Deliverable 3

Numeric and Binary targets Forecasting Models for numeric and categorical target

Lorenzo Ricci and Raul Bometon

January 8, 2024

Contents

#Set up

```
# Clear plots
if(!is.null(dev.list())) dev.off()
# Clean workspace
rm(list=ls())
#Set working directory
setwd("C:/Users/renzo/Documents/ADEI")
filepath<-"C:/Users/renzo/Documents/ADEI/"
```

##Loading Required Packages for this deliverable

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

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

package.check <- lapply(requiredPackages, FUN = function(x) {
```

```
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})

search()
```

0.1 Load processed data from first deliverable

```
load(paste0(filepath,"Deliverable2_Result_Data.RData"))
```

#Refactor

```
names(df)[names(df) == "model"] <- "f.model"
names(df)[names(df) == "year"] <- "f.year"
names(df)[names(df) == "price"] <- "target.price"
names(df)[names(df) == "transmission"] <- "f.transmission"
names(df)[names(df) == "mileage"] <- "q.mileage"
names(df)[names(df) == "fuelType"] <- "f.fuel_type"
names(df)[names(df) == "tax"] <- "q.tax"
names(df)[names(df) == "mpg"] <- "q.mpg"
names(df)[names(df) == "engineSize"] <- "q.engine_size"
names(df)[names(df) == "manufacturer"] <- "f.manufacturer"
names(df)[names(df) == "age"] <- "q.age"
names(df)[names(df) == "auxPrice"] <- "f.aux_price"
names(df)[names(df) == "auxTax"] <- "f.aux_tax"
names(df)[names(df) == "auxMileage"] <- "f.used"
names(df)[names(df) == "auxMpg"] <- "f.efficiency"
names(df)[names(df) == "auxAge"] <- "f.old"
names(df)[names(df) == "auxEngineSize"] <- "f.aux_EngineSize"
names(df)[names(df) == "Audi"] <- "target.audi"

df$target.audi <- factor(df$target.audi)

ll<-which(df$q.age==0);
df$q.age[ll]<-0.1

ll<-which(df$q.tax==0);
df$q.tax[ll]<-0.1
```

We rename auxMpg, auxMileage and auxAge for easier analysis.

#Listing variables

```
names(df)
```

```
##  [1] "f.model"          "f.year"           "target.price"     "f.transmission"
##  [5] "q.mileage"        "f.fuel_type"      "q.tax"            "q.mpg"
##  [9] "q.engine_size"    "f.manufacturer"   "q.age"            "mout"
## [13] "f.aux_price"      "f.aux_tax"        "f.used"           "f.efficiency"
## [17] "f.old"            "f.aux_EngineSize" "target.audi"
```

```
vars_con<-names(df)[c(5,7:9,11)]
vars_dis<-names(df)[c(1:2,4,6,10,13:18)]
vars_res<-names(df)[c(3,19)]
```

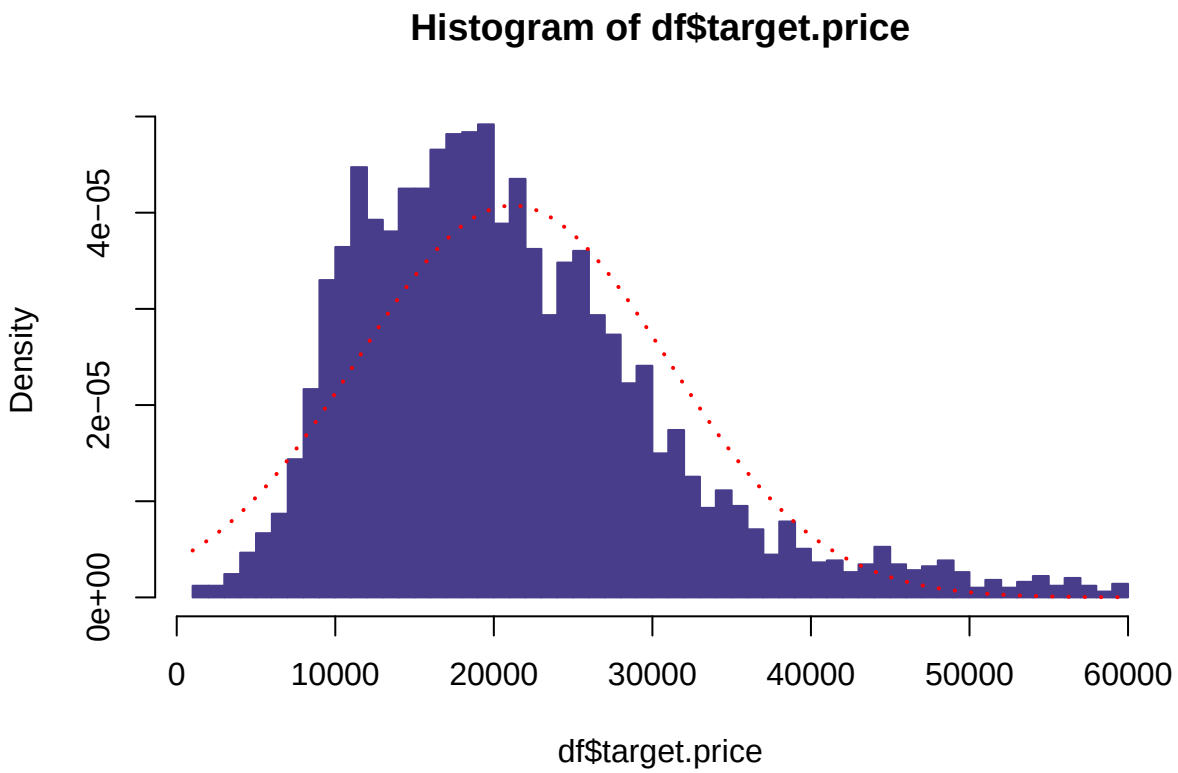
---

1 Quantitative Logistics Regression

Before we begin to see correlations with our target, we should consider the normality of this.

1.1 (0) Normality

```
hist(df$target.price,50,freq=F,col="darkslateblue",border = "darkslateblue")
mm<-mean(df$target.price);ss<-sd(df$target.price)
curve(dnorm(x,mean=mm,sd=ss),col="red",lwd=2,lty=3, add=T)
```




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

```
##
##  Shapiro-Wilk normality test
##
## data:  df$target.price
## W = 0.93984, p-value < 2.2e-16
```

We see that the target price is not normally distributed for the following reasons:

- graph: we can see that there is no symmetry in the plot
- shapiro: we see that the p-value is too small to accept the assumption that target.price is normally distributed

1.1.1 Symmetry

```
skewness(df$target.price)
```

```
## [1] 1.049015
```

Normal data should have 0 skewness: we see that our data is right skewed (1.049).

1.1.2 Kurtosis

```
kurtosis(df$target.price)
```

```
## [1] 4.431356
```

We get a leptokurtic data distribtion. Normal data should have a value of 3, target.price has more (4.43).

1.2 Numerical variables

We use spearman method since out target is not normally distributed

```
round(cor(df[,c("target.price",vars_con)], method="spearman"),dig=2)
```

```
##                target.price q.mileage q.tax q.mpg q.engine_size q.age
## target.price           1.00     -0.64 -0.03 -0.56          0.48 -0.68
## q.mileage             -0.64      1.00  0.18  0.39          0.08  0.85
## q.tax                 -0.03      0.18  1.00 -0.19          0.22  0.21
## q.mpg                 -0.56      0.39 -0.19  1.00         -0.22  0.38
## q.engine_size          0.48      0.08  0.22 -0.22          1.00  0.05
## q.age                 -0.68      0.85  0.21  0.38          0.05  1.00
```

Negative Correlations:

- "q.mileage" has a moderately strong negative correlation with "target.price" (-0.64).
- "q.mpg" also has a moderately strong negative correlation with "target.price" (-0.56).
- "q.age" has a strong negative correlation with "target.price" (-0.68).

Positive Correlations:

- "q.engine_size" has a moderate positive correlation with "target.price" (0.48).
- "q.tax" has weak positive correlation with "target.price."

We will select the most correlated to make an initial model.

```
vars_cexp<-names(df)[c(5,8,9,11)]
```

1.2.1 Model 1

```
model_1 <- lm(target.price~.,data=df[,c("target.price",vars_cexp)]);summary(model_1)
```

```
##
## Call:
## lm(formula = target.price ~ ., data = df[, c("target.price",
##     vars_cexp)])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -20896  -2710   -140   2343  32723
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.922e+04  5.058e+02   38.00   <2e-16 ***
## q.mileage     -1.095e-01  6.154e-03  -17.80   <2e-16 ***
## q.mpg         -1.604e+02  6.805e+00  -23.57   <2e-16 ***
## q.engine_size  9.669e+03  1.391e+02   69.49   <2e-16 ***
```

```
## q.age          -2.022e+03  5.977e+01  -33.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4782 on 4935 degrees of freedom
## Multiple R-squared:  0.7616, Adjusted R-squared:  0.7614
## F-statistic:  3941 on 4 and 4935 DF,  p-value: < 2.2e-16
```

Model 1 explains 76.19% of the variability of the target.We see that this might be a good fit but further investigation is needed to make that statement.

First we want to see the variance inflation factor, if its greater than 5, we need to consider whether or not we keep a variable.

```
vif(model_1)
```

```
##     q.mileage         q.mpg q.engine_size         q.age
##      2.985565      1.322409      1.185404      2.871907
```

There is no vif that exceeds 5 so we can continue with these numerical explicative variables.

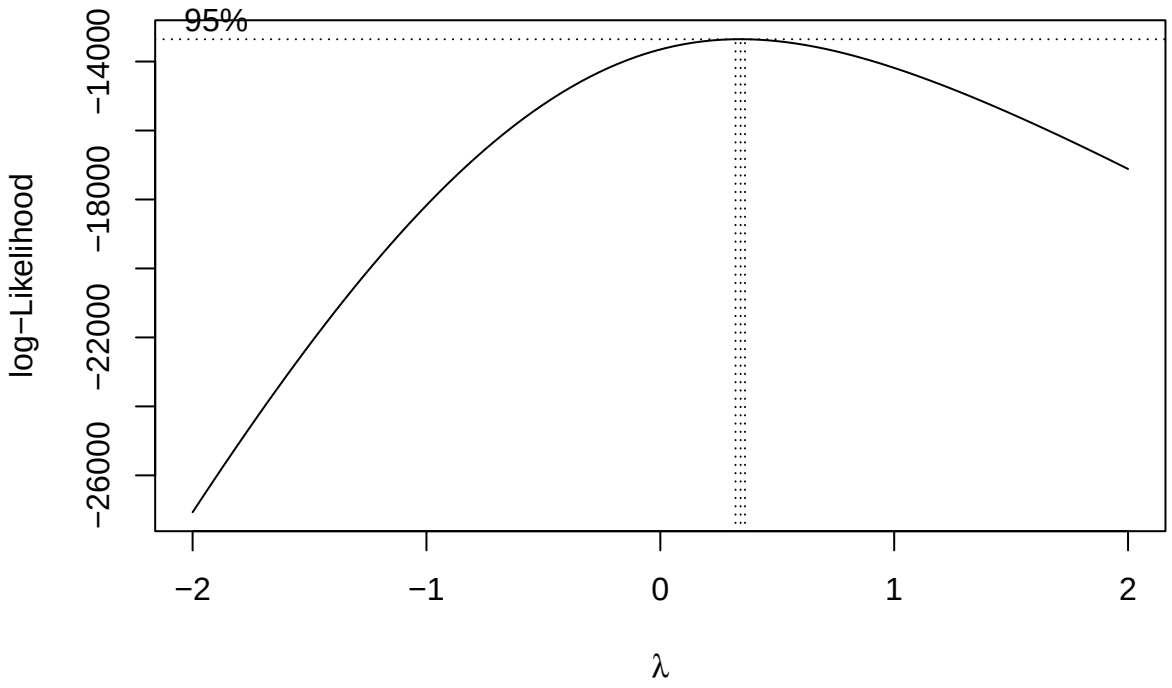
1.2.2 Target variable transformation?

```
library(MASS)
```

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

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

```
boxcox(target.price~q.mileage+q.mpg+q.age+q.engine_size,data=df)
```



Since lambda is very close to 0 logarithmic transformation is needed.

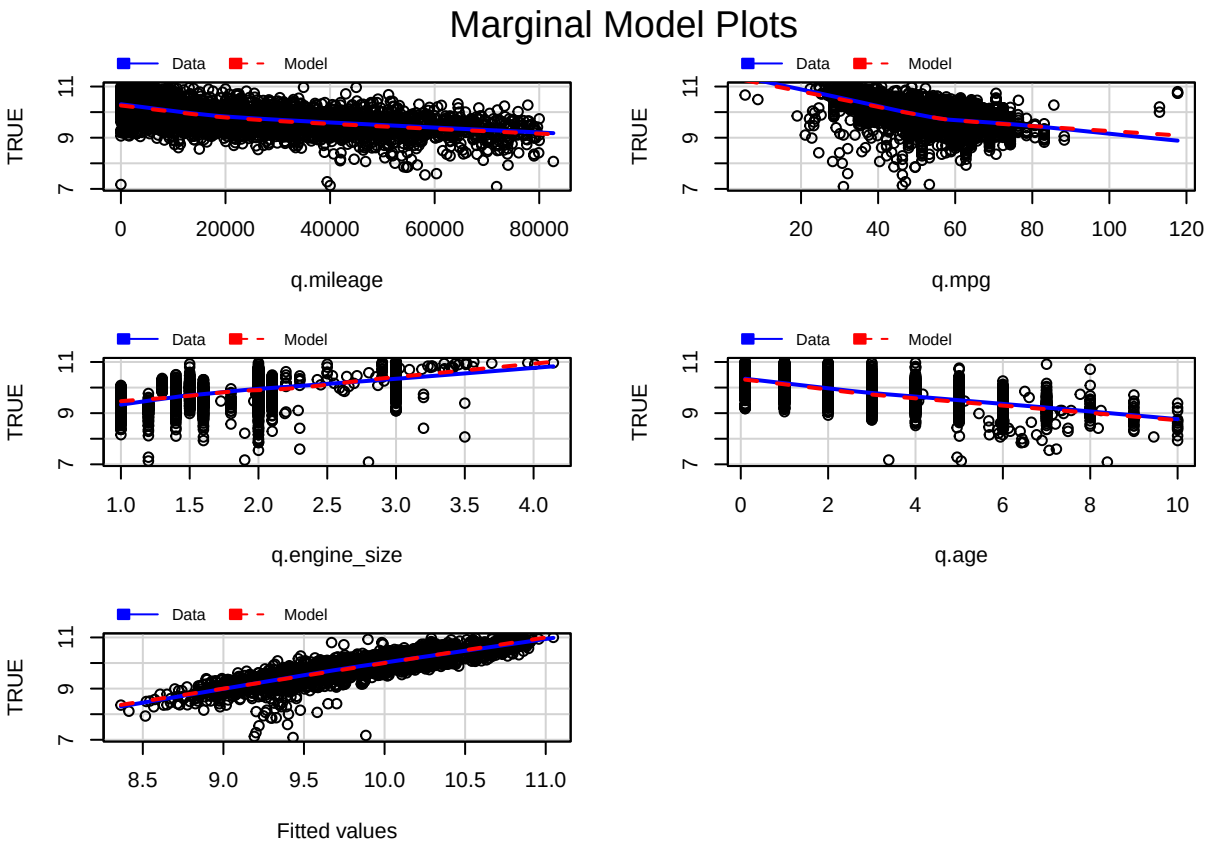
1.2.3 New model

```
model_2 <- lm(log(target.price)~.,data=df[,c("target.price",vars_cexp)]);
summary(model_2)
```

```
##
## Call:
## lm(formula = log(target.price) ~ ., data = df[, c("target.price",
##     vars_cexp)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.71475 -0.11578  0.01409  0.13946  1.12819
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.692e+00  2.386e-02  406.21   <2e-16 ***
## q.mileage     -5.101e-06  2.903e-07  -17.57   <2e-16 ***
## q.mpg         -4.999e-03  3.210e-04  -15.57   <2e-16 ***
## q.engine_size  4.585e-01  6.563e-03   69.86   <2e-16 ***
## q.age         -1.218e-01  2.819e-03  -43.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2256 on 4935 degrees of freedom
## Multiple R-squared:  0.7783, Adjusted R-squared:  0.7781
## F-statistic:  4330 on 4 and 4935 DF,  p-value: < 2.2e-16
```

Let's now discriminate the variables independently:

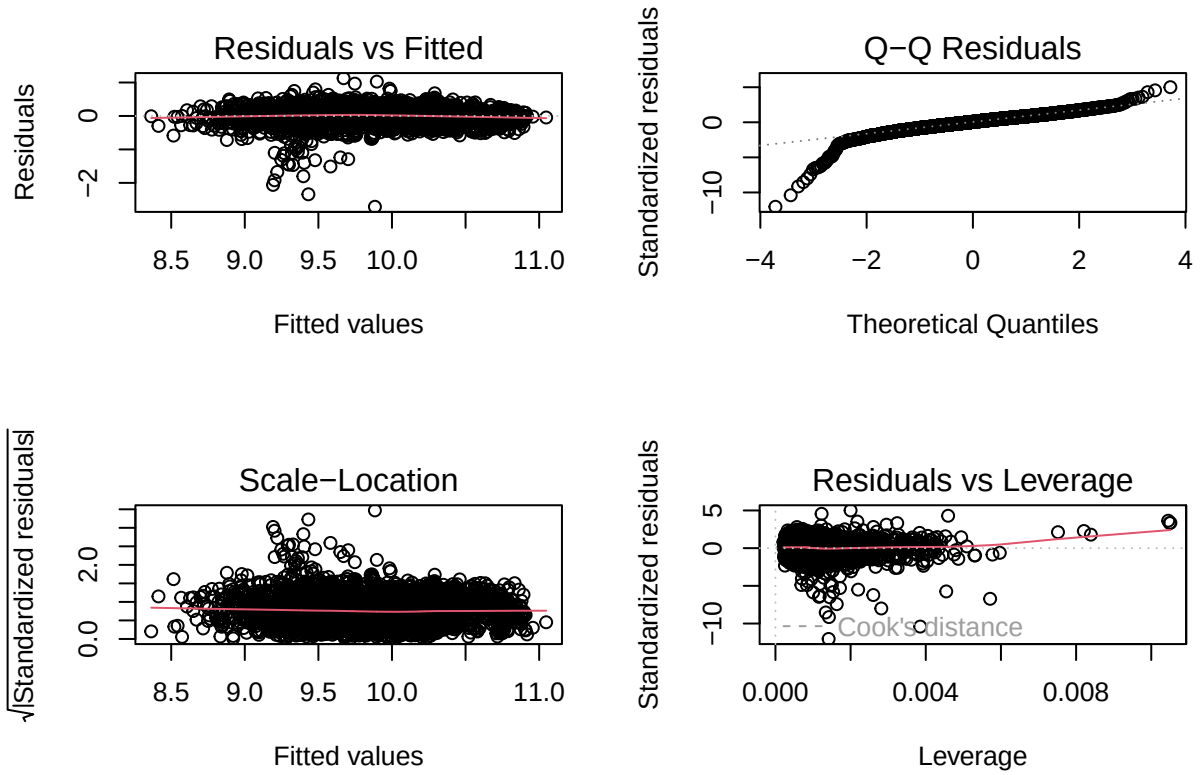
```
marginalModelPlots(model_2)
```



The relationships between the target variable (log(price)) and the predictor variables (mileage, engineSize, mpg, and age) are all non-linear.

1.2.4 Diagnostics

```
par(mfrow=c(2,2))
plot(model_2, id.n=0 )
```




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

The residuals are very close to the fitted values, and the theoretical quantities are very close to the fitted values. This suggests that your model is a good fit for the data.

The Q-Q residuals plot shows that the distribution of the residuals is very close to a normal distribution. This is a good sign, as it indicates that the assumptions of linear regression are met.

The scale-location plot shows that the variance of the residuals is constant across the range of fitted values. This is another good sign, as it indicates that the model is not heteroskedastic.

The residuals vs leverage plot shows that there are no outliers in the data. This is important, as outliers can have a significant impact on the results of linear regression.

Overall, the first results suggest that our model is a good fit for the data and that the assumptions of linear regression are met. This is a good starting point.

1.2.5 Transformations to my regresors

```
boxTidwell(log(target.price)~q.mileage+q.engine_size+q.mpg+q.age,data=df[!df$mout=="YesMOut",])
```

```
##                MLE of lambda Score Statistic (t)  Pr(>|t|)
## q.mileage            0.89810              0.8316    0.4057
## q.engine_size       -0.45068            -21.3049 < 2.2e-16 ***
## q.mpg                0.22325              4.9149 9.172e-07 ***
## q.age                1.29321             -5.5808 2.523e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## iterations =  9
##
## Score test for null hypothesis that all lambdas = 1:
## F = 123.58, df = 4 and 4877, Pr(>F) = < 2.2e-16
```

We apply the transformations suggested my boxTidwell

```
model_3<-lm(log(target.price)~q.mileage+log(q.engine_size)+sqrt(q.mpg)+q.age,data=df[!df$mout=="YesMOut"
summary(model_3)
```

```
##
## Call:
## lm(formula = log(target.price) ~ q.mileage + log(q.engine_size) +
##     sqrt(q.mpg) + q.age, data = df[!df$mout == "YesMOut", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74750 -0.11142  0.00646  0.13129  1.06773
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.045e+01  3.329e-02  313.92   <2e-16 ***
## q.mileage           -5.145e-06  2.719e-07  -18.92   <2e-16 ***
## log(q.engine_size)   8.534e-01  1.137e-02   75.08   <2e-16 ***
## sqrt(q.mpg)         -9.454e-02  4.360e-03  -21.68   <2e-16 ***
## q.age               -1.155e-01  2.632e-03  -43.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2083 on 4881 degrees of freedom
## Multiple R-squared:  0.8019, Adjusted R-squared:  0.8018
## F-statistic:  4940 on 4 and 4881 DF,  p-value: < 2.2e-16
```

The model is explaining 80.12% of the variance in the target variable.

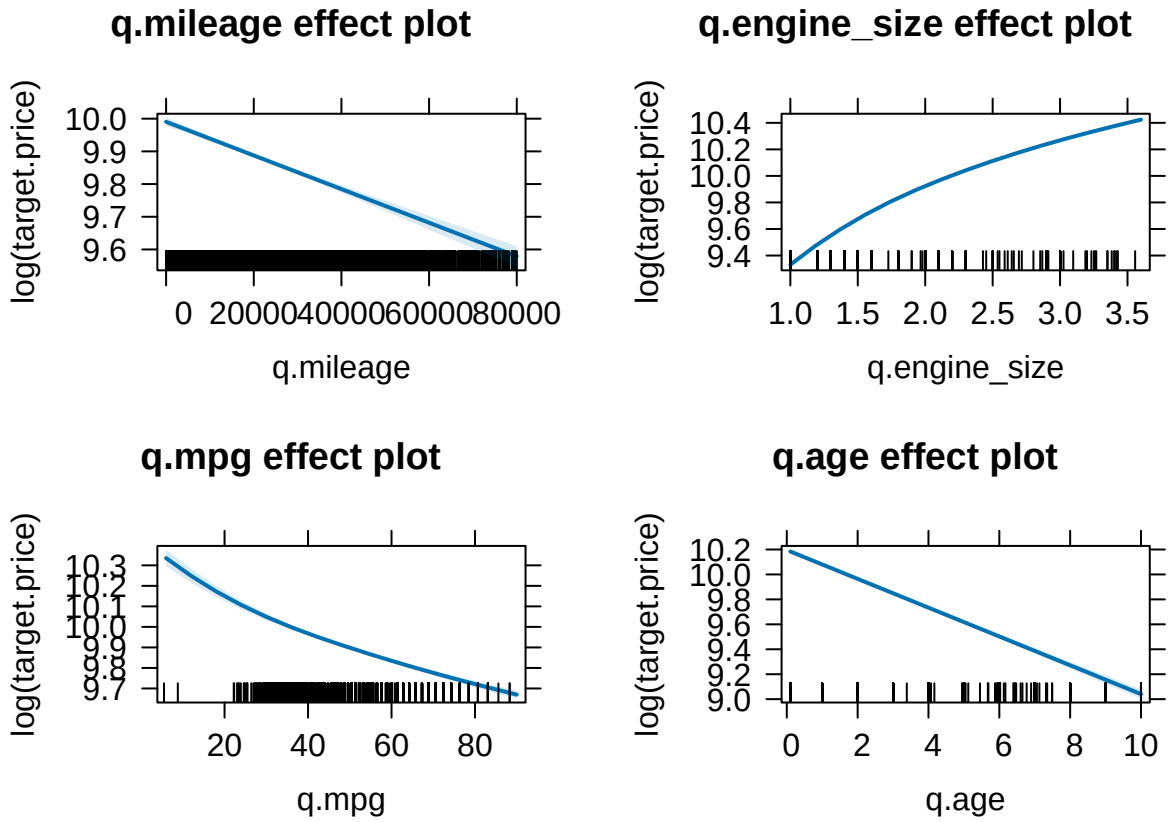
Let's look at the effects of this model:

```
Anova(model_3)
```

```
## Anova Table (Type II tests)
##
## Response: log(target.price)
##                      Sum Sq   Df F value    Pr(>F)
## q.mileage            15.541    1  358.01 < 2.2e-16 ***
## log(q.engine_size) 244.718    1 5637.42 < 2.2e-16 ***
## sqrt(q.mpg)          20.407    1  470.11 < 2.2e-16 ***
## q.age                83.592    1 1925.67 < 2.2e-16 ***
## Residuals           211.883 4881
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-values are all less than 0.001, which is the standard threshold for statistical significance, so the net effects are significant

```
library(effects)
plot(allEffects(model_3))
```

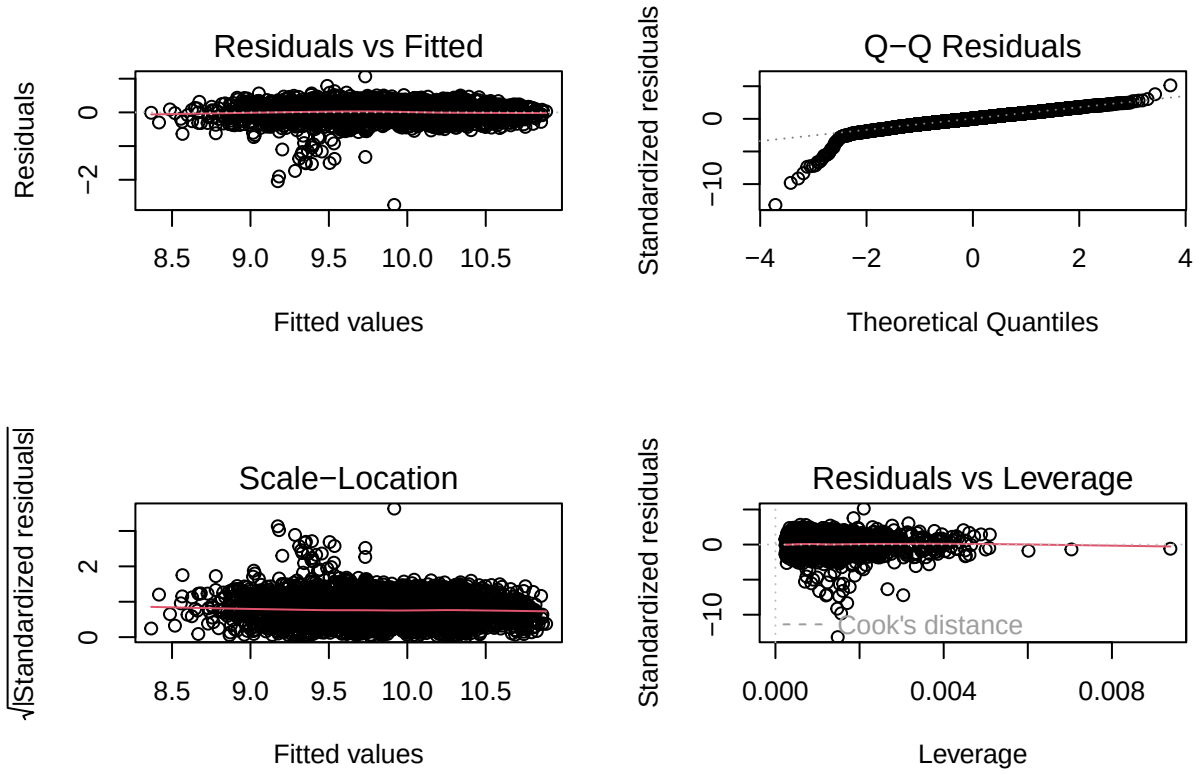


*The interaction effect between mileage and engine size is positive. This means that the relationship between mileage and log(price) is stronger for cars with larger engines.

*The interaction effect between mpg and age is also positive. This means that the relationship between mpg and log(price) is stronger for older cars.

*The three-way interaction between mileage, engine size, and mpg is also positive. This means that the combined effect of these three predictor variables on log(price) is more than the sum of their individual effects.

```
par(mfrow=c(2,2))
plot(model_3, id.n=0 )
```




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

The residues are not completely optimal. We will try with a sqrt of the target.

```
model_4<-lm(sqrt(target.price)~q.mileage+log(q.engine_size)+sqrt(q.mpg)+q.age,data=df[!df$mout=="YesMOut
summary(model_4)
```

```
##
## Call:
## lm(formula = sqrt(target.price) ~ q.mileage + log(q.engine_size) +
##     sqrt(q.mpg) + q.age, data = df[!df$mout == "YesMOut", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -110.357   -8.341   -0.111    8.283   84.285
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.952e+02  2.206e+00   88.48   <2e-16 ***
## q.mileage           -3.588e-04  1.802e-05  -19.91   <2e-16 ***
## log(q.engine_size)   5.793e+01  7.532e-01   76.92   <2e-16 ***
## sqrt(q.mpg)         -8.422e+00  2.889e-01  -29.15   <2e-16 ***
## q.age               -7.225e+00  1.744e-01  -41.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.81 on 4881 degrees of freedom
## Multiple R-squared:  0.812,  Adjusted R-squared:  0.8119
## F-statistic:  5271 on 4 and 4881 DF,  p-value: < 2.2e-16
```

We win a little in variance but increase the residues a lot. We will continue with model_3.

1.3 Factors

Adding variable manufacturer to our model.

```r
model_5<- update(model_3, ~.+f.manufacturer,data=df[!df$mout=="YesMOut",])
summary(model_5)
```

```
##
## Call:
## lm(formula = log(target.price) ~ q.mileage + log(q.engine_size) +
##     sqrt(q.mpg) + q.age + f.manufacturer, data = df[!df$mout ==
##     "YesMOut", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62121 -0.09902  0.01088  0.11120  1.04013
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.069e+01  3.220e-02 332.134  < 2e-16 ***
## q.mileage               -4.877e-06  2.517e-07 -19.372  < 2e-16 ***
## log(q.engine_size)       7.338e-01  1.205e-02  60.895  < 2e-16 ***
## sqrt(q.mpg)             -1.106e-01  4.150e-03 -26.655  < 2e-16 ***
## q.age                   -1.148e-01  2.427e-03 -47.279  < 2e-16 ***
## f.manufacturerBMW       -7.968e-02  8.608e-03  -9.257  < 2e-16 ***
## f.manufacturerMercedes   3.527e-02  8.281e-03   4.259 2.09e-05 ***
## f.manufacturerVW        -1.785e-01  8.101e-03 -22.028  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1921 on 4878 degrees of freedom
## Multiple R-squared:  0.8317, Adjusted R-squared:  0.8315
## F-statistic:  3444 on 7 and 4878 DF,  p-value: < 2.2e-16
```

We can see an improvement of variance to 83.17%

Adding variable transmission to our model.

```r
model_6<- update(model_5, ~.+f.transmission,data=df[!df$mout=="YesMOut",])
summary(model_6)
```

```
##
## Call:
## lm(formula = log(target.price) ~ q.mileage + log(q.engine_size) +
##     sqrt(q.mpg) + q.age + f.manufacturer + f.transmission, data = df[!df$mout ==
##     "YesMOut", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.55652 -0.09353  0.00592  0.10906  1.00762
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.058e+01  3.191e-02 331.471   <2e-16 ***
## q.mileage                -4.694e-06  2.446e-07 -19.187   <2e-16 ***
## log(q.engine_size)        6.583e-01  1.245e-02  52.888   <2e-16 ***
## sqrt(q.mpg)              -1.007e-01  4.066e-03 -24.765   <2e-16 ***
## q.age                    -1.109e-01  2.365e-03 -46.885   <2e-16 ***
## f.manufacturerBMW        -9.219e-02  8.378e-03 -11.004   <2e-16 ***
## f.manufacturerMercedes    5.401e-03  8.208e-03   0.658    0.511
## f.manufacturerVW         -1.640e-01  7.900e-03 -20.763   <2e-16 ***
## f.transmissionSemiAuto    1.304e-01  7.635e-03  17.079   <2e-16 ***
## f.transmissionAutomatic   1.167e-01  8.102e-03  14.401   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1862 on 4876 degrees of freedom
## Multiple R-squared:  0.8419, Adjusted R-squared:  0.8416
## F-statistic:  2885 on 9 and 4876 DF,  p-value: < 2.2e-16
```

We can see an improvement of variance to 84.19%

1.4 Interactions

```
model_7<-lm(log(target.price)~(q.mileage+log(q.engine_size)+sqrt(q.mpg)+q.age)*(f.manufacturer+f.transmi
model_7<-step( model_7, k=log(nrow(df)))
```

```
## Start:  AIC=-16298.26
## log(target.price) ~ (q.mileage + log(q.engine_size) + sqrt(q.mpg) +
##     q.age) * (f.manufacturer + f.transmission)
##
##                                     Df Sum of Sq    RSS    AIC
## - q.age:f.manufacturer               3   0.21288 165.25 -16318
## - q.mileage:f.manufacturer           3   0.31242 165.35 -16314
## - log(q.engine_size):f.transmission  2   0.04504 165.09 -16314
## - q.mileage:f.transmission           2   0.09022 165.13 -16313
## - sqrt(q.mpg):f.manufacturer         3   0.44532 165.49 -16311
## - q.age:f.transmission               2   0.26782 165.31 -16307
## - sqrt(q.mpg):f.transmission         2   0.30620 165.35 -16306
## <none>                                           165.04 -16298
## - log(q.engine_size):f.manufacturer  3   1.19152 166.23 -16289
##
## Step:  AIC=-16317.48
## log(target.price) ~ q.mileage + log(q.engine_size) + sqrt(q.mpg) +
##     q.age + f.manufacturer + f.transmission + q.mileage:f.manufacturer +
##     q.mileage:f.transmission + log(q.engine_size):f.manufacturer +
##     log(q.engine_size):f.transmission + sqrt(q.mpg):f.manufacturer +
##     sqrt(q.mpg):f.transmission + q.age:f.transmission
##
##                                     Df Sum of Sq    RSS    AIC
## - log(q.engine_size):f.transmission  2   0.04533 165.30 -16333
## - q.mileage:f.transmission           2   0.10660 165.36 -16331
## - sqrt(q.mpg):f.manufacturer         3   0.46511 165.72 -16329
## - q.age:f.transmission               2   0.27419 165.53 -16326
## - sqrt(q.mpg):f.transmission         2   0.31316 165.57 -16325
## - q.mileage:f.manufacturer           3   0.73241 165.99 -16321
## <none>                                           165.25 -16318
## - log(q.engine_size):f.manufacturer  3   1.21090 166.47 -16307
##
## Step:  AIC=-16333.15
## log(target.price) ~ q.mileage + log(q.engine_size) + sqrt(q.mpg) +
##     q.age + f.manufacturer + f.transmission + q.mileage:f.manufacturer +
##     q.mileage:f.transmission + log(q.engine_size):f.manufacturer +
##     sqrt(q.mpg):f.manufacturer + sqrt(q.mpg):f.transmission +
##     q.age:f.transmission
##
##                                     Df Sum of Sq    RSS    AIC
## - q.mileage:f.transmission           2   0.09502 165.40 -16347
## - sqrt(q.mpg):f.manufacturer         3   0.45460 165.75 -16345
## - q.age:f.transmission               2   0.27175 165.57 -16342
## - sqrt(q.mpg):f.transmission         2   0.29146 165.59 -16342
## - q.mileage:f.manufacturer           3   0.71850 166.02 -16338
## <none>                                           165.30 -16333
## - log(q.engine_size):f.manufacturer  3   1.41151 166.71 -16317
##
## Step:  AIC=-16347.35
## log(target.price) ~ q.mileage + log(q.engine_size) + sqrt(q.mpg) +
```

```
##     q.age + f.manufacturer + f.transmission + q.mileage:f.manufacturer +
##     log(q.engine_size):f.manufacturer + sqrt(q.mpg):f.manufacturer +
##     sqrt(q.mpg):f.transmission + q.age:f.transmission
##
##                                     Df Sum of Sq    RSS    AIC
## - sqrt(q.mpg):f.manufacturer         3   0.41459 165.81 -16361
## - sqrt(q.mpg):f.transmission         2   0.23561 165.63 -16357
## - q.mileage:f.manufacturer           3   0.64221 166.04 -16354
## <none>                                           165.40 -16347
## - q.age:f.transmission               2   1.09036 166.49 -16332
## - log(q.engine_size):f.manufacturer  3   1.39866 166.79 -16332
##
## Step:  AIC=-16360.64
## log(target.price) ~ q.mileage + log(q.engine_size) + sqrt(q.mpg) +
##     q.age + f.manufacturer + f.transmission + q.mileage:f.manufacturer +
##     log(q.engine_size):f.manufacturer + sqrt(q.mpg):f.transmission +
##     q.age:f.transmission
##
##                                     Df Sum of Sq    RSS    AIC
## - sqrt(q.mpg):f.transmission         2   0.08429 165.89 -16375
## - q.mileage:f.manufacturer           3   0.43359 166.24 -16373
## <none>                                           165.81 -16361
## - q.age:f.transmission               2   0.94014 166.75 -16350
## - log(q.engine_size):f.manufacturer  3   2.16142 167.97 -16323
##
## Step:  AIC=-16375.16
## log(target.price) ~ q.mileage + log(q.engine_size) + sqrt(q.mpg) +
##     q.age + f.manufacturer + f.transmission + q.mileage:f.manufacturer +
##     log(q.engine_size):f.manufacturer + q.age:f.transmission
##
##                                     Df Sum of Sq    RSS    AIC
## - q.mileage:f.manufacturer           3    0.4343 166.33 -16388
## <none>                                           165.89 -16375
## - q.age:f.transmission               2    0.8659 166.76 -16367
## - log(q.engine_size):f.manufacturer  3    2.1354 168.03 -16338
## - sqrt(q.mpg)                        1   22.3912 188.28 -15765
##
## Step:  AIC=-16387.9
## log(target.price) ~ q.mileage + log(q.engine_size) + sqrt(q.mpg) +
##     q.age + f.manufacturer + f.transmission + log(q.engine_size):f.manufacturer +
##     q.age:f.transmission
##
##                                     Df Sum of Sq    RSS    AIC
## <none>                                           166.33 -16388
## - q.age:f.transmission               2    0.7927 167.12 -16382
## - log(q.engine_size):f.manufacturer  3    2.1077 168.44 -16352
## - q.mileage                          1   12.8722 179.20 -16032
## - sqrt(q.mpg)                        1   22.1183 188.45 -15786
```

```
Anova(model_7)
```

```
## Anova Table (Type II tests)
##
## Response: log(target.price)
##                                  Sum Sq   Df  F value    Pr(>F)
## q.mileage                        12.872    1  376.969 < 2.2e-16 ***
## log(q.engine_size)               96.808    1 2835.058 < 2.2e-16 ***
## sqrt(q.mpg)                      22.118    1  647.742 < 2.2e-16 ***
## q.age                            74.480    1 2181.174 < 2.2e-16 ***
## f.manufacturer                   22.059    3  215.339 < 2.2e-16 ***
## f.transmission                   10.907    2  159.705 < 2.2e-16 ***
## log(q.engine_size):f.manufacturer  2.108    3   20.575 3.036e-13 ***
## q.age:f.transmission              0.793    2   11.608 9.349e-06 ***
```

```
## Residuals                          166.329 4871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(model_7)
```

```
##
## Call:
## lm(formula = log(target.price) ~ q.mileage + log(q.engine_size) +
##     sqrt(q.mpg) + q.age + f.manufacturer + f.transmission + log(q.engine_size):f.manufacturer +
##     q.age:f.transmission, data = df[!df$mout == "YesMOut", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.57672 -0.09482  0.00574  0.10692  0.98809
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                              1.068e+01  3.424e-02 311.930
## q.mileage                               -4.727e-06  2.435e-07 -19.416
## log(q.engine_size)                       5.461e-01  2.230e-02  24.482
## sqrt(q.mpg)                             -1.033e-01  4.059e-03 -25.451
## q.age                                   -1.154e-01  2.934e-03 -39.336
## f.manufacturerBMW                       -1.232e-01  2.535e-02  -4.858
## f.manufacturerMercedes                  -4.649e-02  2.294e-02  -2.026
## f.manufacturerVW                        -2.710e-01  1.658e-02 -16.345
## f.transmissionSemiAuto                   8.846e-02  1.231e-02   7.188
## f.transmissionAutomatic                  1.097e-01  1.337e-02   8.204
## log(q.engine_size):f.manufacturerBMW     6.519e-02  3.437e-02   1.897
## log(q.engine_size):f.manufacturerMercedes 9.031e-02  3.243e-02   2.785
## log(q.engine_size):f.manufacturerVW      2.037e-01  2.696e-02   7.555
## q.age:f.transmissionSemiAuto             1.593e-02  3.513e-03   4.536
## q.age:f.transmissionAutomatic            1.847e-03  3.452e-03   0.535
##                                         Pr(>|t|)
## (Intercept)                              < 2e-16 ***
## q.mileage                                < 2e-16 ***
## log(q.engine_size)                       < 2e-16 ***
## sqrt(q.mpg)                              < 2e-16 ***
## q.age                                    < 2e-16 ***
## f.manufacturerBMW                       1.22e-06 ***
## f.manufacturerMercedes                   0.04277 *
## f.manufacturerVW                         < 2e-16 ***
## f.transmissionSemiAuto                  7.55e-13 ***
## f.transmissionAutomatic                 2.94e-16 ***
## log(q.engine_size):f.manufacturerBMW     0.05792 .
## log(q.engine_size):f.manufacturerMercedes  0.00537 **
## log(q.engine_size):f.manufacturerVW     4.97e-14 ***
## q.age:f.transmissionSemiAuto            5.86e-06 ***
## q.age:f.transmissionAutomatic            0.59257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1848 on 4871 degrees of freedom
## Multiple R-squared:  0.8445, Adjusted R-squared:  0.8441
## F-statistic:  1890 on 14 and 4871 DF,  p-value: < 2.2e-16
```

---

2 Binary Logistics Regression

##Split the sample in work and test samples

```r
set.seed(1207)
llwork <- sample(1:nrow(df),round(0.80*nrow(df),0))
df_work <- df[llwork,]
df_test <-df[-llwork,]
```

##First model

```r
bm1<-glm(target.audi~q.mileage+q.tax+q.mpg+q.age,family="binomial",data=df_work)
summary(bm1)
```

```
##
## Call:
## glm(formula = target.audi ~ q.mileage + q.tax + q.mpg + q.age,
##     family = "binomial", data = df_work)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.110e+00  5.974e-01  -1.857 0.063253 .
## q.mileage    1.213e-05  3.396e-06   3.572 0.000355 ***
## q.tax        4.714e-03  3.574e-03   1.319 0.187181
## q.mpg       -2.248e-02  3.816e-03  -5.891 3.83e-09 ***
## q.age       -1.353e-03  3.398e-02  -0.040 0.968240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4070.5  on 3951  degrees of freedom
## Residual deviance: 4008.4  on 3947  degrees of freedom
## AIC: 4018.4
##
## Number of Fisher Scoring iterations: 4
```

```r
vif(bm1)
```

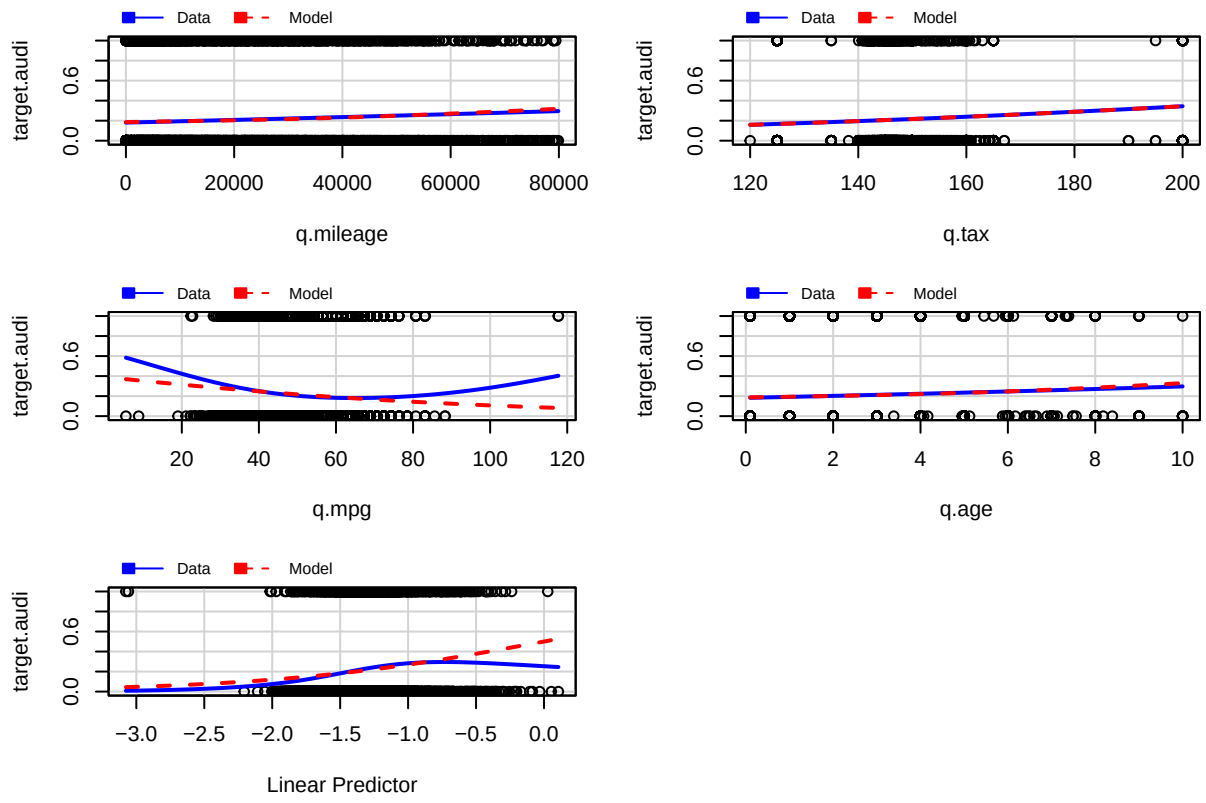
```
## q.mileage     q.tax     q.mpg     q.age
##  2.968798  1.146108  1.231897  2.953990
```

```r
Anova(bm1, test="Wald")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: target.audi
##           Df   Chisq Pr(>Chisq)
## q.mileage  1 12.7569  0.0003547 ***
## q.tax      1  1.7397  0.1871811
## q.mpg      1 34.7065  3.833e-09 ***
## q.age      1  0.0016  0.9682400
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

*Mileage and mpg are the most important predictors, while tax is less important and age is the most irrelevant.

```r
marginalModelPlots(bm1)
```



*The plots show that the relationship between mileage and target.audi is the strongest.

*The plots also show that the relationship between mpg and target.audi is strong.

*The relationships between tax, age and target.audi are weaker than the relationships between mileage and mpg and target.audi. However, these relationships are still statistically significant.

```
bm2 <- glm(target.audi~ poly(q.mileage,2)+poly(q.tax,2)+poly(q.mpg,2)+poly(q.age,2),family="binomial",da
summary(bm2)
```

```
##
## Call:
## glm(formula = target.audi ~ poly(q.mileage, 2) + poly(q.tax,
##     2) + poly(q.mpg, 2) + poly(q.age, 2), family = "binomial",
##     data = df_work)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.35136    0.04001 -33.776  < 2e-16 ***
## poly(q.mileage, 2)1  17.23336    4.56325   3.777 0.000159 ***
## poly(q.mileage, 2)2  -6.80616    3.13855  -2.169 0.030116 *
## poly(q.tax, 2)1       2.89206    2.38995   1.210 0.226243
## poly(q.tax, 2)2       0.78756    2.39345   0.329 0.742120
## poly(q.mpg, 2)1     -16.96446    2.78238  -6.097 1.08e-09 ***
## poly(q.mpg, 2)2       7.33463    2.29717   3.193 0.001409 **
## poly(q.age, 2)1      -2.71914    4.62959  -0.587 0.556976
## poly(q.age, 2)2      -0.48594    3.07608  -0.158 0.874477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4070.5  on 3951  degrees of freedom
## Residual deviance: 3992.6  on 3943  degrees of freedom
## AIC: 4010.6
##
## Number of Fisher Scoring iterations: 4
```

```r
## Check if transformation is needed
AIC(bm1,bm2)
```

```
##     df      AIC
## bm1  5 4018.370
## bm2  9 4010.635
```

```r
anova(bm1,bm2, test = "LR")
```

```
## Analysis of Deviance Table
##
## Model 1: target.audi ~ q.mileage + q.tax + q.mpg + q.age
## Model 2: target.audi ~ poly(q.mileage, 2) + poly(q.tax, 2) + poly(q.mpg,
##     2) + poly(q.age, 2)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3947     4008.4
## 2      3943     3992.6  4   15.735 0.003397 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with polynomial transformations (bm2) shows slightly better fit compared to the original model (bm1). The AIC of bm2 is slightly lower, and the LR test indicates that the transformation resulted in a significantly better fit (p-value < 0.05).

However, it's important to note that overfitting is a potential concern when using polynomial transformations.

```r
bm3<-glm(target.audi~q.mileage+q.tax+q.mpg+q.age,family="binomial",data=df_work[!df_work$mout=="YesMOut'
summary(bm3)
```

```
##
## Call:
## glm(formula = target.audi ~ q.mileage + q.tax + q.mpg + q.age,
##     family = "binomial", data = df_work[!df_work$mout == "YesMOut",
##         ])
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.324e-01  5.985e-01  -1.558 0.119275
## q.mileage    1.325e-05  3.448e-06   3.842 0.000122 ***
## q.tax        4.363e-03  3.561e-03   1.225 0.220395
## q.mpg       -2.581e-02  3.970e-03  -6.500 8.02e-11 ***
## q.age        5.773e-03  3.435e-02   0.168 0.866524
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4017.2  on 3911  degrees of freedom
## Residual deviance: 3945.4  on 3907  degrees of freedom
## AIC: 3955.4
##
## Number of Fisher Scoring iterations: 4
```

```r
Anova(bm3, test="LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: target.audi
##           LR Chisq Df Pr(>Chisq)
## q.mileage   14.602  1  0.0001328 ***
## q.tax        1.488  1  0.2224663
```

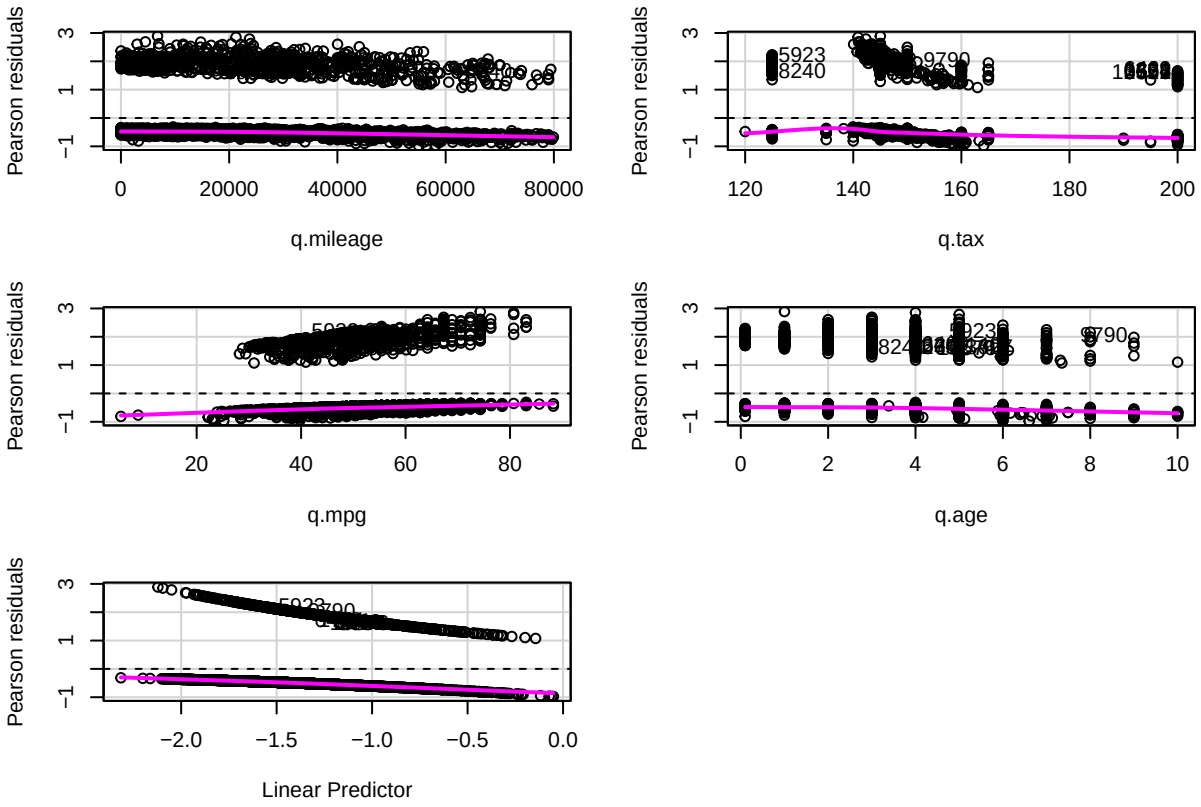
```
## q.mpg       42.843  1   5.93e-11 ***
## q.age        0.028  1  0.8665784
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(bm3)
```

```
## q.mileage     q.tax     q.mpg     q.age
##  2.952322  1.138721  1.268503  2.917841
```

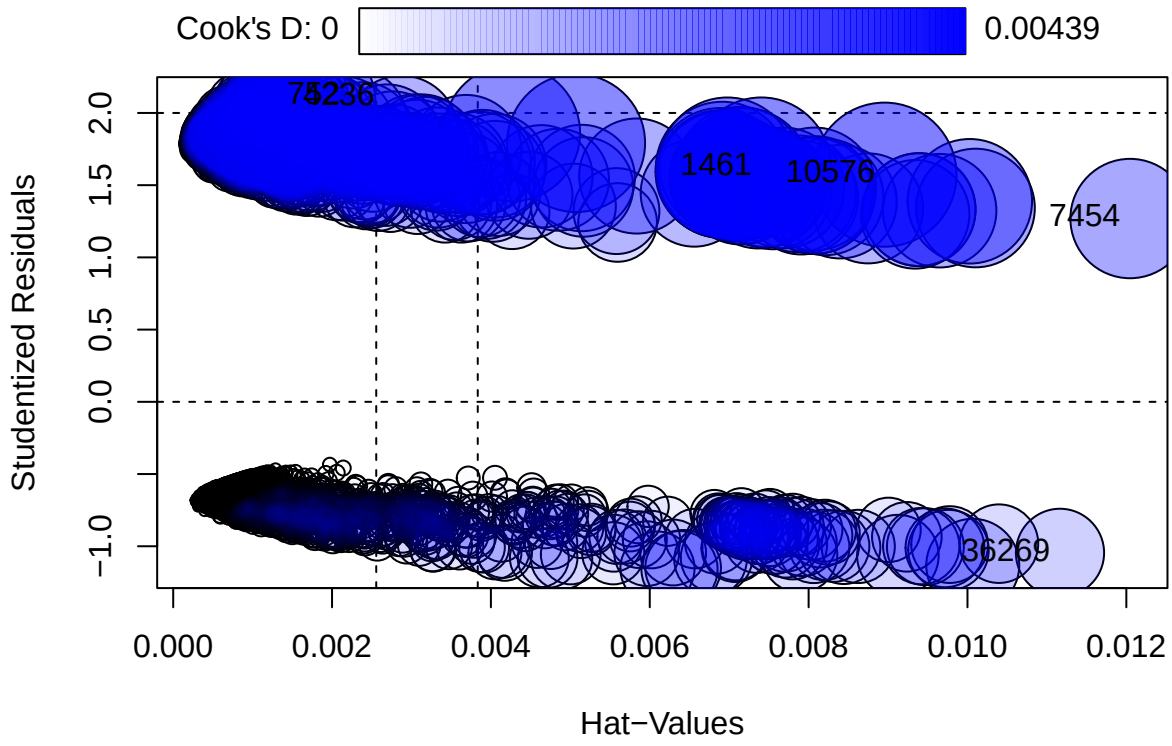
We get slightly better results than bm1.

```
residualPlots(bm3,id=list(method=cooks.distance(bm3),n=10))
```




```
##           Test stat Pr(>|Test stat|)
## q.mileage    4.7436          0.02941 *
## q.tax        0.0256          0.87300
## q.mpg        2.3607          0.12443
## q.age        1.5564          0.21220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
influencePlot( bm3 )
```




```
##          StudRes         Hat       CookD
## 10576  1.572653 0.008955130 0.004388723
## 1461   1.626169 0.007404677 0.004079340
## 4236   2.106023 0.001516756 0.002473538
## 752    2.116776 0.001312694 0.002196761
## 36269 -1.045533 0.011164617 0.001644917
## 7454   1.268884 0.012046810 0.003012021
```

```
outlierTest(bm3)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 752 2.116776           0.034279           NA
```

Overall, the influence plot suggests that the model is not unduly influenced by any particular observation. This is a good thing, as it means that the model is likely to generalize well to new data.

Overall, the outlier test suggests that there is no strong evidence of outliers in the dataset based on the studentized residuals. This is consistent with the findings from the influence plot.

```
bm4 <- step( bm3)
```

```
## Start:  AIC=3955.38
## target.audi ~ q.mileage + q.tax + q.mpg + q.age
##
##             Df Deviance    AIC
## - q.age      1   3945.4 3953.4
## - q.tax      1   3946.9 3954.9
## <none>           3945.4 3955.4
## - q.mileage  1   3960.0 3968.0
```

```
## - q.mpg      1   3988.2 3996.2
##
## Step:  AIC=3953.41
## target.audi ~ q.mileage + q.tax + q.mpg
##
##             Df Deviance    AIC
## - q.tax      1   3947.0 3953.0
## <none>           3945.4 3953.4
## - q.mileage  1   3982.5 3988.5
## - q.mpg      1   3988.4 3994.4
##
## Step:  AIC=3953
## target.audi ~ q.mileage + q.mpg
##
##             Df Deviance    AIC
## <none>           3947.0 3953.0
## - q.mileage  1   3990.2 3994.2
## - q.mpg      1   3999.0 4003.0
```

```
anova( bm4, bm3, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: target.audi ~ q.mileage + q.mpg
## Model 2: target.audi ~ q.mileage + q.tax + q.mpg + q.age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3909     3947.0
## 2      3907     3945.4  2   1.6192    0.445
```

```
waldtest( bm4, bm3, test="Chisq")
```

```
## Wald test
##
## Model 1: target.audi ~ q.mileage + q.mpg
## Model 2: target.audi ~ q.mileage + q.tax + q.mpg + q.age
##   Res.Df Df  Chisq Pr(>Chisq)
## 1   3909
## 2   3907  2 1.6378     0.4409
```

```
x2 <- cbind(df_work$q.tax[!df_work$mout=="YesMOut"],df_work$q.age[!df_work$mout=="YesMOut"])
z<-glm.scoretest(bm4,x2);z # library statmod
```

```
## [1] 1.2694136 0.3619108
```

```
2*(1-pnorm(abs(z)))
```

```
## [1] 0.2042936 0.7174187
```

```
llres <- which(abs(rstudent(bm3))>2.5);length(llres)
```

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

```
df_work[llres,]
```

```
##  [1] f.model          f.year           target.price     f.transmission
##  [5] q.mileage        f.fuel_type      q.tax            q.mpg
##  [9] q.engine_size    f.manufacturer   q.age            mout
## [13] f.aux_price      f.aux_tax        f.used           f.efficiency
## [17] f.old            f.aux_EngineSize target.audi
## <0 rows> (or 0-length row.names)
```

The model with the reduced set of predictor variables (q.mileage, q.mpg) is a good fit for the data and has less predictors than the original mode.

##Adding Factors

```
bm5 <- update(bm4, ~.+f.fuel_type+f.transmission+f.year+f.aux_EngineSize,data=df_work[!df_work$mout=="Ye
vif(bm5)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## q.mileage         3.121698  1        1.766833
## q.mpg             2.500509  1        1.581300
## f.fuel_type       2.631957  2        1.273707
## f.transmission    1.467328  2        1.100606
## f.year            3.808634 18        1.037845
## f.aux_EngineSize 2.664378  2        1.277612
```

```
summary(bm5)
```

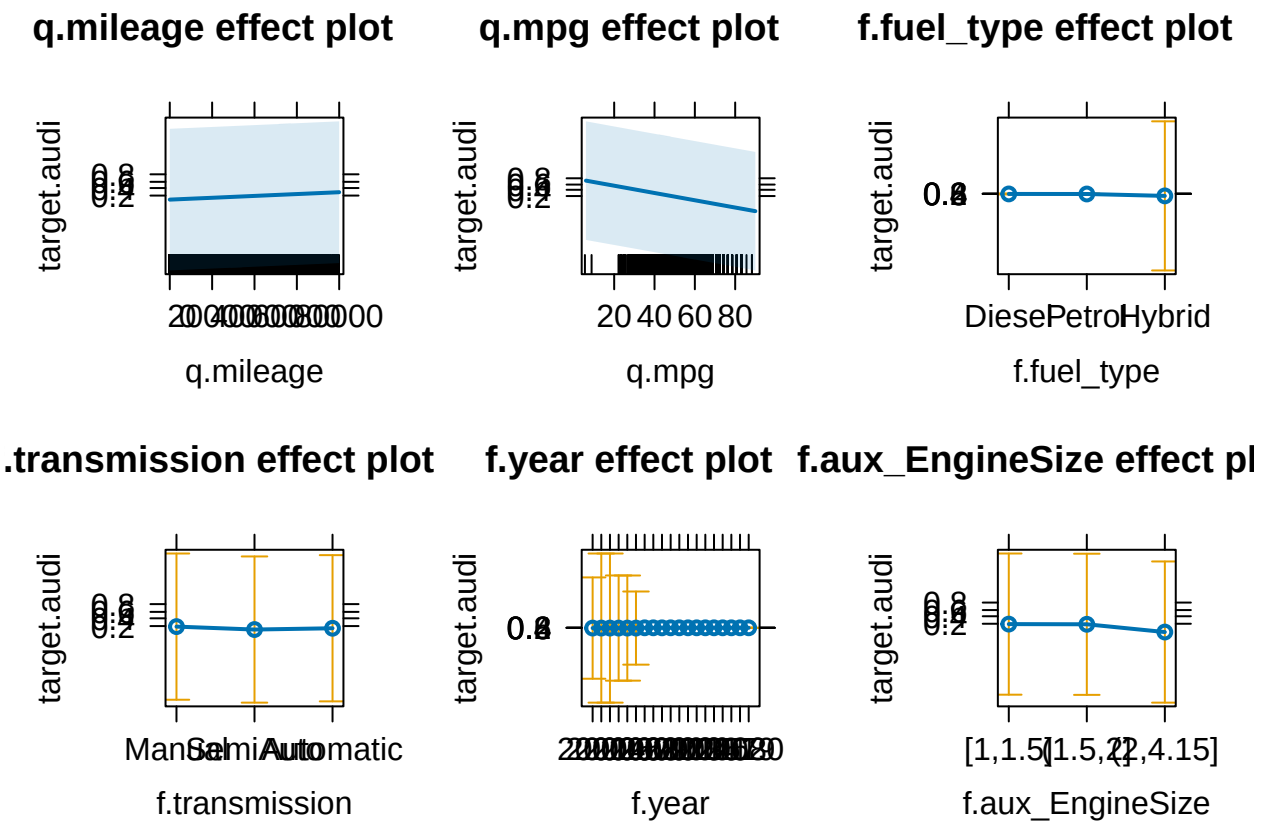
```
##
## Call:
## glm(formula = target.audi ~ q.mileage + q.mpg + f.fuel_type +
##     f.transmission + f.year + f.aux_EngineSize, family = "binomial",
##     data = df_work[!df_work$mout == "YesMOut", ])
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.438e+01  1.630e+03  -0.009 0.992963
## q.mileage                   1.209e-05  3.621e-06   3.340 0.000838 ***
## q.mpg                      -5.674e-02  5.807e-03  -9.771  < 2e-16 ***
## f.fuel_typePetrol          -5.139e-01  1.318e-01  -3.899 9.64e-05 ***
## f.fuel_typeHybrid          -1.567e+01  3.057e+02  -0.051 0.959110
## f.transmissionSemiAuto     -3.699e-01  1.091e-01  -3.390 0.000700 ***
## f.transmissionAutomatic    -2.073e-01  1.151e-01  -1.801 0.071771 .
## f.year2002                 -5.919e-01  2.901e+03   0.000 0.999837
## f.year2004                  2.321e-01  2.901e+03   0.000 0.999936
## f.year2005                 -1.735e-01  2.350e+03   0.000 0.999941
## f.year2006                 -8.666e-02  2.350e+03   0.000 0.999971
## f.year2007                 -1.272e-01  2.011e+03   0.000 0.999950
## f.year2008                  1.624e+01  1.630e+03   0.010 0.992051
## f.year2009                  1.730e+01  1.630e+03   0.011 0.991535
## f.year2010                  1.492e+01  1.630e+03   0.009 0.992697
## f.year2011                  1.592e+01  1.630e+03   0.010 0.992211
## f.year2012                  1.636e+01  1.630e+03   0.010 0.991991
## f.year2013                  1.646e+01  1.630e+03   0.010 0.991945
## f.year2014                  1.638e+01  1.630e+03   0.010 0.991982
## f.year2015                  1.664e+01  1.630e+03   0.010 0.991855
## f.year2016                  1.672e+01  1.630e+03   0.010 0.991816
## f.year2017                  1.645e+01  1.630e+03   0.010 0.991949
## f.year2018                  1.624e+01  1.630e+03   0.010 0.992053
## f.year2019                  1.616e+01  1.630e+03   0.010 0.992094
## f.year2020                  1.630e+01  1.630e+03   0.010 0.992022
## f.aux_EngineSize(1.5,2]    -2.555e-02  1.311e-01  -0.195 0.845465
## f.aux_EngineSize(2,4.15]   -1.041e+00  1.897e-01  -5.488 4.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4017.2  on 3911  degrees of freedom
## Residual deviance: 3800.4  on 3885  degrees of freedom
## AIC: 3854.4
##
## Number of Fisher Scoring iterations: 15
```

```r
Anova(bm5, test="LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: target.audi
##                   LR Chisq Df Pr(>Chisq)
## q.mileage          11.106  1  0.0008604 ***
## q.mpg             100.055  1  < 2.2e-16 ***
## f.fuel_type        43.990  2  2.804e-10 ***
## f.transmission     11.586  2  0.0030483 **
## f.year             38.110 18  0.0037454 **
## f.aux_EngineSize   66.559  2  3.523e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results suggest that the model with the added factors is a better predictor of the target variable than the model without the added factors. The factors that are most important for predicting the target variable are mileage, fuel type, transmission, and aux_EngineSize.

```r
plot(allEffects(bm5))
```




```r
bm6<- step( bm5 )
```

```
## Start:  AIC=3854.41
## target.audi ~ q.mileage + q.mpg + f.fuel_type + f.transmission +
##     f.year + f.aux_EngineSize
##
##                    Df Deviance    AIC
## <none>                  3800.4 3854.4
## - f.year           18   3838.5 3856.5
## - f.transmission    2   3812.0 3862.0
## - q.mileage         1   3811.5 3863.5
## - f.fuel_type       2   3844.4 3894.4
## - f.aux_EngineSize  2   3867.0 3917.0
## - q.mpg             1   3900.5 3952.5
```

```r
vif(bm6)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## q.mileage        3.121698  1        1.766833
## q.mpg            2.500509  1        1.581300
## f.fuel_type      2.631957  2        1.273707
## f.transmission   1.467328  2        1.100606
## f.year           3.808634 18        1.037845
## f.aux_EngineSize 2.664378  2        1.277612
```

```r
summary(bm6)
```

```
##
## Call:
## glm(formula = target.audi ~ q.mileage + q.mpg + f.fuel_type +
##     f.transmission + f.year + f.aux_EngineSize, family = "binomial",
##     data = df_work[!df_work$mout == "YesMOut", ])
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.438e+01  1.630e+03  -0.009 0.992963
## q.mileage                  1.209e-05  3.621e-06   3.340 0.000838 ***
## q.mpg                     -5.674e-02  5.807e-03  -9.771  < 2e-16 ***
## f.fuel_typePetrol         -5.139e-01  1.318e-01  -3.899 9.64e-05 ***
## f.fuel_typeHybrid         -1.567e+01  3.057e+02  -0.051 0.959110
## f.transmissionSemiAuto    -3.699e-01  1.091e-01  -3.390 0.000700 ***
## f.transmissionAutomatic   -2.073e-01  1.151e-01  -1.801 0.071771 .
## f.year2002                -5.919e-01  2.901e+03   0.000 0.999837
## f.year2004                 2.321e-01  2.901e+03   0.000 0.999936
## f.year2005                -1.735e-01  2.350e+03   0.000 0.999941
## f.year2006                -8.666e-02  2.350e+03   0.000 0.999971
## f.year2007                -1.272e-01  2.011e+03   0.000 0.999950
## f.year2008                 1.624e+01  1.630e+03   0.010 0.992051
## f.year2009                 1.730e+01  1.630e+03   0.011 0.991535
## f.year2010                 1.492e+01  1.630e+03   0.009 0.992697
## f.year2011                 1.592e+01  1.630e+03   0.010 0.992211
## f.year2012                 1.636e+01  1.630e+03   0.010 0.991991
## f.year2013                 1.646e+01  1.630e+03   0.010 0.991945
## f.year2014                 1.638e+01  1.630e+03   0.010 0.991982
## f.year2015                 1.664e+01  1.630e+03   0.010 0.991855
## f.year2016                 1.672e+01  1.630e+03   0.010 0.991816
## f.year2017                 1.645e+01  1.630e+03   0.010 0.991949
## f.year2018                 1.624e+01  1.630e+03   0.010 0.992053
## f.year2019                 1.616e+01  1.630e+03   0.010 0.992094
## f.year2020                 1.630e+01  1.630e+03   0.010 0.992022
## f.aux_EngineSize(1.5,2]   -2.555e-02  1.311e-01  -0.195 0.845465
## f.aux_EngineSize(2,4.15] -1.041e+00  1.897e-01  -5.488 4.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4017.2  on 3911  degrees of freedom
## Residual deviance: 3800.4  on 3885  degrees of freedom
## AIC: 3854.4
##
## Number of Fisher Scoring iterations: 15
```

```r
AIC(bm1,bm2,bm3,bm4,bm5,bm6)
```

```
## Warning in AIC.default(bm1, bm2, bm3, bm4, bm5, bm6): models are not all fitted
## to the same number of observations
```

```
##     df      AIC
## bm1  5 4018.370
## bm2  9 4010.635
## bm3  5 3955.381
## bm4  3 3953.000
## bm5 27 3854.414
## bm6 27 3854.414
```

We see that the best fit is bm5. ## Interactions

```
#bm7 <- update(bm6, ~.*(f.used),data=df_work[!df_work$mout=="YesMOut",])
#vif(bm7)
#summary(bm7)
#Anova(bm7, test="LR")

#bm8<- step( bm7 )
#vif(bm8)
#summary(bm8)
#Anova(bm8, test="LR")
#anova( bm8, bm7, test="Chisq")

#residualPlots(bm8,id=list(method=cooks.distance(bm8),n=10))
```

2.1 Diagnostics

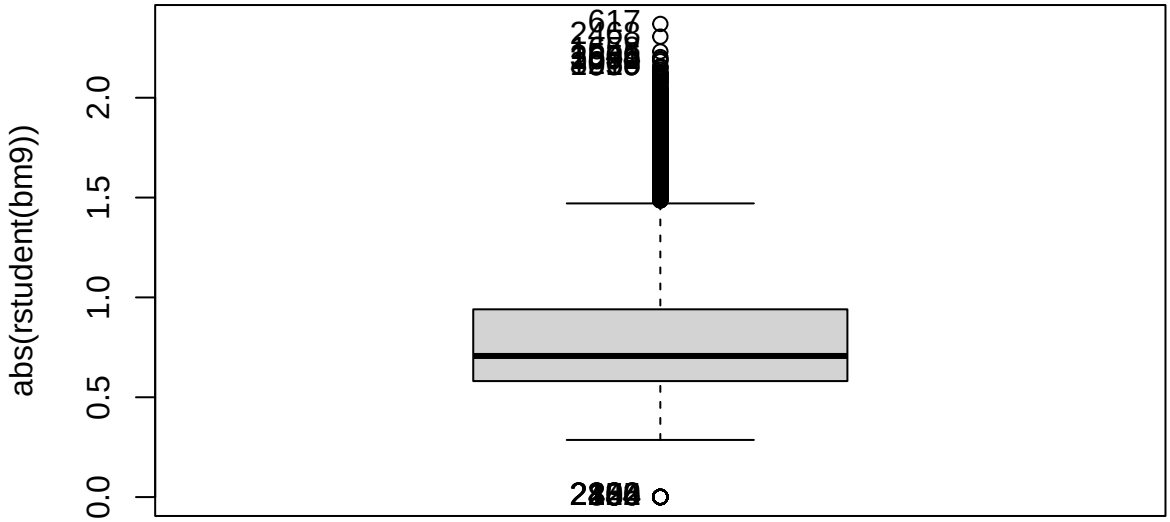
```
dfwork <- df_work[!df_work$mout=="YesMOut",]
bm9<-glm(target.audi ~ q.mileage + q.mpg + f.transmission + f.fuel_type +
    + f.aux_EngineSize + q.mileage:f.transmission,family="binomial",data=dfwork)

vif(bm9)
```

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

```
##                              GVIF Df GVIF^(1/(2*Df))
## q.mileage                2.843087  1        1.686146
## q.mpg                    2.037894  1        1.427548
## f.transmission           6.763298  2        1.612648
## f.fuel_type              2.413629  2        1.246429
## f.aux_EngineSize         2.497450  2        1.257113
## q.mileage:f.transmission 7.905360  2        1.676797
```

```
Boxplot(abs(rstudent(bm9)))
```




```
##  [1] 2444  836 2104  154  104  232  155 2120 2264 2852  617 2468 1658 1546 3655
## [16] 1354 2691 3268 1019  950
```

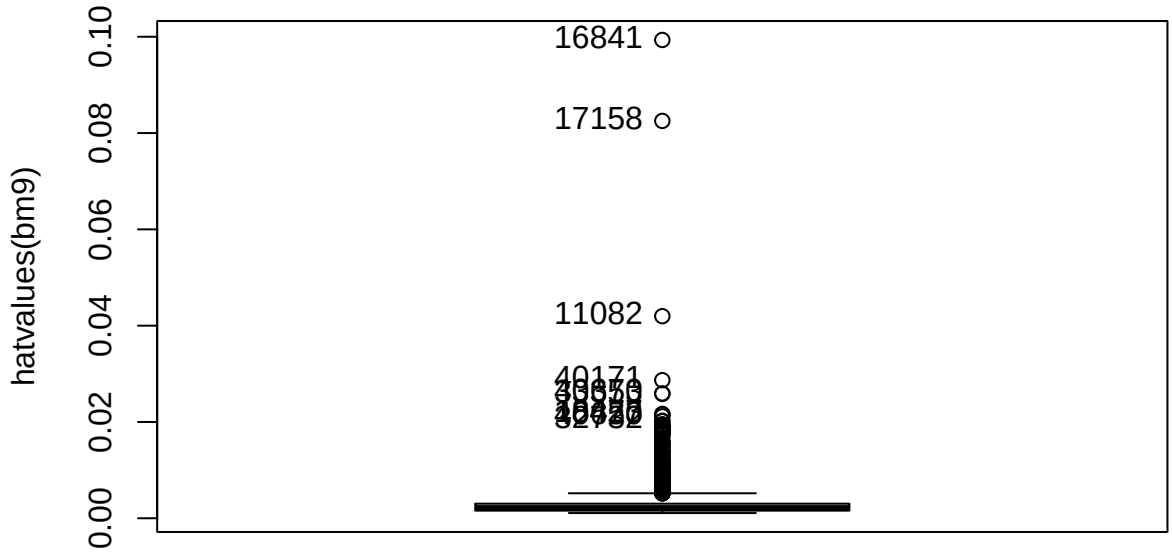
```
llres <- which(abs(rstudent(bm9))>2.3);llres
```

```
##  365 5259
##  617 2468
```

```
which(row.names(dfwork) %in% names(rstudent(bm9)[llres]))
```

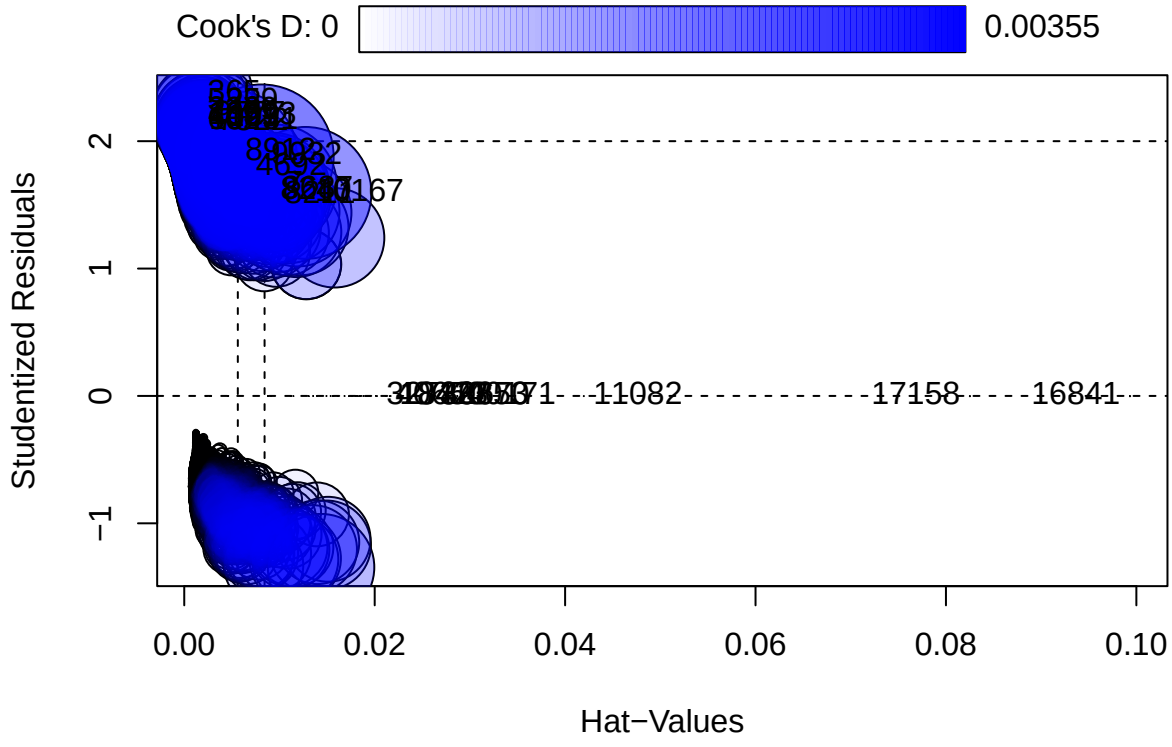
```
## [1]  617 2468
```

```
Boxplot(hatvalues(bm9),id=list(labels=row.names(dfwork)))
```




```
##  [1] "16841" "17158" "11082" "40171" "40670" "33353" "18427" "18473" "40650"
## [10] "32732"
```

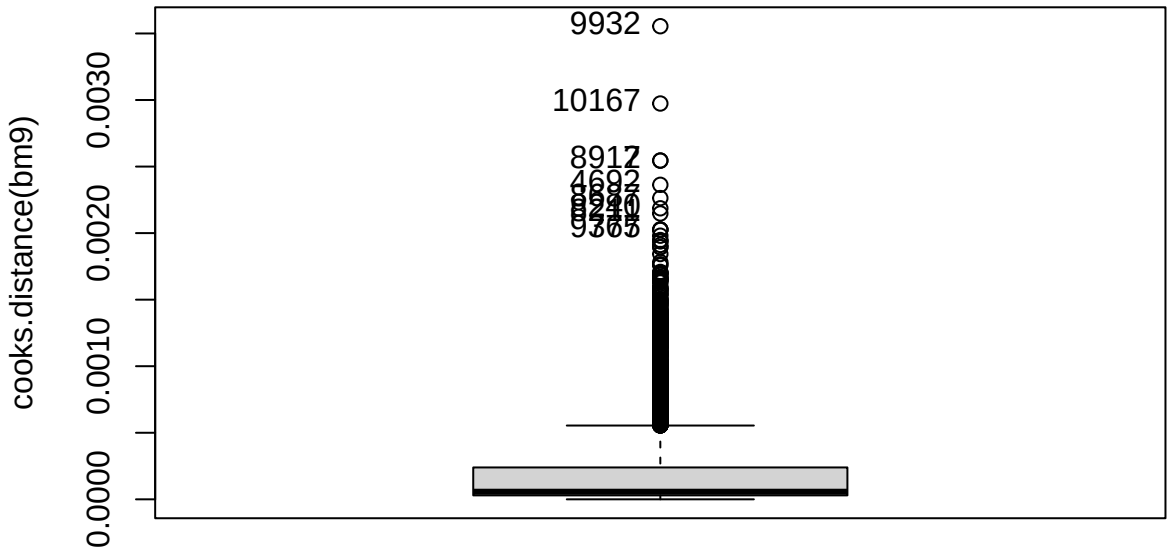
```
influencePlot(bm9, id=list(n=10))
```




```
##              StudRes         Hat        CookD
```

```
## 11082 -0.0009405816 0.041970568 1.799480e-09
## 32732 -0.0006495541 0.020245551 4.003492e-10
## 365    2.3694348204 0.001445532 2.026796e-03
## 18427 -0.0006723306 0.021674253 4.601888e-10
## 7522   2.1512848776 0.002460612 2.023901e-03
## 6027   2.1521130976 0.001415048 1.169834e-03
## 40650 -0.0006644766 0.021176268 4.388383e-10
## 8687   1.6142213959 0.009275087 2.263330e-03
## 10167  1.5939364025 0.012728151 2.973679e-03
## 1389   2.1912147185 0.001514482 1.373815e-03
## 18473 -0.0006710550 0.021592985 4.566685e-10
## 4692   1.7956497194 0.006495788 2.362826e-03
## 7      1.6514912778 0.009614023 2.545587e-03
## 9777   2.2035957883 0.002174560 2.027162e-03
## 40670 -0.0007362384 0.025933218 6.645844e-10
## 16841 -0.0014703746 0.099335579 1.140512e-08
## 2269   2.2313739536 0.001404861 1.404055e-03
## 8240   1.6069201676 0.009104653 2.186421e-03
## 8211   1.5779476304 0.009528043 2.147641e-03
## 40171 -0.0007748116 0.028680706 8.174675e-10
## 5259   2.3058563240 0.001425987 1.708451e-03
## 33353 -0.0007347116 0.025827195 6.590180e-10
## 4199   2.1826248557 0.001607112 1.427784e-03
## 8912   1.9132170266 0.005377012 2.544194e-03
## 10431  2.1578902959 0.001610248 1.348783e-03
## 9932   1.8795138634 0.008120644 3.554366e-03
## 10513  2.1937703084 0.001455544 1.328721e-03
## 17158 -0.0013336811 0.082520379 7.584824e-09
```

```
Boxplot(cooks.distance(bm9),id=list(labels=row.names(dfwork)))
```




```
##  [1] "9932"  "10167" "7"     "8912"  "4692"  "8687"  "8240"  "8211"  "9777"
## [10] "365"
```

```r
llout<-which(abs(cooks.distance(bm9))>0.02);length(llout)
```

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

```r
which(row.names(dfwork) %in% names(cooks.distance(bm9)[llout]))
```

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

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

```
## [1]  617 2468
```

```r
bm10<-glm(target.audi ~ q.mileage + q.mpg + f.transmission + f.fuel_type +
    + f.aux_EngineSize + q.mileage:f.transmission, family="binomial",data=dfwork[-llrem,])
summary(bm10)
```

```
##
## Call:
## glm(formula = target.audi ~ q.mileage + q.mpg + f.transmission +
##     f.fuel_type + +f.aux_EngineSize + q.mileage:f.transmission,
##     family = "binomial", data = dfwork[-llrem, ])
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      1.341e+00  3.667e-01   3.658 0.000254 ***
## q.mileage                        1.668e-05  3.413e-06   4.886 1.03e-06 ***
## q.mpg                           -4.807e-02  5.210e-03  -9.226  < 2e-16 ***
## f.transmissionSemiAuto          -5.589e-01  1.596e-01  -3.502 0.000461 ***
## f.transmissionAutomatic          4.910e-03  1.696e-01   0.029 0.976899
## f.fuel_typePetrol               -4.065e-01  1.260e-01  -3.225 0.001259 **
## f.fuel_typeHybrid               -1.442e+01  1.887e+02  -0.076 0.939061
## f.aux_EngineSize(1.5,2]          3.422e-02  1.306e-01   0.262 0.793262
## f.aux_EngineSize(2,4.15]        -8.989e-01  1.826e-01  -4.923 8.52e-07 ***
## q.mileage:f.transmissionSemiAuto  1.037e-05  5.118e-06   2.027 0.042705 *
## q.mileage:f.transmissionAutomatic -9.746e-06  4.971e-06  -1.961 0.049916 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4010.9  on 3909  degrees of freedom
## Residual deviance: 3814.3  on 3899  degrees of freedom
## AIC: 3836.3
##
## Number of Fisher Scoring iterations: 14
```

```r
bm0<-glm(target.audi ~ 1, family="binomial", data=dfwork[-llrem,])

vif(bm10)
```

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

```
##                              GVIF Df GVIF^(1/(2*Df))
## q.mileage                2.839449  1        1.685067
## q.mpg                    2.039953  1        1.428269
## f.transmission           6.764419  2        1.612715
## f.fuel_type              2.413889  2        1.246463
## f.aux_EngineSize         2.499660  2        1.257391
## q.mileage:f.transmission 7.914056  2        1.677258
```

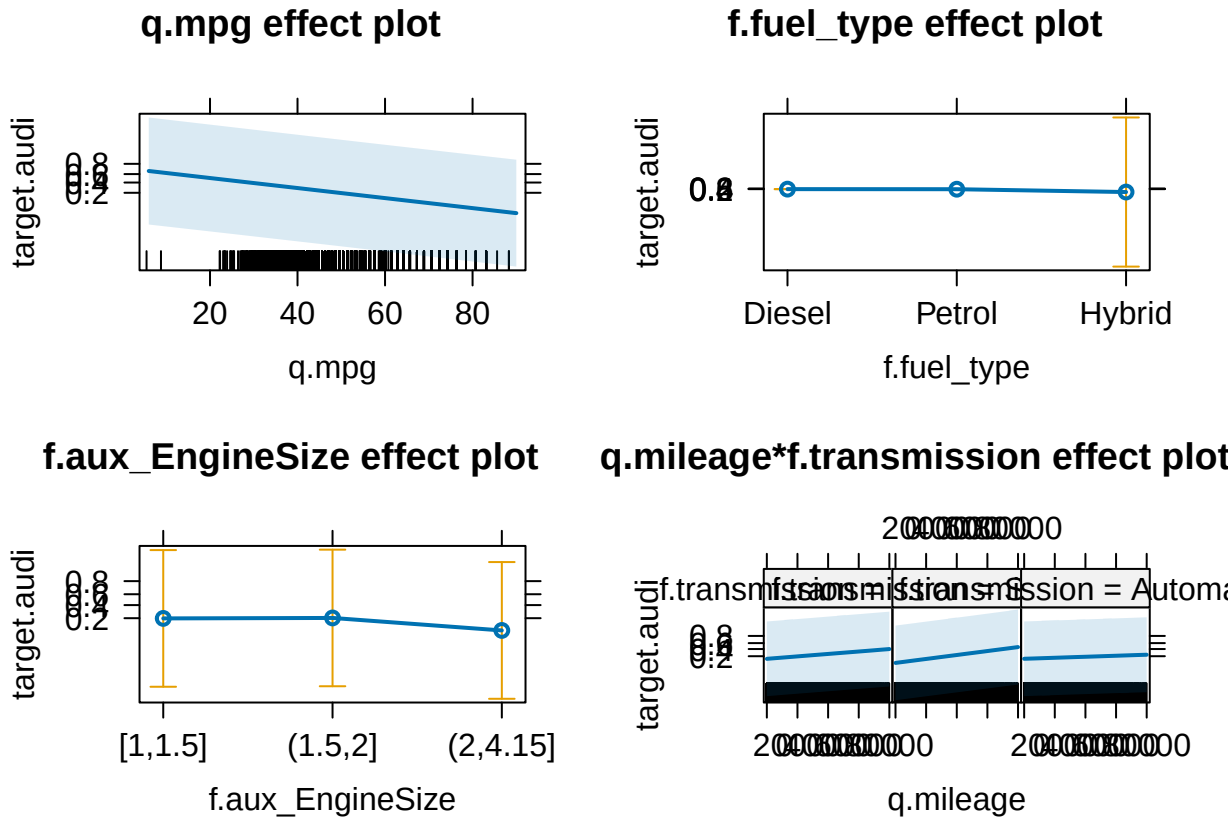
```r
summary(bm10)
```

```
##
## Call:
## glm(formula = target.audi ~ q.mileage + q.mpg + f.transmission +
##     f.fuel_type + +f.aux_EngineSize + q.mileage:f.transmission,
##     family = "binomial", data = dfwork[-llrem, ])
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      1.341e+00  3.667e-01   3.658 0.000254 ***
## q.mileage                        1.668e-05  3.413e-06   4.886 1.03e-06 ***
## q.mpg                           -4.807e-02  5.210e-03  -9.226  < 2e-16 ***
## f.transmissionSemiAuto          -5.589e-01  1.596e-01  -3.502 0.000461 ***
## f.transmissionAutomatic          4.910e-03  1.696e-01   0.029 0.976899
## f.fuel_typePetrol               -4.065e-01  1.260e-01  -3.225 0.001259 **
## f.fuel_typeHybrid               -1.442e+01  1.887e+02  -0.076 0.939061
## f.aux_EngineSize(1.5,2]          3.422e-02  1.306e-01   0.262 0.793262
## f.aux_EngineSize(2,4.15]        -8.989e-01  1.826e-01  -4.923 8.52e-07 ***
## q.mileage:f.transmissionSemiAuto  1.037e-05  5.118e-06   2.027 0.042705 *
## q.mileage:f.transmissionAutomatic -9.746e-06  4.971e-06  -1.961 0.049916 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4010.9  on 3909  degrees of freedom
## Residual deviance: 3814.3  on 3899  degrees of freedom
## AIC: 3836.3
##
## Number of Fisher Scoring iterations: 14
```

```r
Anova(bm10)
```

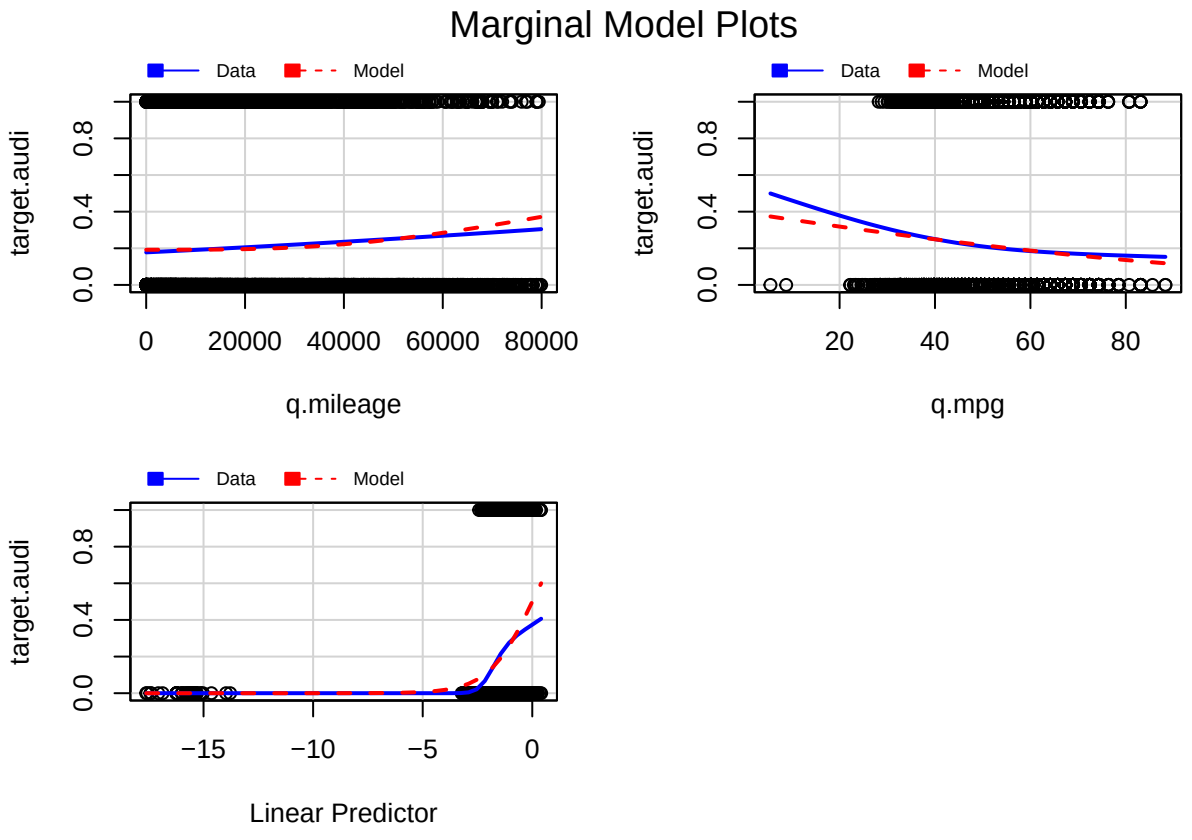
```
## Analysis of Deviance Table (Type II tests)
##
## Response: target.audi
##                           LR Chisq Df Pr(>Chisq)
## q.mileage                   49.760  1  1.737e-12 ***
## q.mpg                       88.074  1  < 2.2e-16 ***
## f.transmission              11.971  2  0.0025156 **
## f.fuel_type                 35.459  2  1.996e-08 ***
## f.aux_EngineSize            59.458  2  1.227e-13 ***
## q.mileage:f.transmission    13.891  2  0.0009631 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(allEffects(bm10))
```

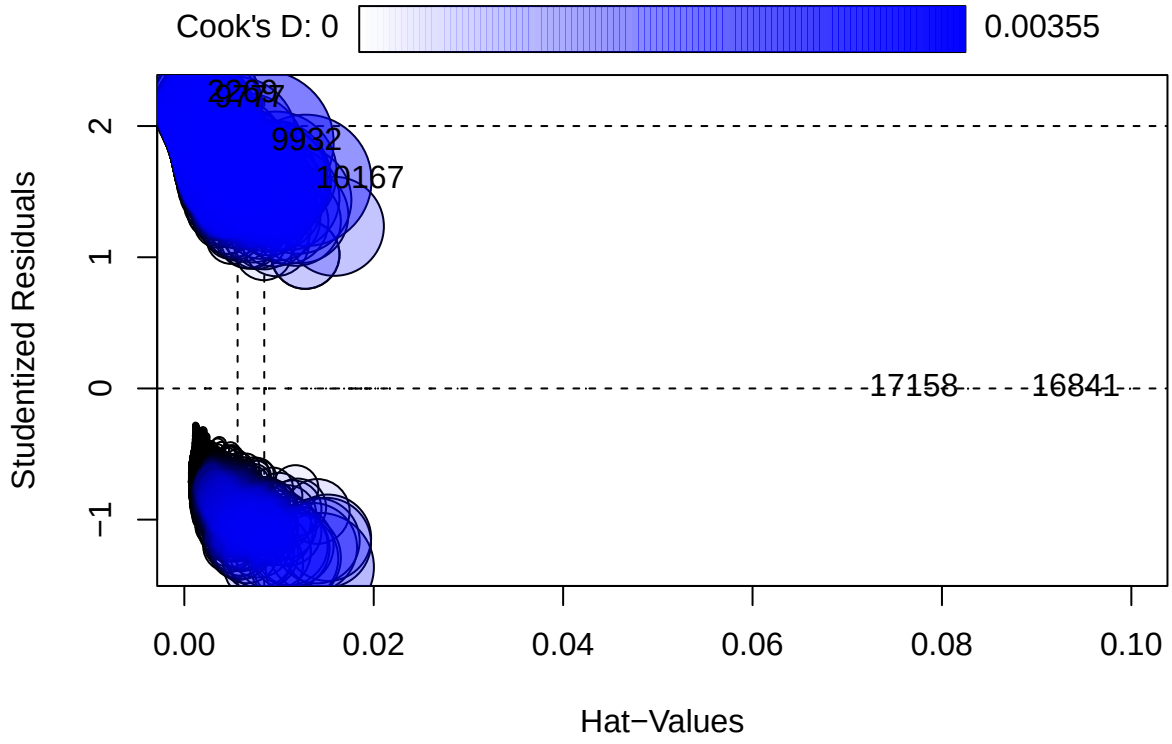



```
marginalModelPlots(bm10)
```

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




```
influencePlot(bm10)
```




```
##             StudRes         Hat        CookD
## 10167  1.586163993 0.012835201 2.948371e-03
## 9777   2.205248623 0.002173715 2.034404e-03
## 16841 -0.001475984 0.099863097 1.156332e-08
## 2269   2.244625759 0.001388570 1.433062e-03
## 9932   1.876780258 0.008159705 3.549705e-03
## 17158 -0.001337082 0.082763676 7.649029e-09
```

```
outlierTest(bm10)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 2269 2.244626           0.024792           NA
```

3 Goodness of fit and Predictive Capacity

```
# H0: Model fits data
Anova(bm10)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: target.audi
##                            LR Chisq Df Pr(>Chisq)
## q.mileage                    49.760  1  1.737e-12 ***
## q.mpg                        88.074  1  < 2.2e-16 ***
## f.transmission               11.971  2  0.0025156 **
## f.fuel_type                  35.459  2  1.996e-08 ***
## f.aux_EngineSize             59.458  2  1.227e-13 ***
## q.mileage:f.transmission     13.891  2  0.0009631 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
1-pchisq(bm10$deviance, bm10$df.residual)
```

```
## [1] 0.8311169
```

```r
X2bm10<-sum((resid(bm10,"pearson")^2));X2bm10
```

```
## [1] 3763.507
```

```r
1-pchisq( X2bm10, bm10$df.res)
```

```
## [1] 0.9388173
```

```r
# PseudoR2
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.3.2
```

```
##
## Attaching package: 'DescTools'
```

```
## The following object is masked from 'package:car':
##
##     Recode
```

```r
PseudoR2(bm10, which='all') # Not working for grouped data
```

```
##        McFadden     McFaddenAdj        CoxSnell      Nagelkerke   AldrichNelson
##    4.901825e-02    4.353321e-02    4.904006e-02    7.644670e-02    4.787598e-02
## VeallZimmermann           Efron McKelveyZavoina            Tjur             AIC
##    9.454743e-02    4.196654e-02    4.842704e-01    4.525981e-02    3.836304e+03
##             BIC          logLik         logLik0              G2
##    3.905288e+03   -1.907152e+03   -2.005456e+03    1.966079e+02
```

```r
# Sheather
1 - (bm10$deviance / bm10$null.deviance)
```

```
## [1] 0.04901825
```

```r
# McFadden
1-(as.numeric(logLik(bm10))/as.numeric(logLik(bm0)))
```

```
## [1] 0.04901825
```

```r
library(ResourceSelection)
```

```
## Warning: package 'ResourceSelection' was built under R version 4.3.2
```

```
## ResourceSelection 0.3-6   2023-06-27
```

```r
pred_test <- predict(bm10, newdata=df_test, type="response")
ht <- hoslem.test(as.numeric(df_test$target.audi)-1, pred_test)
ht
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  as.numeric(df_test$target.audi) - 1, pred_test
## X-squared = 15.64, df = 8, p-value = 0.04783
```

```r
cbind(ht$observed, ht$expected)
```

```
##                  y0 y1    yhat0     yhat1
## [4.79e-09,0.105] 92  7 92.41853  6.581467
## (0.105,0.135]    86 13 87.07377 11.926232
## (0.135,0.16]     85 14 84.33959 14.660409
## (0.16,0.177]     77 21 81.49454 16.505462
## (0.177,0.203]    78 21 80.22236 18.777643
## (0.203,0.229]    83 16 77.64872 21.351280
## (0.229,0.254]    77 21 74.36937 23.630631
## (0.254,0.282]    65 34 72.55002 26.449981
## (0.282,0.329]    73 26 68.89608 30.103917
## (0.329,0.557]    74 25 60.42260 38.577397
```

```r
# ROC Curve

library("ROCR")
```

```
## Warning: package 'ROCR' was built under R version 4.3.2
```

```r
library("cvAUC")
```

```
## Warning: package 'cvAUC' was built under R version 4.3.2
```

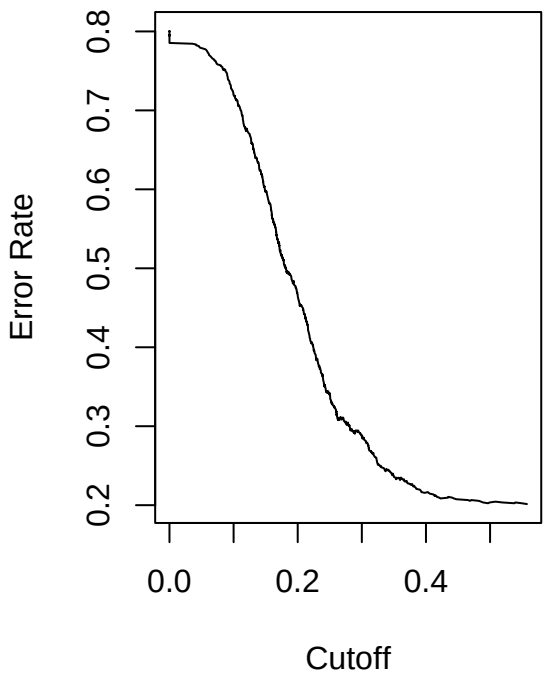
```
##
## Attaching package: 'cvAUC'
```

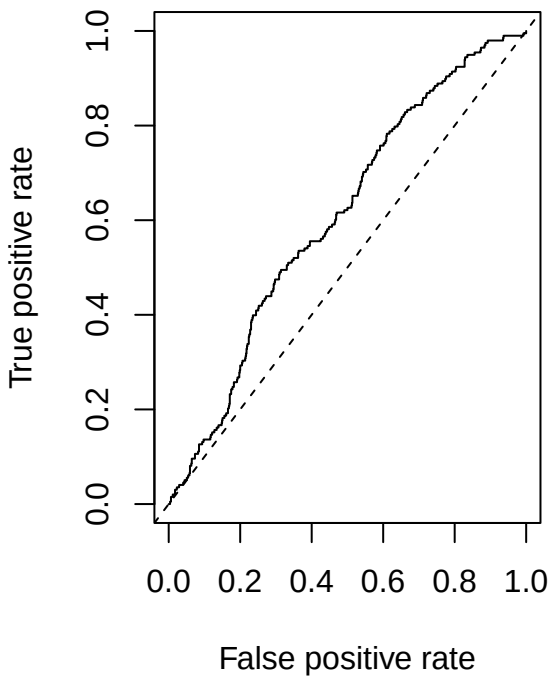
```
## The following object is masked from 'package:DescTools':
##
##     AUC
```

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

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

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






```
AUC(predict(bm10,type="response"),dfwork$target.audi[-llrem])
```

```
## [1] 0.6559984
```

4 Confusion Table

```
audi.est <- ifelse(pred_test<0.4,0,1)
tt<-table(audi.est,df_test$target.audi);tt
```

```
##
## audi.est  No Yes
##        0 767 191
##        1  23   7
```

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

```
## [1] 78.34008
```

```
prob.audi <- bm0$fit
audi.est <- ifelse(prob.audi<0.5,0,1)
tt<-table(audi.est,dfwork$target.audi[-llrem]);tt
```

```
##
## audi.est   No  Yes
##        0 3092  818
```

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

```
## [1] 79.07928
```