STAT 420: Final Project Report

Contents

```
#install.packages("mltools")
#install.packages("caret")
#install.packages("dplyr")
#install.packages("ggplots2")
library(mltools)
library(data.table)
```

Loading dataset in R from the CSV file, and removing some columns which are not needed

```
car_data = read.csv("Cars_data.csv")
car_data = subset(car_data, select = -c(Vehicle.Style, Market.Category))
car_data = na.omit(car_data)

#unique(car_data$Transmission.Type)
#colnames(car_data)
#unique(car_data$Engine.Fuel.Type)
#unique(car_data$Driven_Wheels)
#unique(car_data$Vehicle.Size)
car_data$Make<-NULL
car_data$Model<- NULL
#head(car_data)
```

Removing extreme prices less than $3000 and greater than $100,000

```
car_data_priced<-car_data[!(car_data$MSRP>100000 | car_data$MSRP< 10000 ),]

range(car_data_priced$MSRP)
```

```
## [1] 10135 99950
```

Removing the non automatic/manual transmission types, and storing this new data in car_data_transd dataframe

```
car_data_transd<-car_data_priced[!(car_data_priced$Transmission.Type=="AUTOMATED_MANUAL" | car_data_pri

unique(car_data_transd$Transmission.Type)
```

```
## [1] "MANUAL"    "AUTOMATIC"
```

Removing certain fuel types, keeping only gasoline and diesel. Storing the result in car_data_fuel dataframe

```
car_data_fuel<-car_data_transd[!(grepl("flex", car_data_transd$Engine.Fuel.Type, fixed = TRUE)
|car_data_transd$Engine.Fuel.Type=="electric" | car_data_transd$Engine.Fuel.Type=="" | car_data_transd$
```

```
unique(car_data_fuel$Engine.Fuel.Type)
```

```
## [1] "premium unleaded (required)"    "premium unleaded (recommended)"
## [3] "regular unleaded"               "diesel"
```

```
range(car_data_fuel$MSRP)
```

```
## [1] 10135 99950
```

Assigning the different types of gasoline to a single "gasoline value". Now, the only two values for fuel type will be "gasoline" and "diesel" as visible below

```
car_data_fuel$Engine.Fuel.Type[car_data_fuel$Engine.Fuel.Type == "premium unleaded (required)" ] <- "gas
car_data_fuel$Engine.Fuel.Type[car_data_fuel$Engine.Fuel.Type == "regular unleaded" ] <- "gasoline"
car_data_fuel$Engine.Fuel.Type[car_data_fuel$Engine.Fuel.Type == "premium unleaded (recommended)" ] <-

unique(car_data_fuel$Engine.Fuel.Type)
```

```
## [1] "gasoline" "diesel"
```

Making categorical variables factors, and adding age variable

```
car_data_factored = car_data_fuel
car_data_factored$Vehicle.Size <- factor(car_data_factored$Vehicle.Size)
car_data_factored$Transmission.Type <- factor(car_data_factored$Transmission.Type)
car_data_factored$Engine.Fuel.Type <- factor(car_data_factored$Engine.Fuel.Type)
car_data_factored$Driven_Wheels <- factor(car_data_factored$Driven_Wheels)
car_data_factored$Engine.Cylinders <- factor(car_data_factored$Engine.Cylinders)
car_data_factored$Number.of.Doors <- factor(car_data_factored$Number.of.Doors)

levels(car_data_factored$Vehicle.Size)
```

```
## [1] "Compact" "Large"   "Midsize"
```

```
levels(car_data_factored$Transmission.Type)
```

```
## [1] "AUTOMATIC" "MANUAL"
```

```
levels(car_data_factored$Engine.Fuel.Type)
```

```
## [1] "diesel"   "gasoline"
```

```
levels(car_data_factored$Driven_Wheels)
```

```
## [1] "all wheel drive"   "four wheel drive"  "front wheel drive"
## [4] "rear wheel drive"
```

```r
levels(car_data_factored$Engine.Cylinders)
```

```
## [1] "3"  "4"  "5"  "6"  "8"  "10" "12"
```

```r
levels(car_data_factored$Number.of.Doors)
```

```
## [1] "2" "3" "4"
```

```r
#car_data_factored = one_hot(as.data.table(car_data_factored))

car_data_factored$ReleasedYearsAgo <- with(car_data_factored, 2020 - Year)
```

Removing repetitive/unnecessary variable(s)

```r
car_data_factored$Year <- NULL
```

Modeling

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```r
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```r
set.seed(100)
#train-test  split using 65% of the data
samplesize = round(0.65*nrow(car_data_factored), 0)
index = sample(seq_len(nrow(car_data_factored)), size = samplesize)
data_train = car_data_factored[index,]
data_test = car_data_factored[-index,]
msrp_mod = lm(MSRP ~., data_train)
summary(msrp_mod)
```

```
##
## Call:
## lm(formula = MSRP ~ ., data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43498  -4570   -473   3241  52654
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.099e+04  3.224e+03   3.410 0.000655 ***
## Engine.Fuel.Typegasoline       -1.582e+04  1.161e+03 -13.629  < 2e-16 ***
## Engine.HP                       1.532e+02  2.939e+00  52.125  < 2e-16 ***
## Engine.Cylinders4              -1.059e+03  2.344e+03  -0.452 0.651413
## Engine.Cylinders5              -1.527e+03  2.516e+03  -0.607 0.543870
## Engine.Cylinders6              -2.161e+03  2.428e+03  -0.890 0.373380
## Engine.Cylinders8              -2.857e+03  2.532e+03  -1.128 0.259169
## Engine.Cylinders10             -6.984e+02  4.476e+03  -0.156 0.876016
## Engine.Cylinders12              2.478e+04  5.620e+03   4.410 1.05e-05 ***
## Transmission.TypeMANUAL        -3.694e+03  3.396e+02 -10.880  < 2e-16 ***
## Driven_Wheelsfour wheel drive  -2.611e+03  5.040e+02  -5.181 2.28e-07 ***
## Driven_Wheelsfront wheel drive -5.686e+03  3.597e+02 -15.806  < 2e-16 ***
## Driven_Wheelsrear wheel drive  -3.902e+03  3.951e+02  -9.878  < 2e-16 ***
## Number.of.Doors3               -2.475e+03  1.169e+03  -2.118 0.034255 *
## Number.of.Doors4               -2.058e+03  3.587e+02  -5.737 1.02e-08 ***
## Vehicle.SizeLarge               1.846e+03  4.460e+02   4.139 3.54e-05 ***
## Vehicle.SizeMidsize            -4.389e+02  3.177e+02  -1.382 0.167169
## highway.MPG                     1.841e+02  5.808e+01   3.169 0.001537 **
## city.mpg                        2.659e+02  5.266e+01   5.049 4.59e-07 ***
## Popularity                     -3.900e-01  9.122e-02  -4.276 1.94e-05 ***
## ReleasedYearsAgo                2.535e+01  3.694e+01   0.686 0.492498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8597 on 5283 degrees of freedom
## Multiple R-squared:  0.7042, Adjusted R-squared:  0.7031
## F-statistic: 628.8 on 20 and 5283 DF,  p-value: < 2.2e-16
```

```
msrp_mod2 = lm(MSRP ~ highway.MPG + Popularity, data_test)
#summary(msrp_mod2)
#anova(msrp_mod2, msrp_mod)
```

```
alias(msrp_mod)
```

```
## Model :
## MSRP ~ Engine.Fuel.Type + Engine.HP + Engine.Cylinders + Transmission.Type +
##     Driven_Wheels + Number.of.Doors + Vehicle.Size + highway.MPG +
##     city.mpg + Popularity + ReleasedYearsAgo
```

Trying Polynomial Model with AIC choice

```r
MSRP_big_mod = lm(
  MSRP ~ . + I(Engine.HP ^ 2) + I(ReleasedYearsAgo ^ 2) + I(city.mpg ^ 2) + I(highway.MPG ^ 2) +  + I(P
  data = data_train)

MSRP_mod_back_aic = step(MSRP_big_mod, direction = "backward", trace = 0)

summary(MSRP_mod_back_aic)
```

```
##
## Call:
## lm(formula = MSRP ~ Engine.Fuel.Type + Engine.HP + Engine.Cylinders +
##     Transmission.Type + Driven_Wheels + Number.of.Doors + Vehicle.Size +
##     highway.MPG + city.mpg + Popularity + ReleasedYearsAgo +
##     I(Engine.HP^2) + I(ReleasedYearsAgo^2) + I(highway.MPG^2) +
##     I(Popularity^2), data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32186  -4489   -423   3313  51901
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -9.916e+03  4.114e+03  -2.410 0.015988 *
## Engine.Fuel.Typegasoline        -1.705e+04  1.155e+03 -14.756  < 2e-16 ***
## Engine.HP                        1.961e+02  7.863e+00  24.935  < 2e-16 ***
## Engine.Cylinders4               -3.950e+03  2.359e+03  -1.674 0.094093 .
## Engine.Cylinders5               -4.073e+03  2.532e+03  -1.608 0.107794
## Engine.Cylinders6               -4.696e+03  2.449e+03  -1.918 0.055219 .
## Engine.Cylinders8               -3.218e+03  2.538e+03  -1.268 0.204898
## Engine.Cylinders10               7.537e+03  4.503e+03   1.674 0.094209 .
## Engine.Cylinders12               2.684e+04  5.583e+03   4.808 1.57e-06 ***
## Transmission.TypeMANUAL         -3.302e+03  3.375e+02  -9.783  < 2e-16 ***
## Driven_Wheelsfour wheel drive   -1.397e+03  5.247e+02  -2.662 0.007799 **
## Driven_Wheelsfront wheel drive -5.405e+03  3.590e+02 -15.057  < 2e-16 ***
## Driven_Wheelsrear wheel drive   -3.577e+03  3.919e+02  -9.127  < 2e-16 ***
## Number.of.Doors3                -6.904e+02  1.185e+03  -0.583 0.560052
## Number.of.Doors4                -1.679e+03  3.569e+02  -4.706 2.59e-06 ***
## Vehicle.SizeLarge                1.084e+03  4.517e+02   2.399 0.016478 *
## Vehicle.SizeMidsize             -9.180e+02  3.238e+02  -2.835 0.004603 **
## highway.MPG                      1.424e+03  1.849e+02   7.702 1.59e-14 ***
## city.mpg                         4.813e+02  5.754e+01   8.365  < 2e-16 ***
## Popularity                      -1.564e+00  2.789e-01  -5.607 2.17e-08 ***
## ReleasedYearsAgo                -4.486e+02  1.307e+02  -3.431 0.000606 ***
## I(Engine.HP^2)                  -7.210e-02  1.129e-02  -6.388 1.82e-10 ***
## I(ReleasedYearsAgo^2)            2.606e+01  6.303e+00   4.134 3.62e-05 ***
## I(highway.MPG^2)                -2.214e+01  3.174e+00  -6.976 3.41e-12 ***
## I(Popularity^2)                  2.258e-04  4.900e-05   4.607 4.18e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8492 on 5279 degrees of freedom
## Multiple R-squared:  0.7116, Adjusted R-squared:  0.7103
## F-statistic: 542.8 on 24 and 5279 DF,  p-value: < 2.2e-16
```

Assumptions

```
plot_func = function(model, pointcol = "blue",linecol = "green") {
  plot(fitted(model), resid(model), col = pointcol, pch = 20, xlab = "Fitted", ylab = "Residuals")
  abline(h = 0, col = linecol, lwd = 2)

}
```

```
library(car)
```

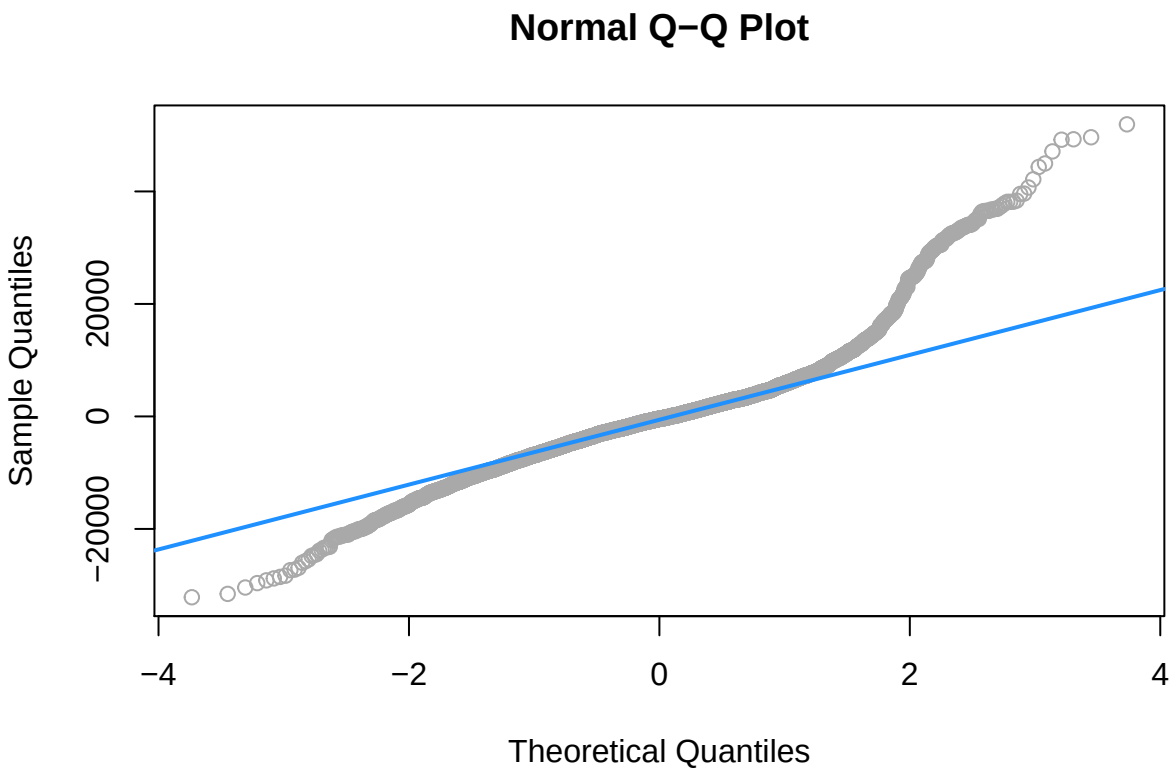
```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

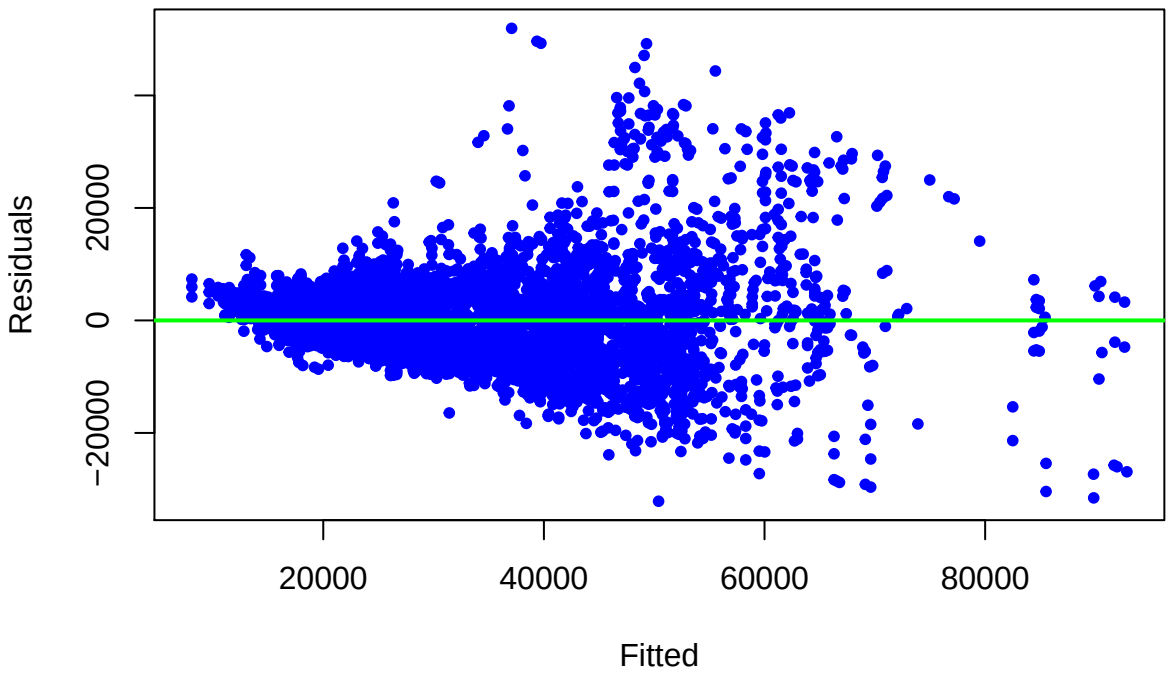
```
## The following object is masked from 'package:dplyr':
##
##     recode
```

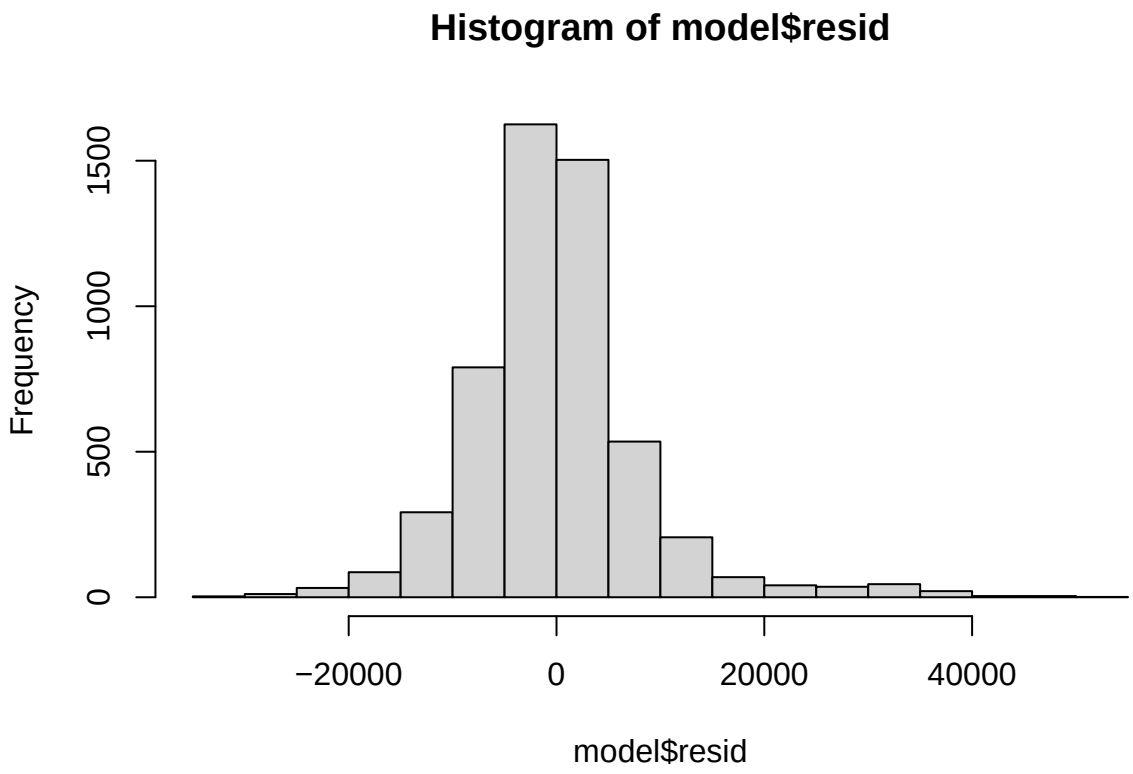
```
assumption_tester = function(model) {

  qqnorm(resid(model), main = "Normal Q-Q Plot", col = "darkgrey")
  qqline(resid(model), col = "dodgerblue", lwd = 2)

  #normality test
  print(shapiro.test(model$residuals[0:5000]))


  #multicollinearity
  vif = vif(model)
  print("Max VIF Value:")
  print(max(vif))
  print(vif)

  #Constant Variance
  plot_func(model)
  hist(model$resid)

}
```

```
assumption_tester(MSRP_big_mod)
```




```
##
##  Shapiro-Wilk normality test
##
## data:  model$residuals[0:5000]
## W = 0.91739, p-value < 2.2e-16
##
## [1] "Max VIF Value:"
## [1] 192.9373
##                              GVIF Df GVIF^(1/(2*Df))
## Engine.Fuel.Type         1.142934  1        1.069081
## Engine.HP               36.098960  1        6.008241
## Engine.Cylinders         9.567093  6        1.207068
## Transmission.Type        1.412183  1        1.188353
## Driven_Wheels            3.454874  3        1.229529
## Number.of.Doors          1.631708  2        1.130214
## Vehicle.Size             2.476926  2        1.254522
## highway.MPG            192.937269  1       13.890186
## city.mpg               102.828997  1       10.140463
## Popularity              10.315446  1        3.211767
## ReleasedYearsAgo        27.783388  1        5.270995
## I(Engine.HP^2)          25.542789  1        5.053987
## I(ReleasedYearsAgo^2)   26.010556  1        5.100055
## I(city.mpg^2)           68.372346  1        8.268757
## I(highway.MPG^2)       166.922914  1       12.919865
## I(Popularity^2)         10.079475  1        3.174819
```


Residuals
20000
0
−20000
20000
40000
60000
80000
Fitted

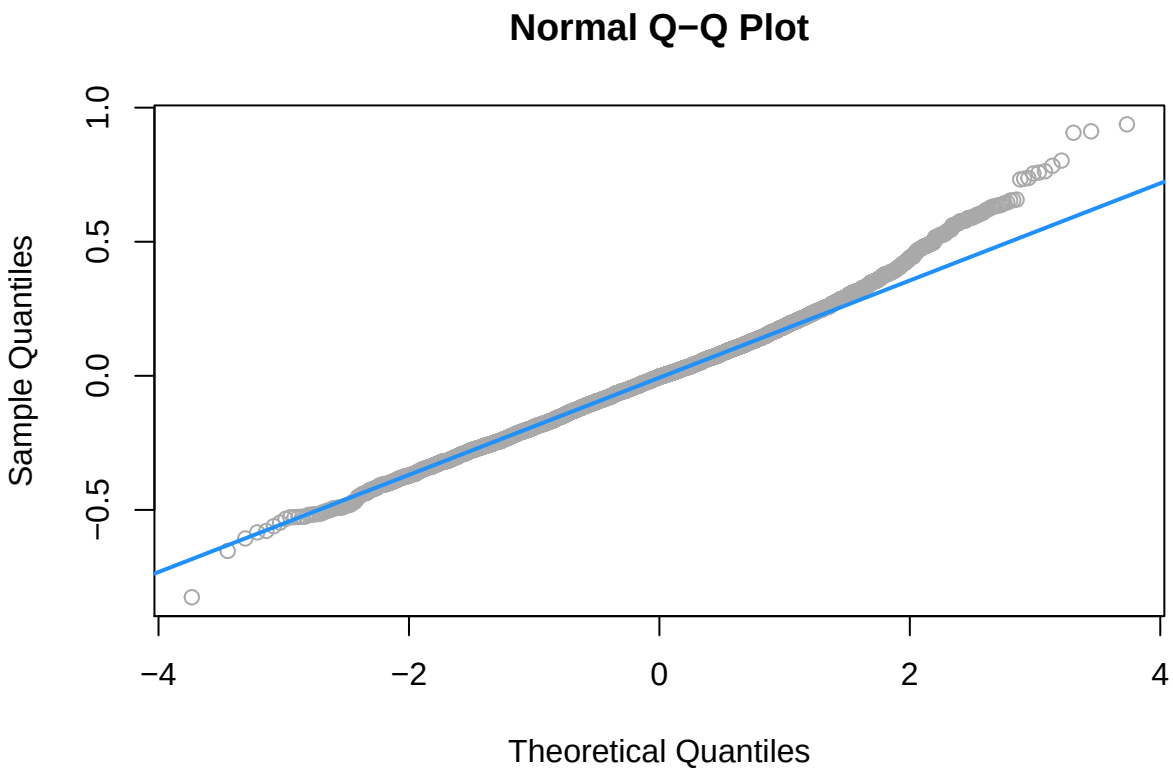



```
#alias(MSRP_mod_back_aic)
#assumption_tester(MSRP_mod_back_aic)
```

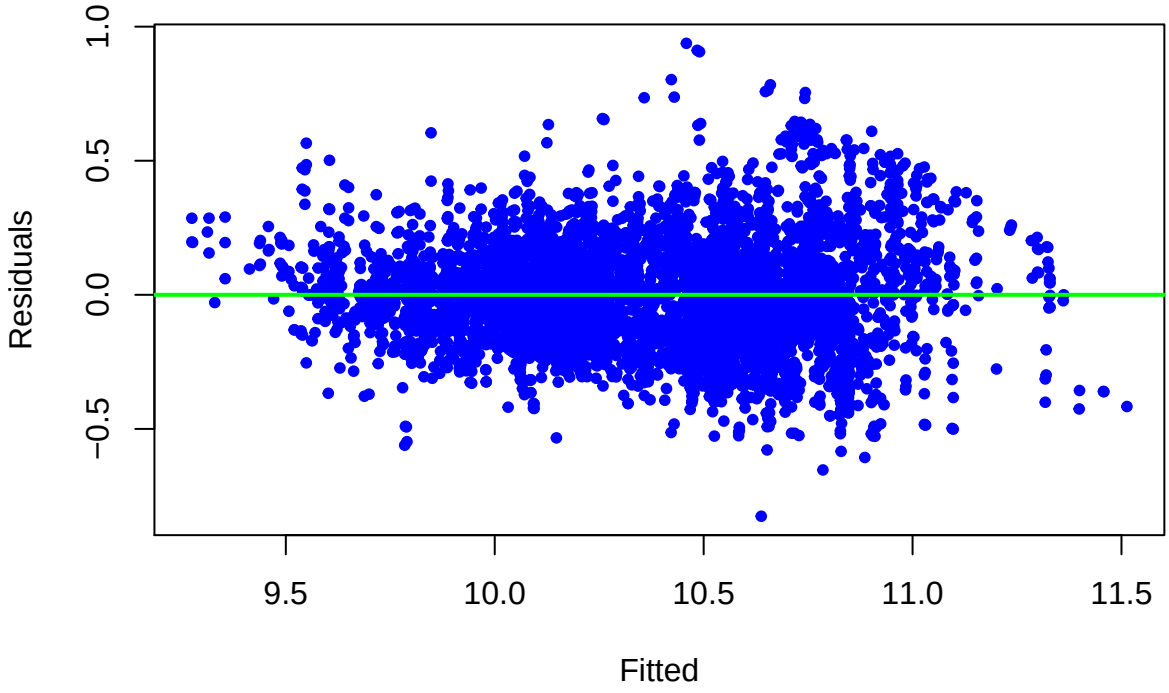
Making model improvements

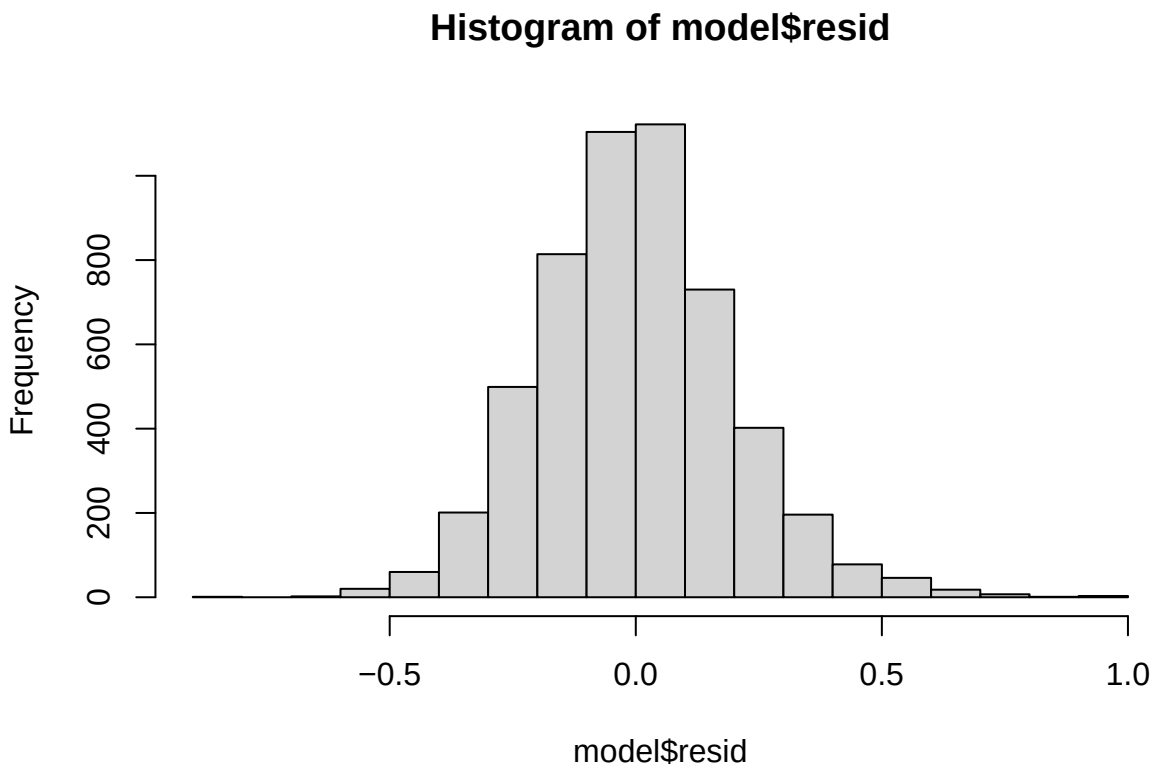
```
car_removed_predictors = lm(log(MSRP) ~ Engine.Fuel.Type + log(Engine.HP) +  Transmission.Type +
    Driven_Wheels + Number.of.Doors +
    I(ReleasedYearsAgo^2) + I(city.mpg^2) +
    I(highway.MPG^2) + I(Popularity^2), data = data_train)
```

```
assumption_tester(car_removed_predictors)
```




```
##
##  Shapiro-Wilk normality test
##
## data:  model$residuals[0:5000]
## W = 0.99171, p-value < 2.2e-16
##
## [1] "Max VIF Value:"
## [1] 6.593353
##                            GVIF Df GVIF^(1/(2*Df))
## Engine.Fuel.Type       1.088441  1        1.043284
## log(Engine.HP)         2.414690  1        1.553927
## Transmission.Type      1.247376  1        1.116860
## Driven_Wheels          2.296602  3        1.148632
## Number.of.Doors        1.370338  2        1.081949
## I(ReleasedYearsAgo^2) 1.708483  1        1.307090
## I(city.mpg^2)          4.169605  1        2.041961
## I(highway.MPG^2)       6.593353  1        2.567752
## I(Popularity^2)        1.021570  1        1.010728
```


Residuals
1.0
0.5
0.0
−0.5
9.5
10.0
10.5
11.0
11.5
Fitted




```
#summary(car_removed_predictors)
```

```
#plot( MSRP~Engine.HP, data = car_data_factored, scientific = FALSE)

#hist( car_data_factored$MSRP, scientific = FALSE)
```