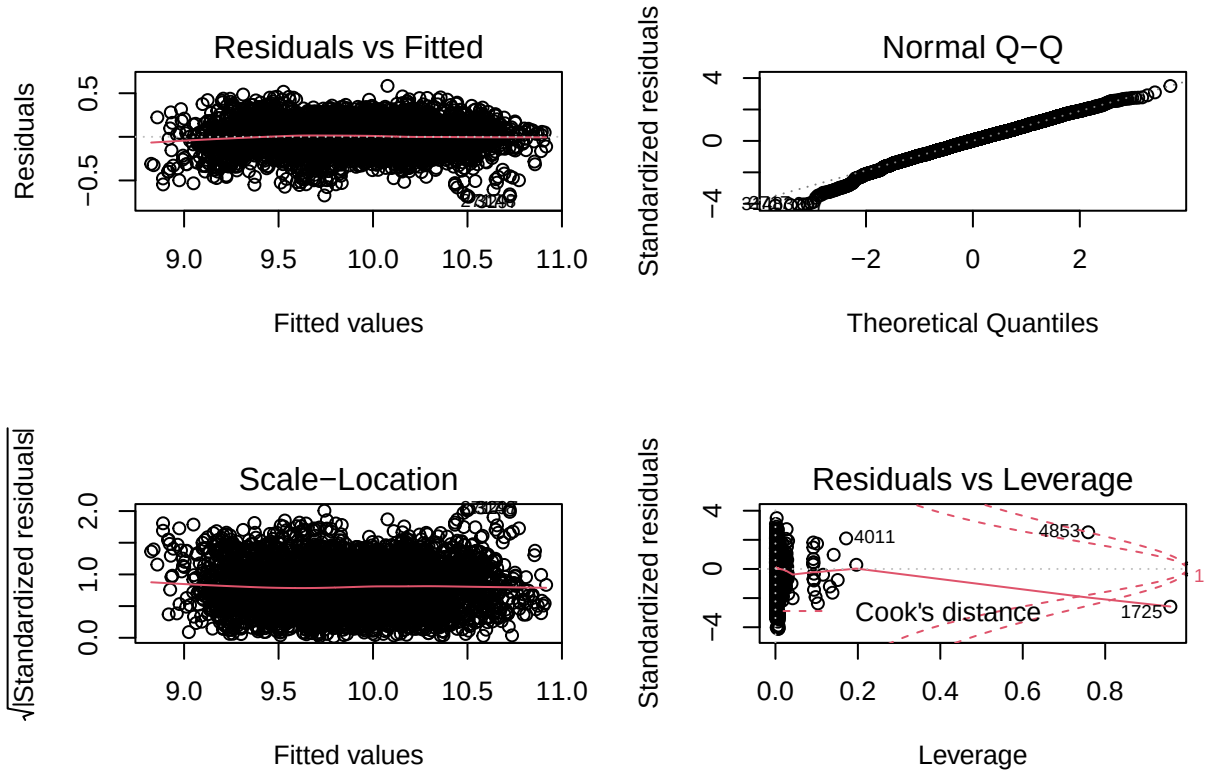
```
##     engineSize_num + fuelType:I(mpg^(-1/2))
##
##                           Df Sum of Sq    RSS    AIC
## - aux_tax                  2    0.1544 126.61 -16064
## <none>                                  126.46 -16053
## - poly(tax, 3)             3    1.1615 127.62 -16037
## - fuelType:I(mpg^(-1/2))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

```
summary(m7)
```

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

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

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



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

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

2.5 Diagnostics

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

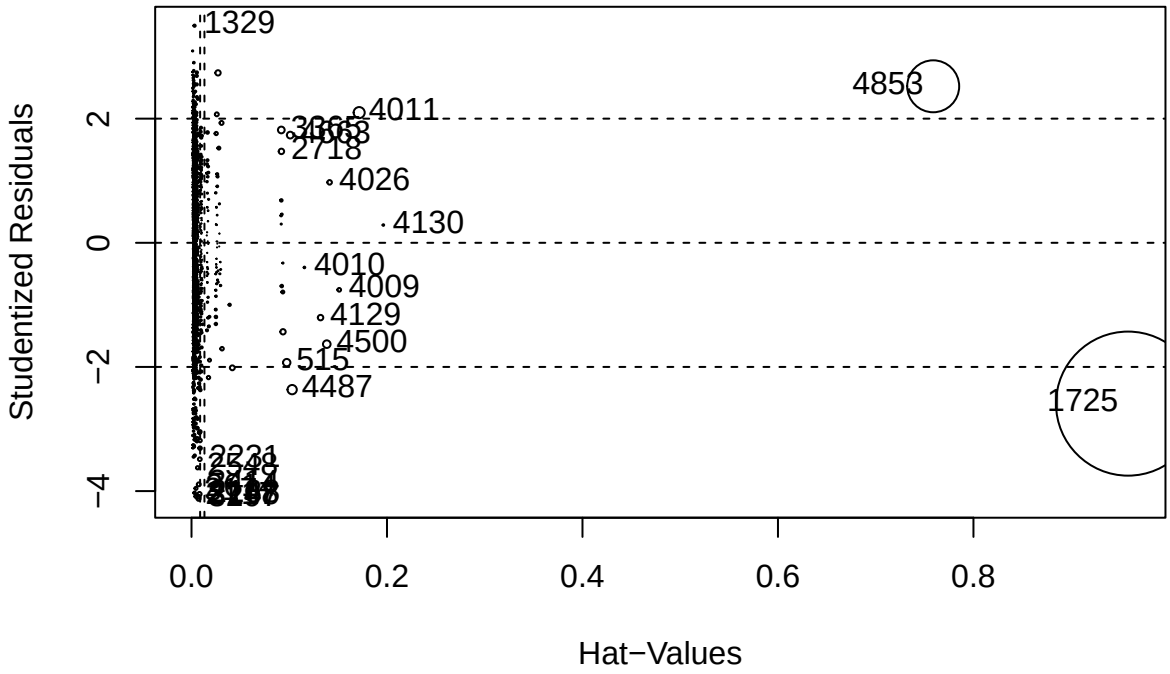
```
dfwork <- df[!df$mout=="YesMOut",]
# Define initial parameters:
p <-length(m7$coefficients)
n <- length(m7$fitted.values)
h_param <- 3

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

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

```
length(llres)
```

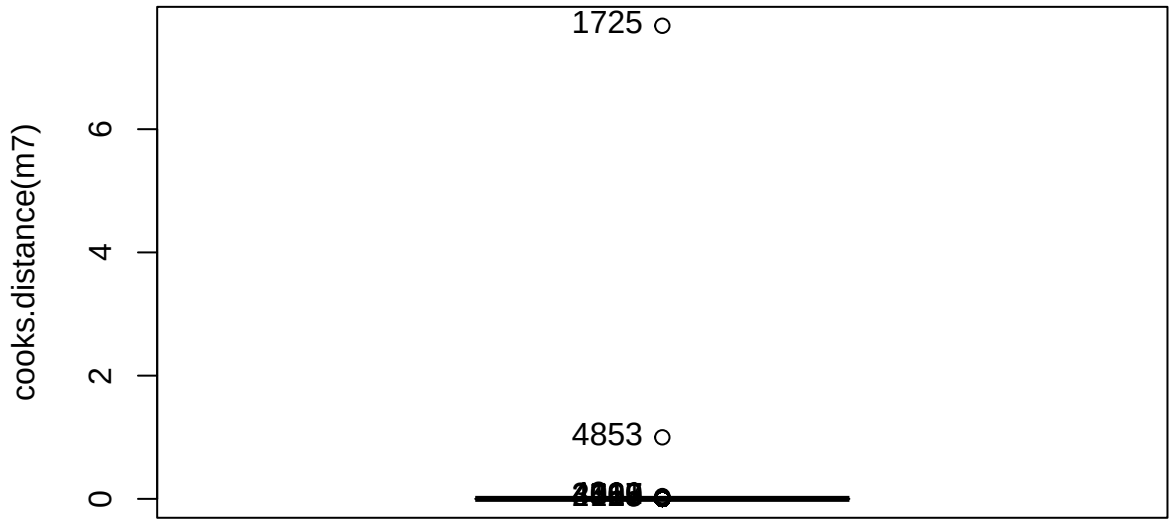
```
## [1] 30
```

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




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

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




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

```
# A priori influential observation
ll_priori_influential <- which(abs(hatvalues(m7))>h_param*(p/n))
length(ll_priori_influential)
```

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

```
# A posteriori influential observation:
ll_posteriori_influential<-which(abs(cooks.distance(m7))>(4/(n-p)));
length(ll_posteriori_influential)
```

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

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

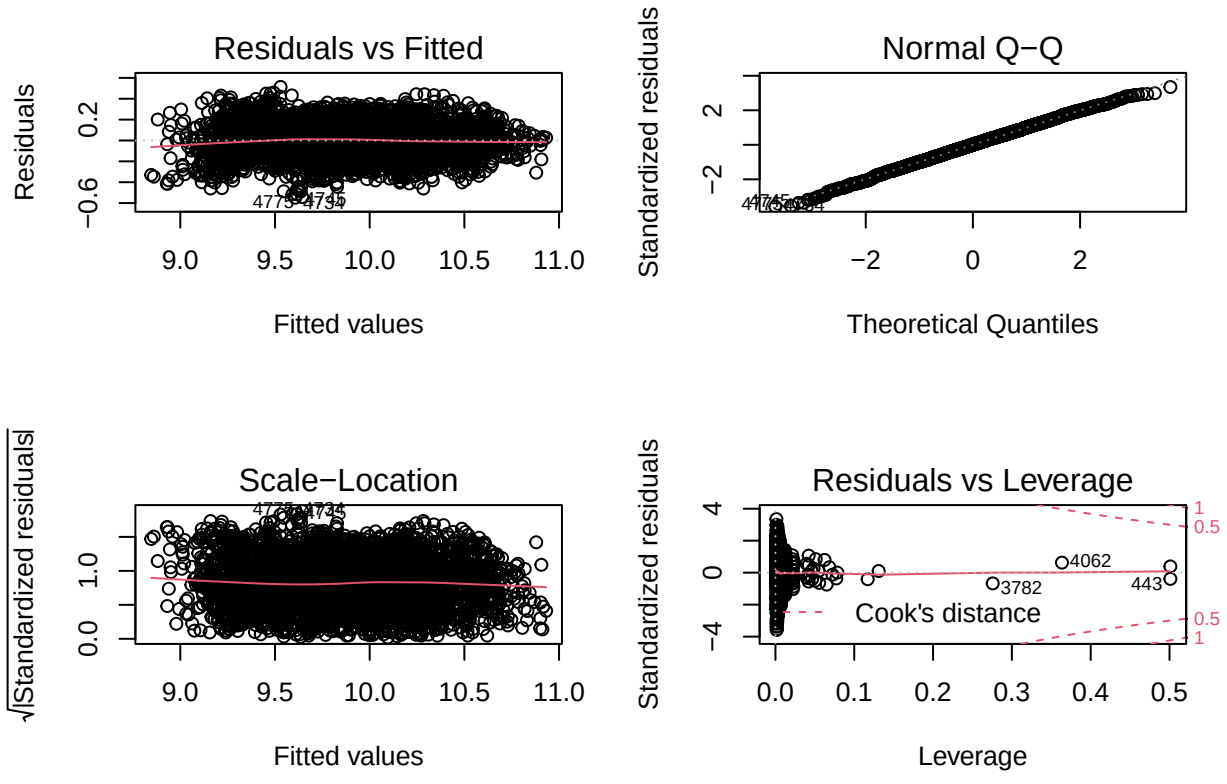
```
## [1] 209
```

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

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

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

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



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

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

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

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

3 Binary/Logistic Regression Models

3.1 Dividing/Splitting the sample

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


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

3.2 Only numeric variables

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

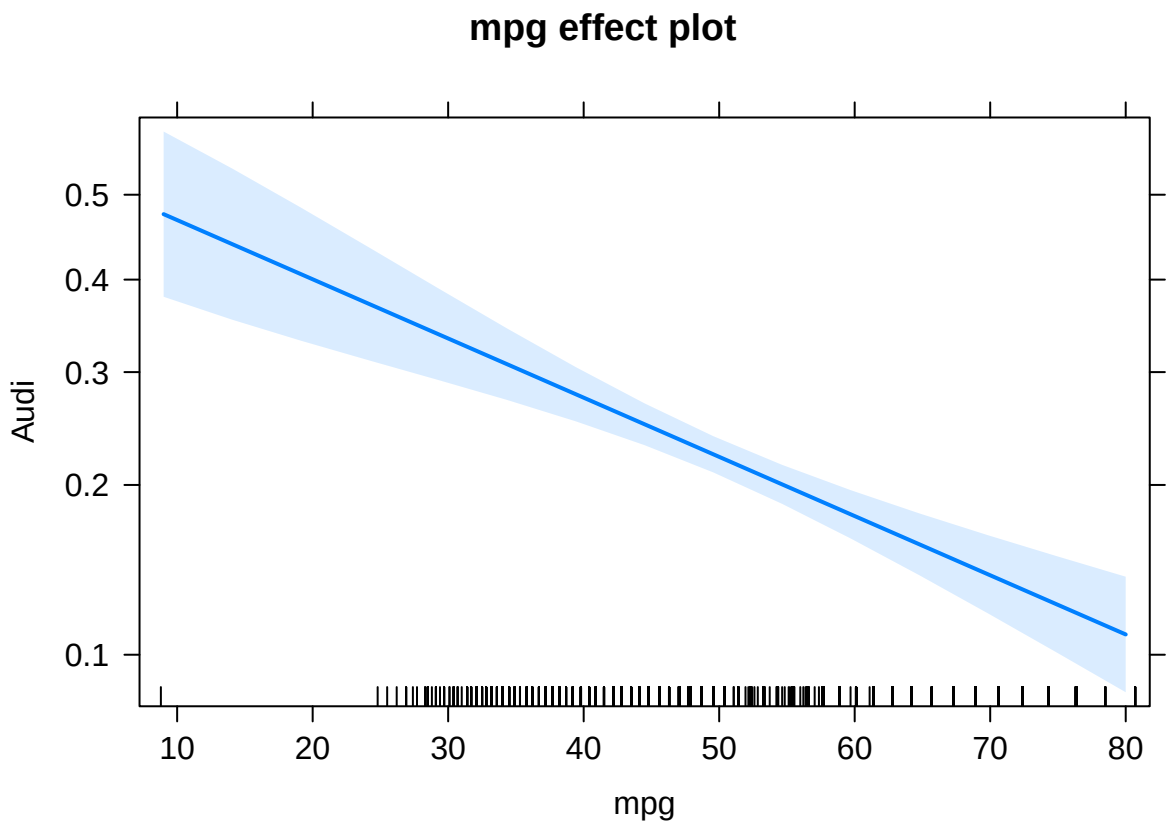
```
## Analysis of Deviance Table
##
## Model 1: Audi ~ 1
## Model 2: Audi ~ mileage + tax + mpg + years_sell
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      3169     3288.2
## 2      3165     3242.1  4   46.112 2.334e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1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

```
summary(m1)
```

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

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

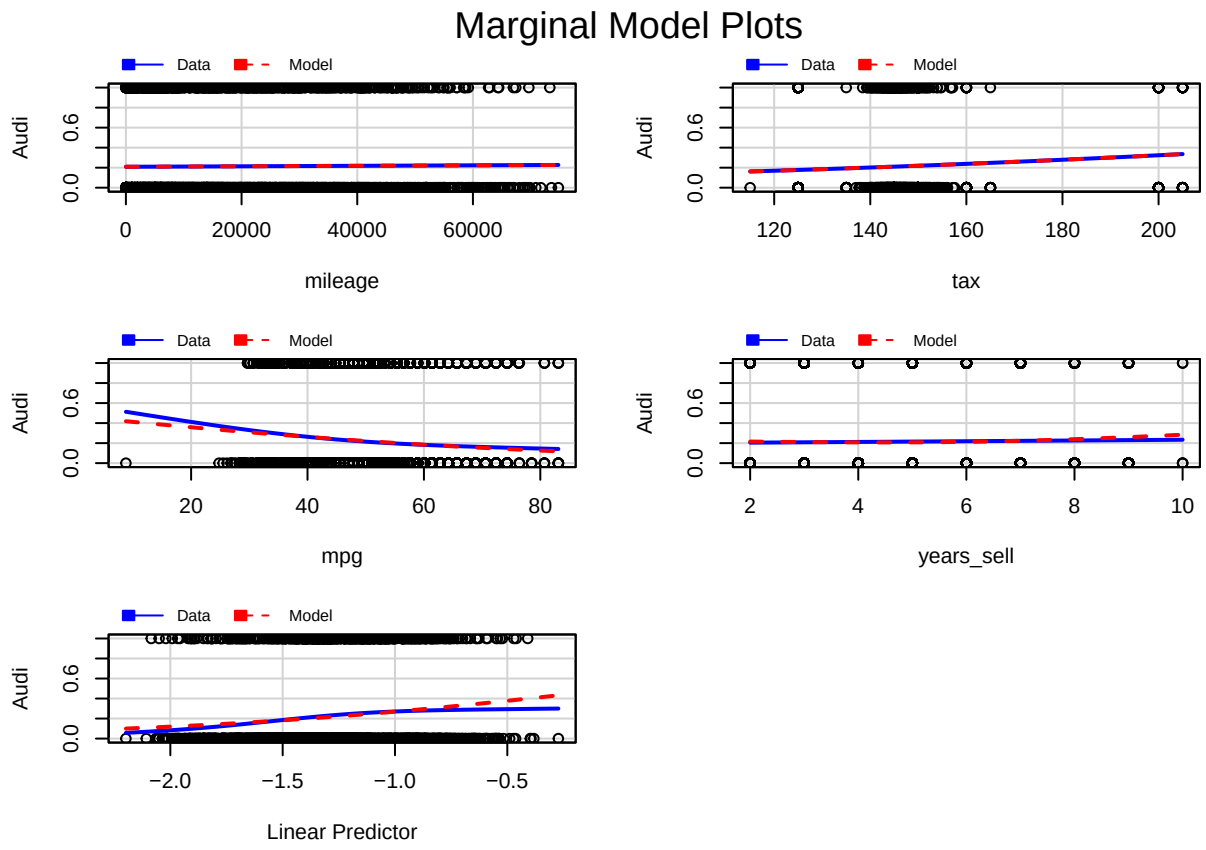
```
plot(allEffects(m1), selection = 3)
```



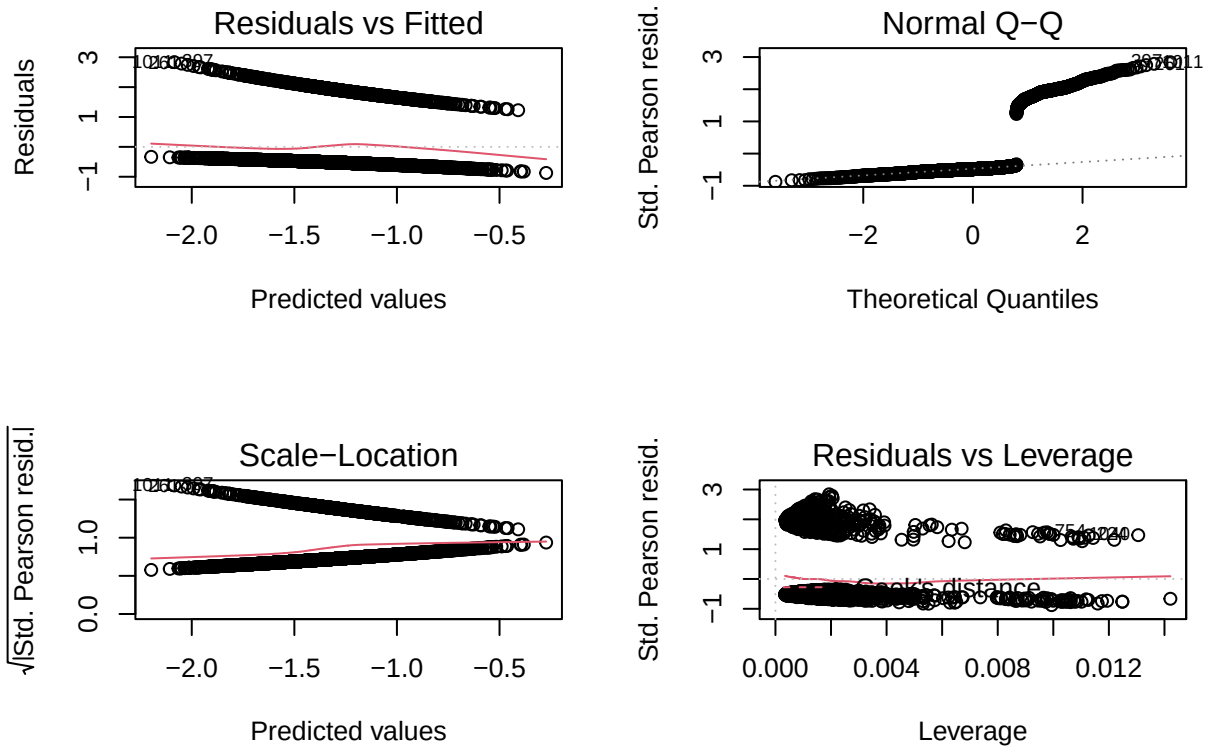

```
vif(m1)
```

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

```
marginalModelPlots(m1)
```




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



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

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

We can say that against more mpg less probability of being Audi. More specifically, increasing by 1 unit mpg then exp(-0.028*1) = 0.97 -> 100*(1-0.97) = 3 %, the probability of being Audi decreases by 3 %.

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

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

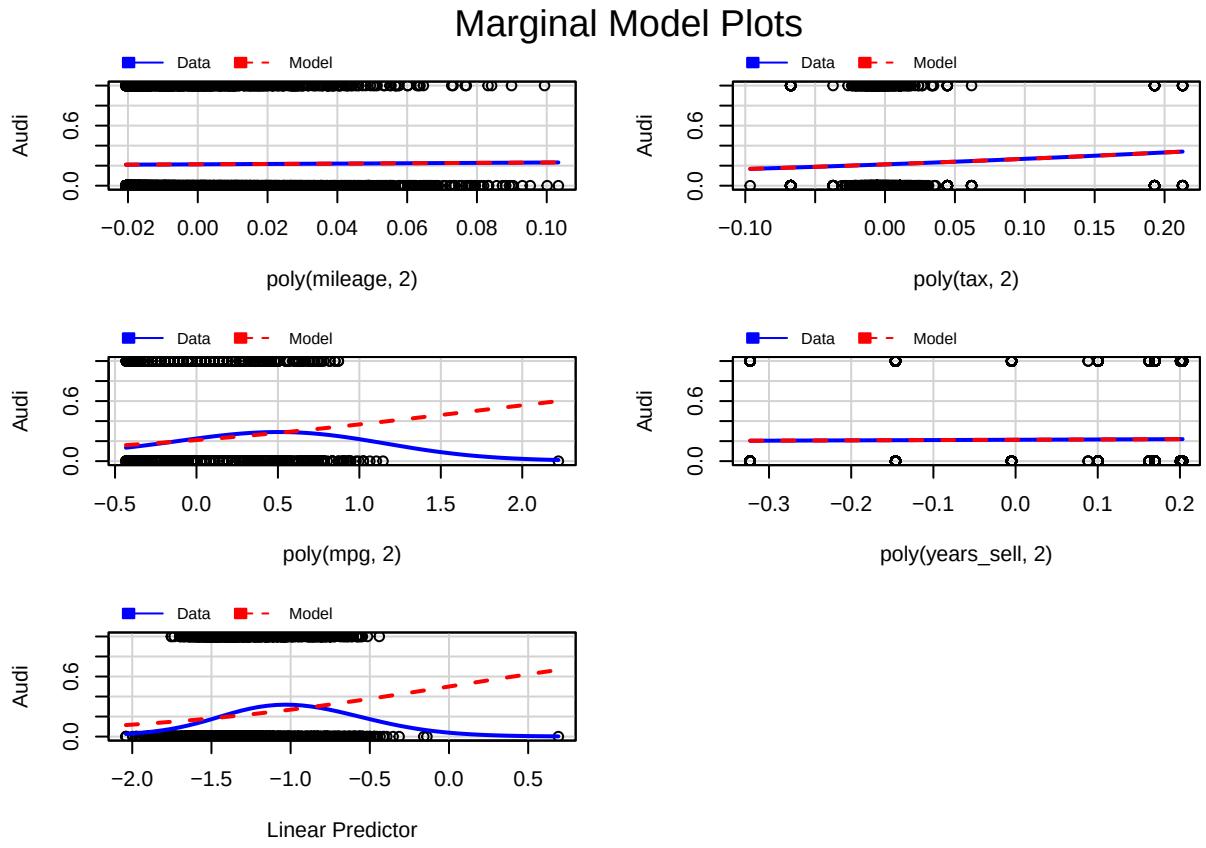
3.2.1 Transformations

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

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

```
marginalModelPlots(m2)
```

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



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

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

3.3 Including factors

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

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

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

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

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

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

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

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

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

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

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

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

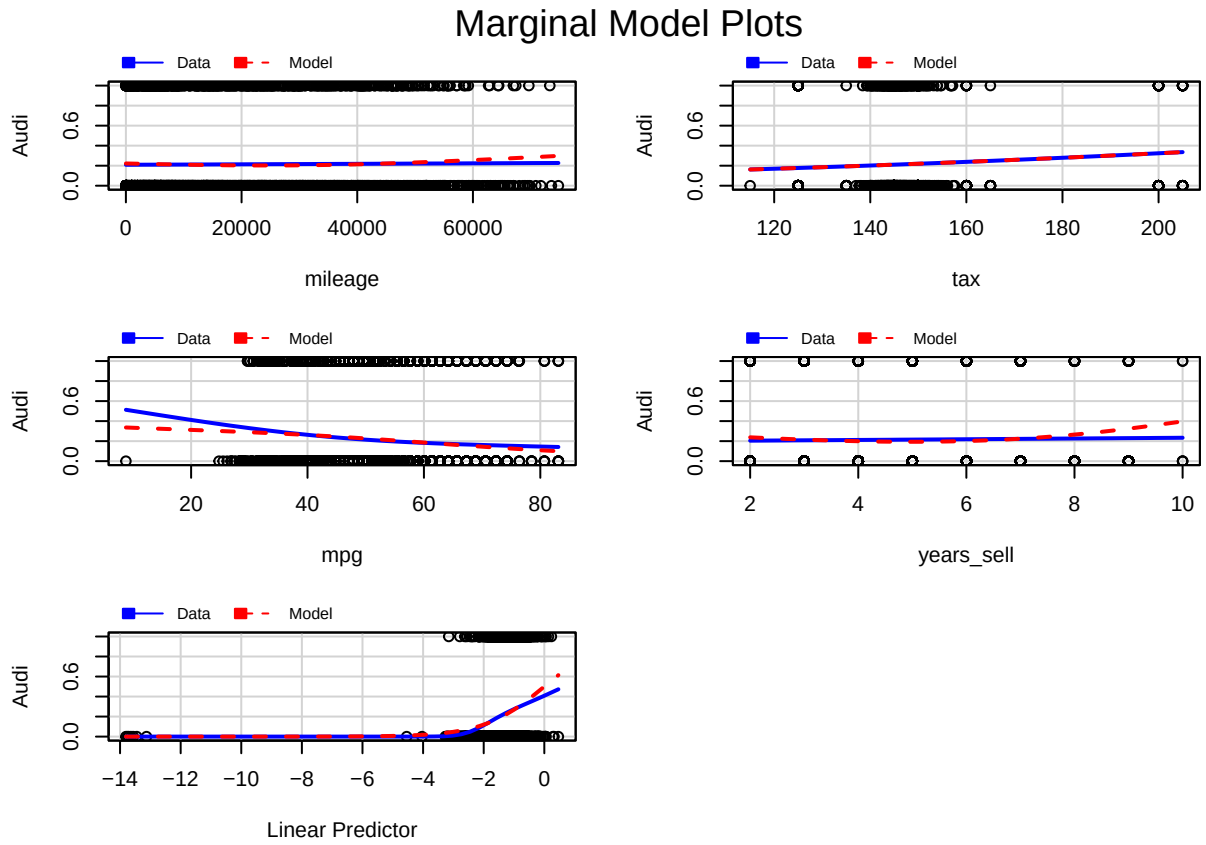
```
summary(m2)
```

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

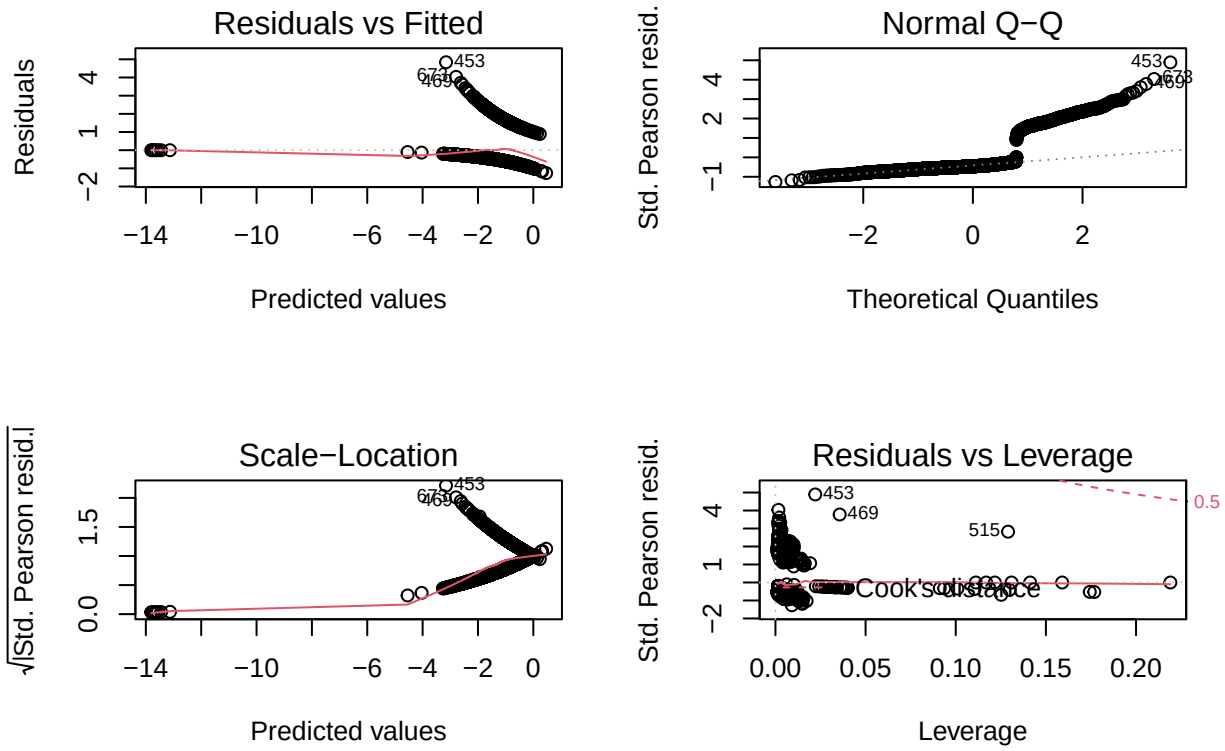
```
## mpg                             -5.002e-02  5.766e-03  -8.676  < 2e-16 ***
## years_sell                       1.161e-01  4.849e-02   2.394  0.01668 *
## fuelTypef.Fuel-Hybrid           -1.658e+00  7.485e-01  -2.215  0.02674 *
## fuelTypef.Fuel-Other            -1.261e+01  2.013e+02  -0.063  0.95003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

```
marginalModelPlots(m2)
```

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




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



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

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

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

3.4 Interactions

3.4.1 Factors interaction

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

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

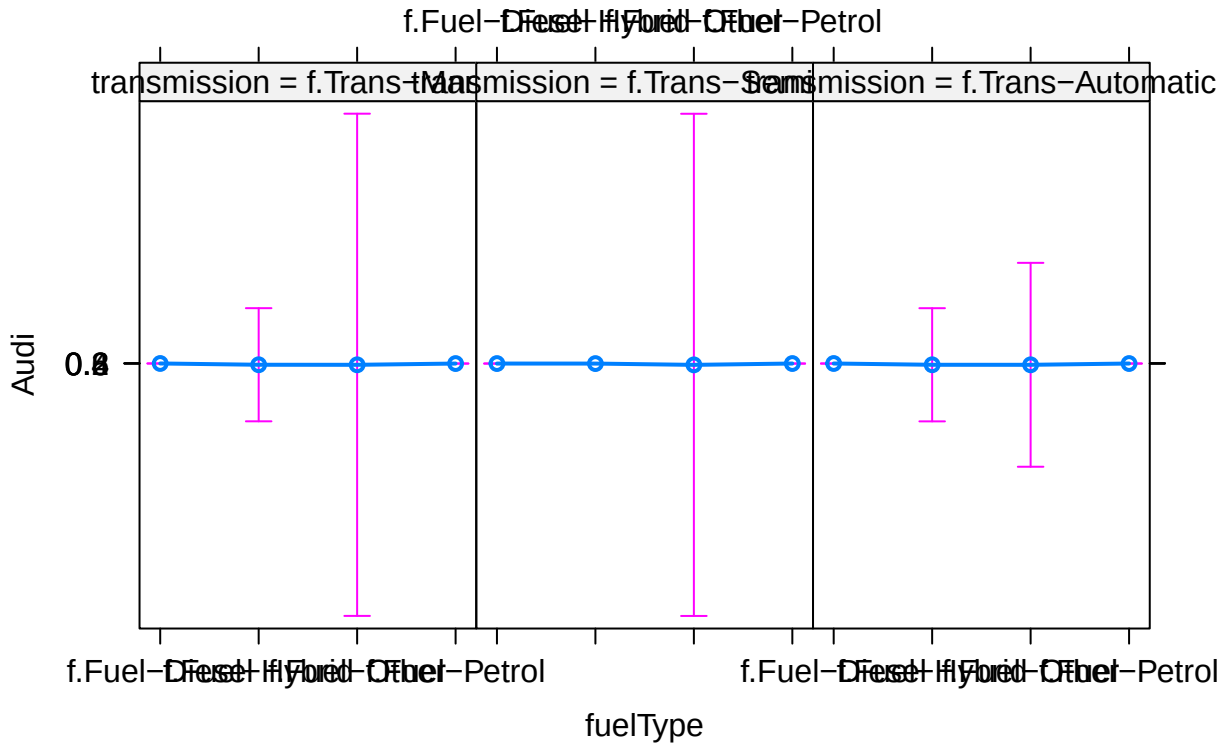
```
## Analysis of Deviance Table
##
## Model 1: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##     engineSize_num
## Model 2: Audi ~ mileage + tax + mpg + years_sell + fuelType + transmission +
##     engineSize_num + fuelType:transmission
##   Resid. Df Resid. Dev Df Deviance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

```
summary(m3)
```

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

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

```
plot(allEffects(m3), selection = 6)
```



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

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

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

3.4.2 Factor and covariate interaction

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

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

```
summary(m3)
```

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

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

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

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

3.5 Best model selection

```
m4 <- step(m3)
```

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

```
summary(m4)
```

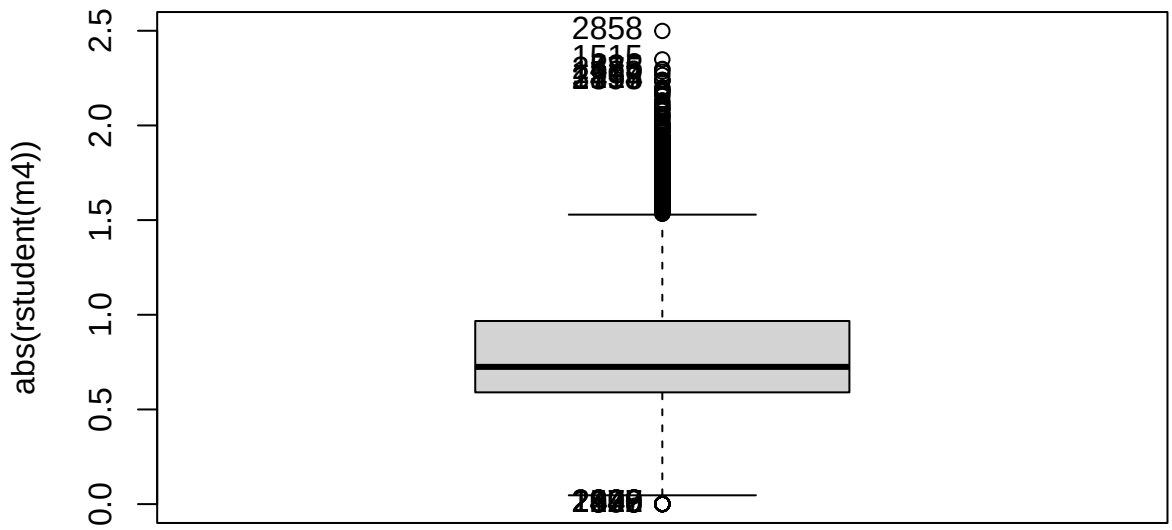
```
##
## Call:
## glm(formula = Audi ~ years_sell + fuelType + transmission + mpg +
##     engineSize_num + fuelType:transmission + fuelType:mpg, family = "binomial",
##     data = df_train[!df_train$mout == "YesMOut", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5446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

```
##                                                  Estimate Std. Error
## (Intercept)                                     1.937e-01  1.153e+00
## years_sell                                      1.447e-01  2.877e-02
## fuelTypef.Fuel-Hybrid                          -2.415e+01  3.254e+02
## fuelTypef.Fuel-Other                           -1.599e+01  2.937e+03
## fuelTypef.Fuel-Petrol                          -1.942e+00  5.879e-01
## transmissionf.Trans-SemiAuto                   -8.147e-01  1.627e-01
## transmissionf.Trans-Automatic                  -4.493e-01  1.657e-01
## mpg                                            -5.767e-02  6.781e-03
## engineSize_nummedium_engine                     5.385e-01  1.088e+00
## engineSize_numsmall_engine                      1.660e+00  1.085e+00
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto  1.478e+01  3.251e+02
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto         NA         NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto  9.998e-01  2.263e-01
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic        NA         NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic  4.617e-01  1.573e+03
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic 6.572e-01  2.525e-01
## fuelTypef.Fuel-Hybrid:mpg                       1.730e-01  2.085e-01
## fuelTypef.Fuel-Other:mpg                        1.930e-02  5.239e+01
## fuelTypef.Fuel-Petrol:mpg                       2.133e-02  1.031e-02
##                                                z value Pr(>|z|)
## (Intercept)                                      0.168 0.866571
## years_sell                                       5.031 4.87e-07 ***
## fuelTypef.Fuel-Hybrid                           -0.074 0.940835
## fuelTypef.Fuel-Other                            -0.005 0.995658
## fuelTypef.Fuel-Petrol                           -3.303 0.000958 ***
## transmissionf.Trans-SemiAuto                    -5.006 5.55e-07 ***
## transmissionf.Trans-Automatic                   -2.711 0.006715 **
## mpg                                             -8.505  < 2e-16 ***
## engineSize_nummedium_engine                      0.495 0.620741
## engineSize_numsmall_engine                       1.531 0.125790
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-SemiAuto   0.045 0.963735
## fuelTypef.Fuel-Other:transmissionf.Trans-SemiAuto      NA       NA
## fuelTypef.Fuel-Petrol:transmissionf.Trans-SemiAuto   4.418 9.95e-06 ***
## fuelTypef.Fuel-Hybrid:transmissionf.Trans-Automatic     NA       NA
## fuelTypef.Fuel-Other:transmissionf.Trans-Automatic   0.000 0.999766
## fuelTypef.Fuel-Petrol:transmissionf.Trans-Automatic  2.603 0.009244 **
## fuelTypef.Fuel-Hybrid:mpg                        0.830 0.406672
## fuelTypef.Fuel-Other:mpg                         0.000 0.999706
## fuelTypef.Fuel-Petrol:mpg                        2.068 0.038614 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3288.2  on 3169  degrees of freedom
## Residual deviance: 3116.5  on 3153  degrees of freedom
## AIC: 3150.5
##
## Number of Fisher Scoring iterations: 14
```

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

3.6 Diagnostics

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

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




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

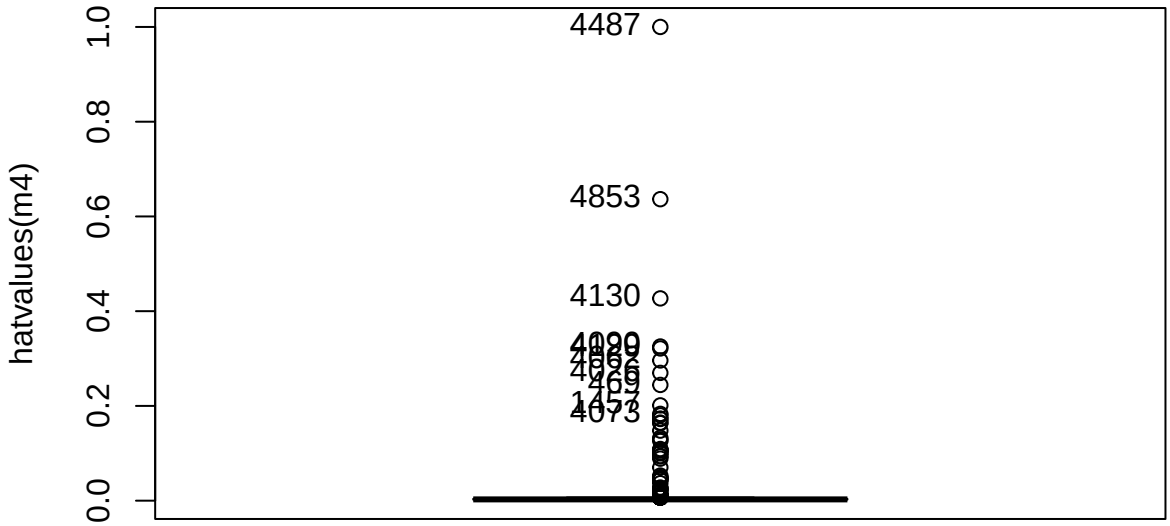
```
llres <- which(abs(rstudent(m4))>2.3);llres
```

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

```
length(llres)
```

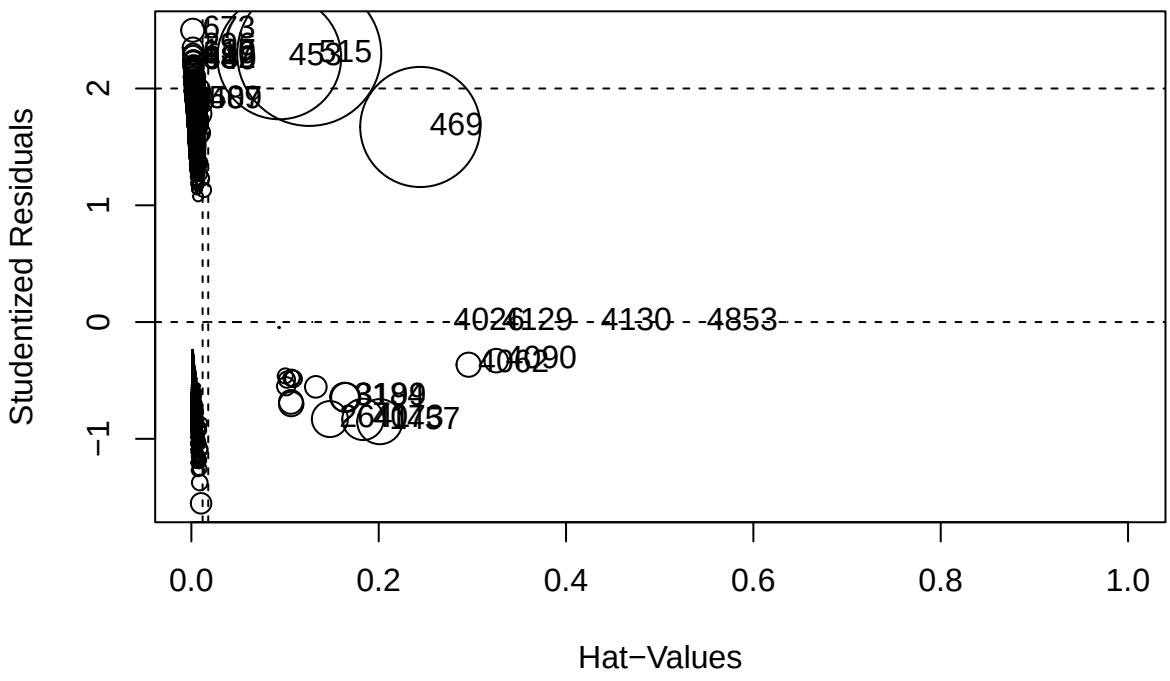
```
## [1] 2
```

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




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

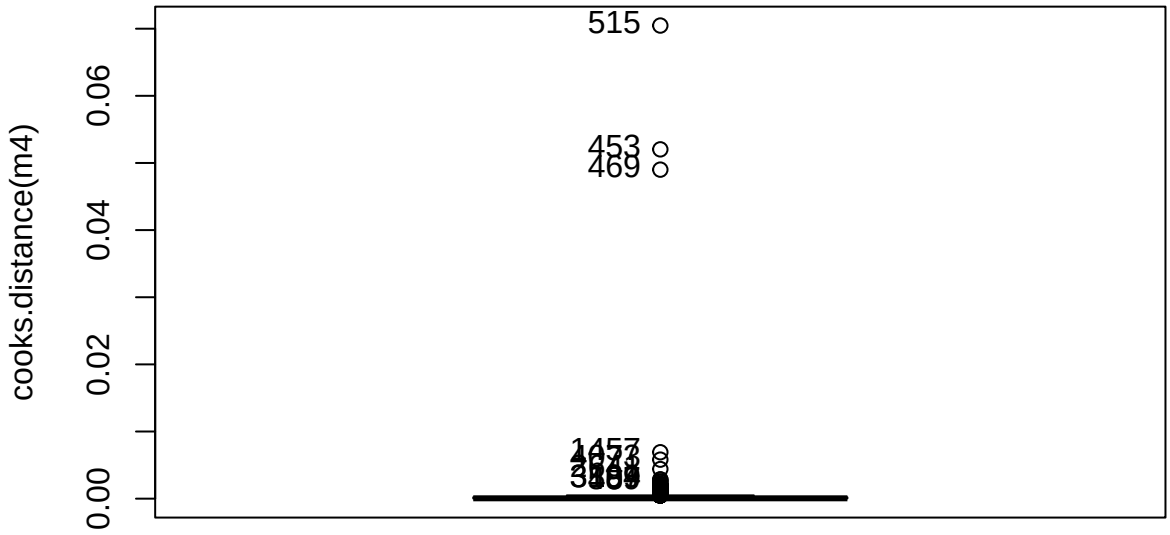
```
influencePlot(m4, id=list(n=10))
```




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

```
## 3184 -0.6428063500 0.163562794 2.820319e-03
## 489   1.8999970558 0.008552104 2.534071e-03
## 515   2.2964501221 0.125831943 7.046707e-02
## 137   2.2983808864 0.001786319 1.357241e-03
## 2641 -0.8307120179 0.147868032 4.404702e-03
## 3199 -0.6463170903 0.164770856 2.880610e-03
## 1457 -0.8529318526 0.201149886 6.911298e-03
## 4062 -0.3653341578 0.295962419 1.981350e-03
## 4090 -0.3305089636 0.325434897 1.884224e-03
## 682   2.2388520430 0.001800724 1.183851e-03
## 296   2.3488654494 0.001662117 1.430953e-03
## 453   2.2664506546 0.093850400 5.202563e-02
## 4487           NaN 1.000000000          NaN
## 4026 -0.0006439113 0.269771914 5.207606e-09
## 507   1.8999970558 0.008552104 2.534071e-03
## 4853 -0.0007232717 0.636405952 3.949903e-08
## 4073 -0.8336172288 0.183226560 5.808469e-03
## 4129 -0.0007733337 0.321305178 9.921021e-09
## 610   2.2837248410 0.001691206 1.240339e-03
## 98    2.2405351894 0.002101929 1.385813e-03
## 469   1.6705719361 0.244366660 4.900554e-02
## 673   2.4989786627 0.001435293 1.808261e-03
## 640   2.2314071531 0.002233374 1.439725e-03
## 439   2.2405351894 0.002101929 1.385813e-03
```

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




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

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

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

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

```
## [1]  525 1539 2793 1515 2858
```

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

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

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

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

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

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

These model has a deviance of 3088.5.

3.7 Prediction

```
library(ResourceSelection)
```

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

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

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

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

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

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

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

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

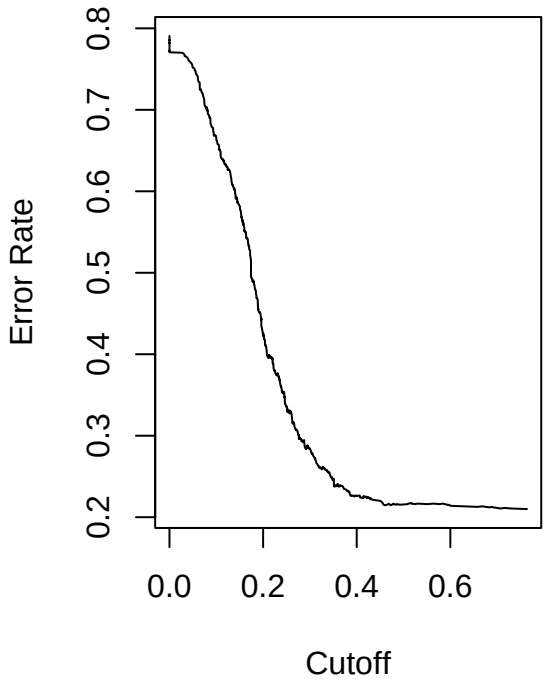
```
## AUC 0.3.2
```

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

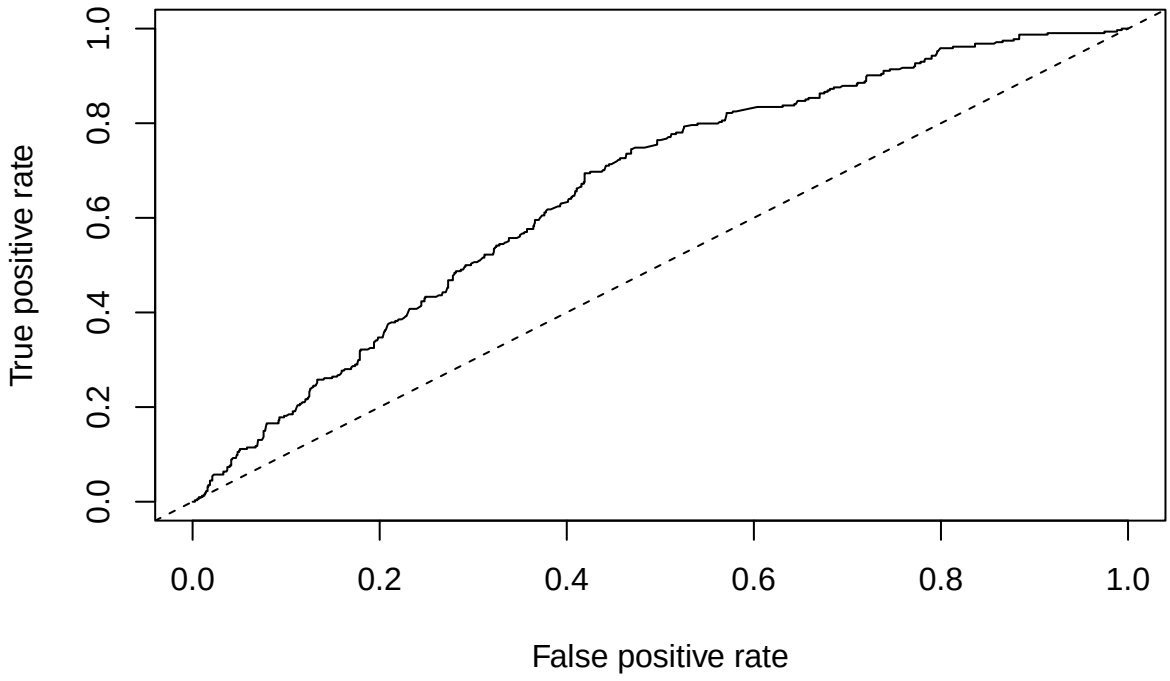
```
dadesroc<-prediction(pred_test,df_test$Audi)
par(mfrow=c(1,2))
performance(dadesroc,"auc",fpr.stop=0.05)
```

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

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




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




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

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

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

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

3.8 Confusion matrix

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

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

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

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

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

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

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

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

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