**Analysis of Automobile Data Set**

### **Introduction:**

The main aim of this document is to explore and analyze the Automobile dataset in order to find a relationship from this data. The document have three main parts: data preparation and pre-processing, data analysis and data modeling. In each part there is some explanation about the steps that have been taken. Finally, in the data modeling part a linear regression model have been built. Also, the performance of the model have been improved two time to select the best variables that fits the model.

**Part 1 data preparation and pre-processing**

library(plyr)

library(ggplot2)

library(dplyr)

library(DT)

library(tidyr)

library(wesanderson)

library(mice)

#set the directory to file path

#loading the dataset

car\_data <- read.csv("Auto1-DS-TestData.csv", header = TRUE, stringsAsFactors = FALSE)

# Descriptive analysis to see the data types and some basic summary for every column in the data

str(car\_data)

'data.frame': 205 obs. of 26 variables:

$ symboling : int 3 3 1 2 2 2 1 1 1 0 ...

$ normalized.losses: chr "?" "?" "?" "164" ...

$ make : chr "alfa-romero" "alfa-romero" "alfa-romero" "audi" ...

$ fuel.type : chr "gas" "gas" "gas" "gas" ...

$ aspiration : chr "std" "std" "std" "std" ...

$ num.of.doors : chr "two" "two" "two" "four" ...

$ body.style : chr "convertible" "convertible" "hatchback" "sedan" ...

$ drive.wheels : chr "rwd" "rwd" "rwd" "fwd" ...

$ engine.location : chr "front" "front" "front" "front" ...

$ wheel.base : num 88.6 88.6 94.5 99.8 99.4 ...

$ length : num 169 169 171 177 177 ...

$ width : num 64.1 64.1 65.5 66.2 66.4 66.3 71.4 71.4 71.4 67.9 ...

$ height : num 48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 ...

$ curb.weight : int 2548 2548 2823 2337 2824 2507 2844 2954 3086 3053 ...

$ engine.type : chr "dohc" "dohc" "ohcv" "ohc" ...

$ num.of.cylinders : chr "four" "four" "six" "four" ...

$ engine.size : int 130 130 152 109 136 136 136 136 131 131 ...

$ fuel.system : chr "mpfi" "mpfi" "mpfi" "mpfi" ...

$ bore : chr "3.47" "3.47" "2.68" "3.19" ...

$ stroke : chr "2.68" "2.68" "3.47" "3.40" ...

$ compression.ratio: num 9 9 9 10 8 8.5 8.5 8.5 8.3 7 ...

$ horsepower : chr "111" "111" "154" "102" ...

$ peak.rpm : chr "5000" "5000" "5000" "5500" ...

$ city.mpg : int 21 21 19 24 18 19 19 19 17 16 ...

$ highway.mpg : int 27 27 26 30 22 25 25 25 20 22 ...

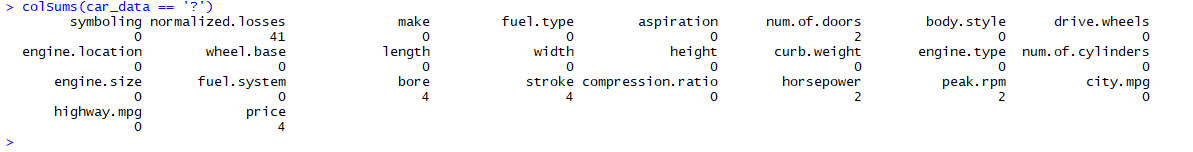
$ price : chr "13495" "16500" "16500" "13950"

summary(car\_data)

#after having a look on the obtained results we can see clearly that there is ? on the data

#its better to eliminate the those data and replace them.

colSums(car\_data == '?')



# I can see that ? appear in (normalized.losses 41 , num.of.doors 2, bore 4, stroke 4 ,

#horsepower 2, peak.rpm 2 and price 4)

# Well the main challenge is to find the

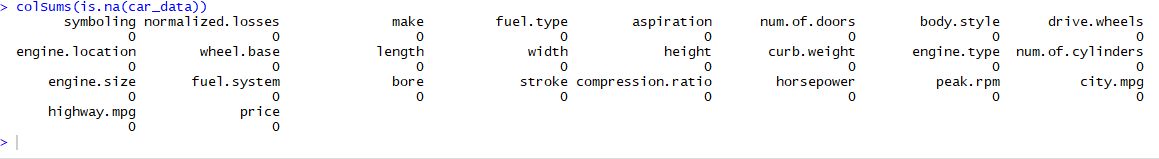
# 1- best way to replace the? With some values and there are many techniques (use the mean or remove rows)

# 2- the ? Appears in numerical and categorical values

# So let's try to find a pattern from the existing data we might be able to use that.

#before doing that lets check if there is any missing values as well.

colSums(is.na(car\_data))

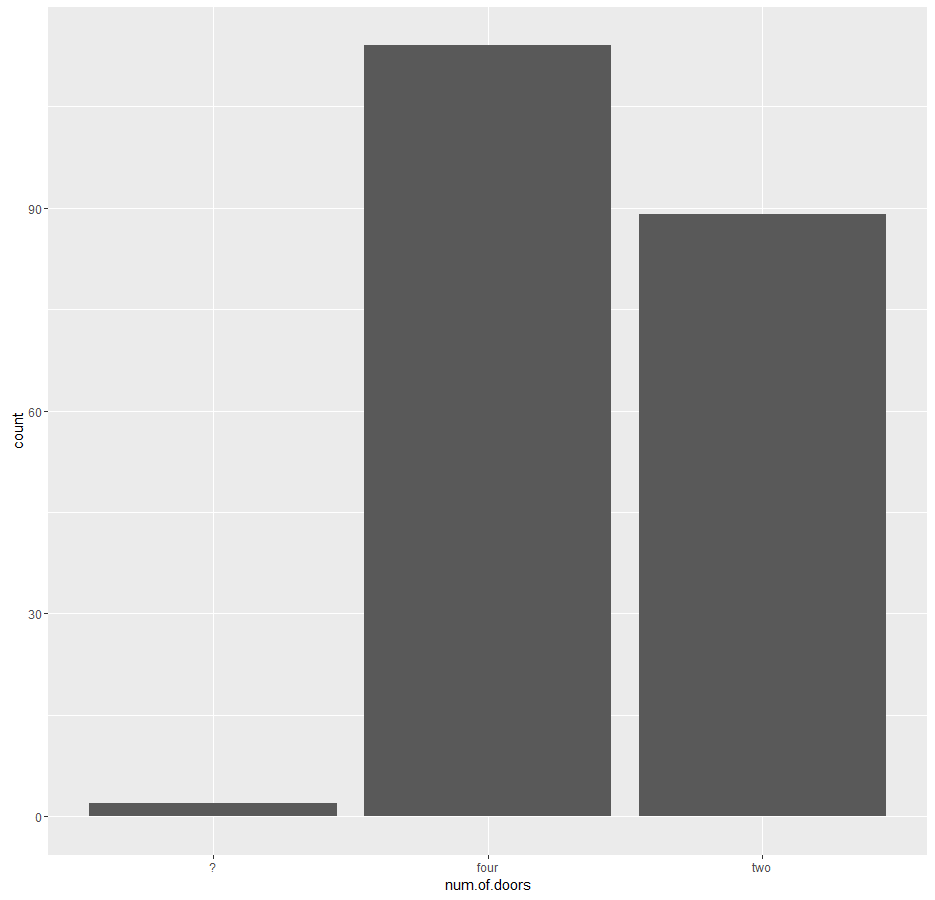


#Great there is no missing values appears among the observation.

#Now, let handle the ? for category (number of door), I will try to plot some graphs to understand better.

ggplot(data=car\_data, aes(x=num.of.doors)) +

geom\_bar(stat="count")



# as we can see that the there is ? in the data. in addition, we can see that the cars with four doors much bigger there is a high chance that ? means four

#lets try to see if we can make use of car brand with the ? values

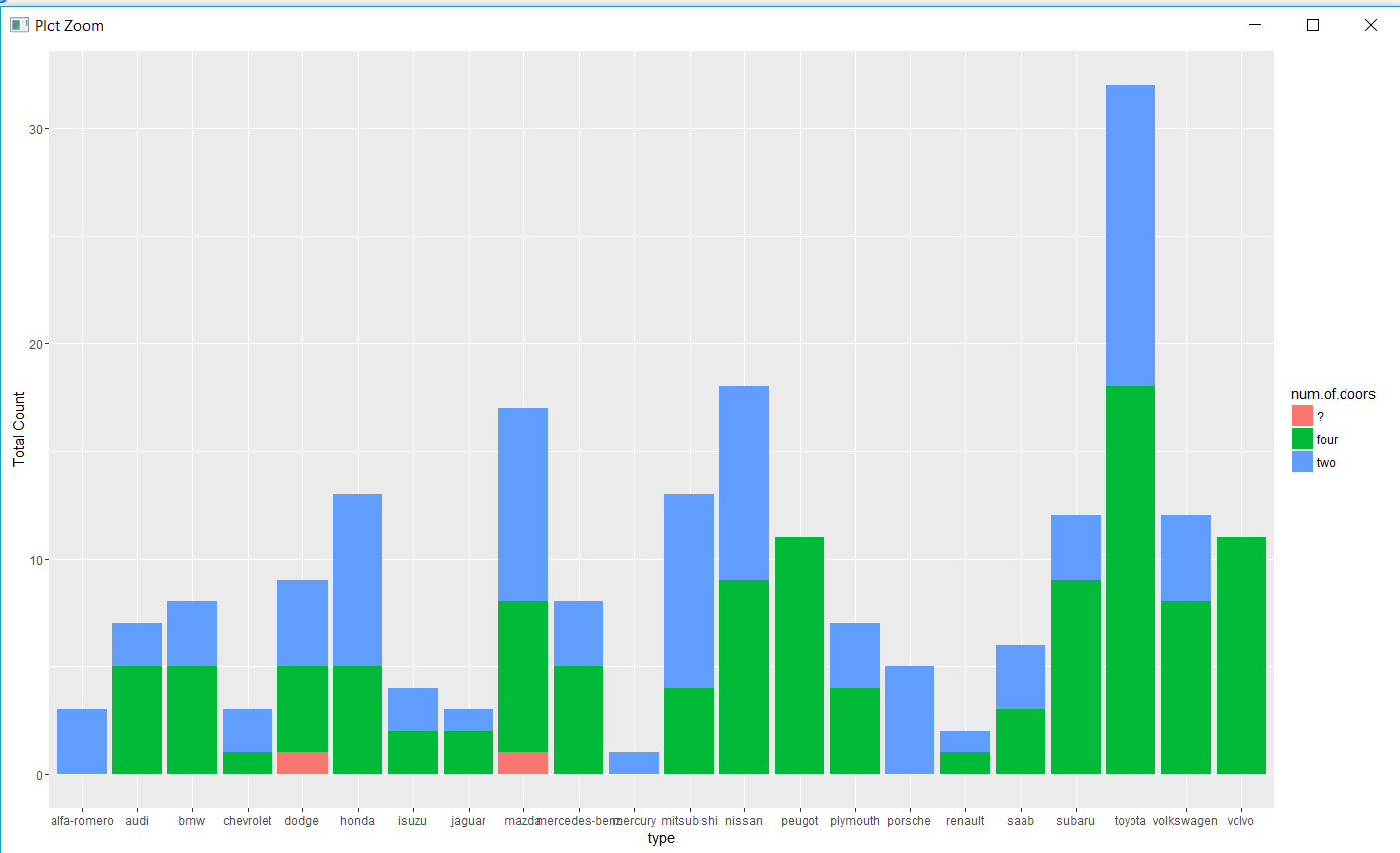
ggplot(car\_data, aes(x = make, fill = factor(num.of.doors))) +

geom\_bar() +

xlab("type") +

ylab("Total Count") +

labs(fill = "num.of.doors")



# Nice, we can see that ? Appears in two car brand which are the mazda and dodge cars. But, there is alot of cats also having two and four doors

# lets try to add body style variable it might makes us have better insights.

ggplot(car\_data, aes(x = num.of.doors, fill = body.style)) +

geom\_bar() +

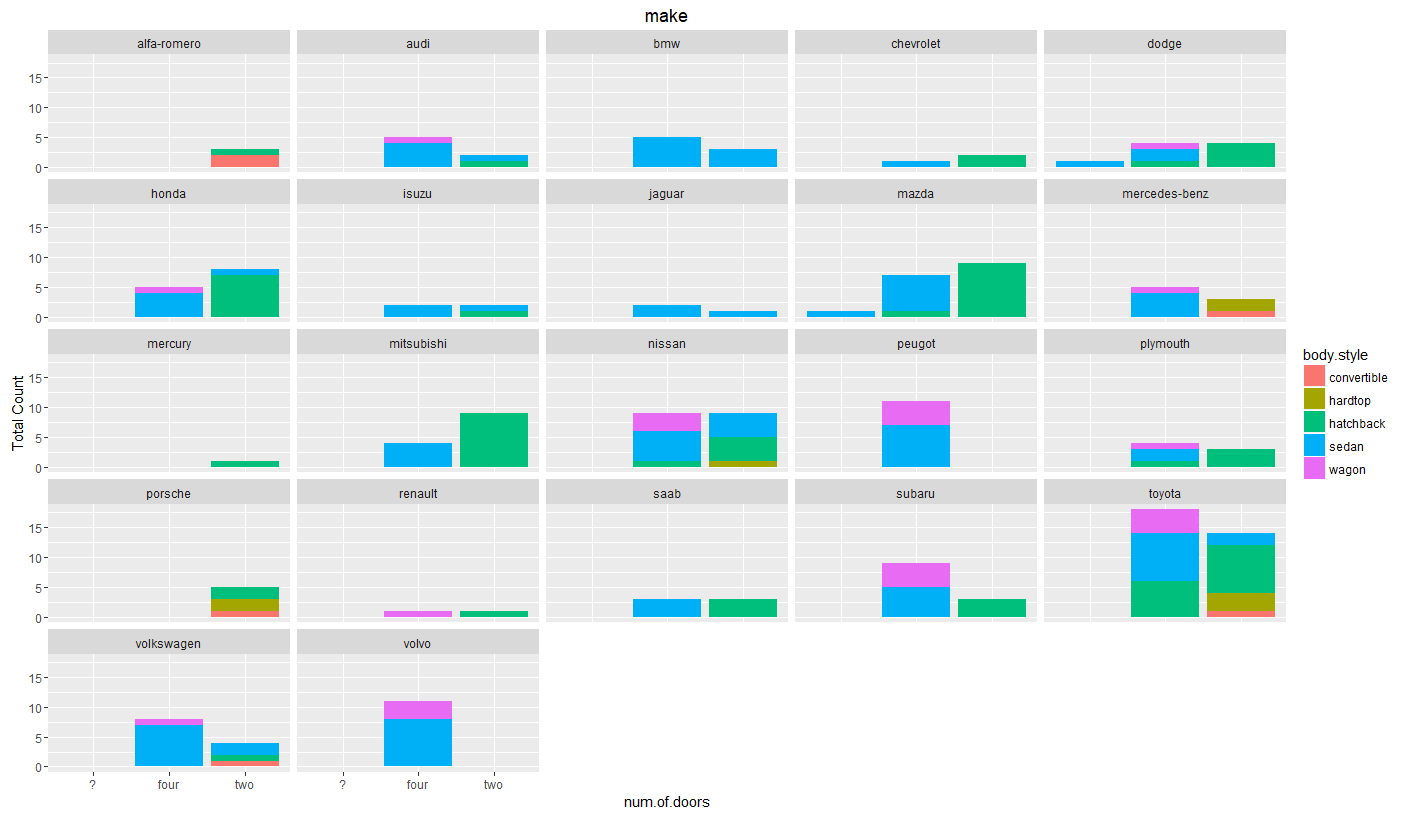
facet\_wrap(~make) +

ggtitle("make") +

xlab("num.of.doors") +

ylab("Total Count") +

labs(fill = "body.style")



#we can see clearly that the majority of sedan is four doors which make sense. I will replace the? with four

car\_data$num.of.doors[car\_data$num.of.doors=="?"] <- "four"

#lets have a quick look to the data now.

ggplot(car\_data, aes(x = num.of.doors, fill = factor(body.style))) +

geom\_bar() +

xlab("num.of.doors") +

ylab("Total Count") +

labs(fill = "body.style")

# the data looks much better. lets handle the other ? that appears in numerical variables.

#(normalized.losses 41 , bore 4, stroke 4 , horsepower 2, peak.rpm 2 and price 4)

car\_data$price[car\_data$price == "?"] <- NA

car\_data$peak.rpm[car\_data$peak.rpm == "?"] <- NA

car\_data$horsepower[car\_data$horsepower == "?"] <- NA

car\_data$stroke[car\_data$stroke == "?"] <- NA

car\_data$bore[car\_data$bore == "?"] <- NA

car\_data$normalized.losses[car\_data$normalized.losses == "?"] <- NA

str(car\_data)

#we need to convert the variables into numerical

car\_data$price<-as.numeric(as.character(car\_data$price))

car\_data$peak.rpm<-as.numeric(as.character(car\_data$peak.rpm))

car\_data$horsepower<-as.numeric(as.character(car\_data$horsepower))

car\_data$stroke<-as.numeric(as.character(car\_data$stroke))

car\_data$bore<-as.numeric(as.character(car\_data$bore))

car\_data$normalized.losses<-as.numeric(as.character(car\_data$normalized.losses))

# due to the sensitivity of the price its better we remove all the NA values in the price variable

car\_data<-subset(car\_data, !is.na(price))

# one of the awesome library is MICE which try to impute the missing values. lets try to use that.

imputed <- mice(car\_data,m=1,maxit=5,meth='pmm',seed=500)

#lets see the summary of the data

summary(imputed)

imputed$imp$horsepower

imputed$imp$stroke

imputed$imp$bore

imputed$imp$peak.rpm

imputed$imp$normalized.losses

#the first col is the number of column on the data set and the second one is the imputed data.

# replace the missing values with the imputed values

fulldata <- complete(imputed,1)

completedData <-fulldata

# The data is ready for modeling. Let's have some fun

**Part 2: Data Analysis**

# One of the important variable in cars is the millage

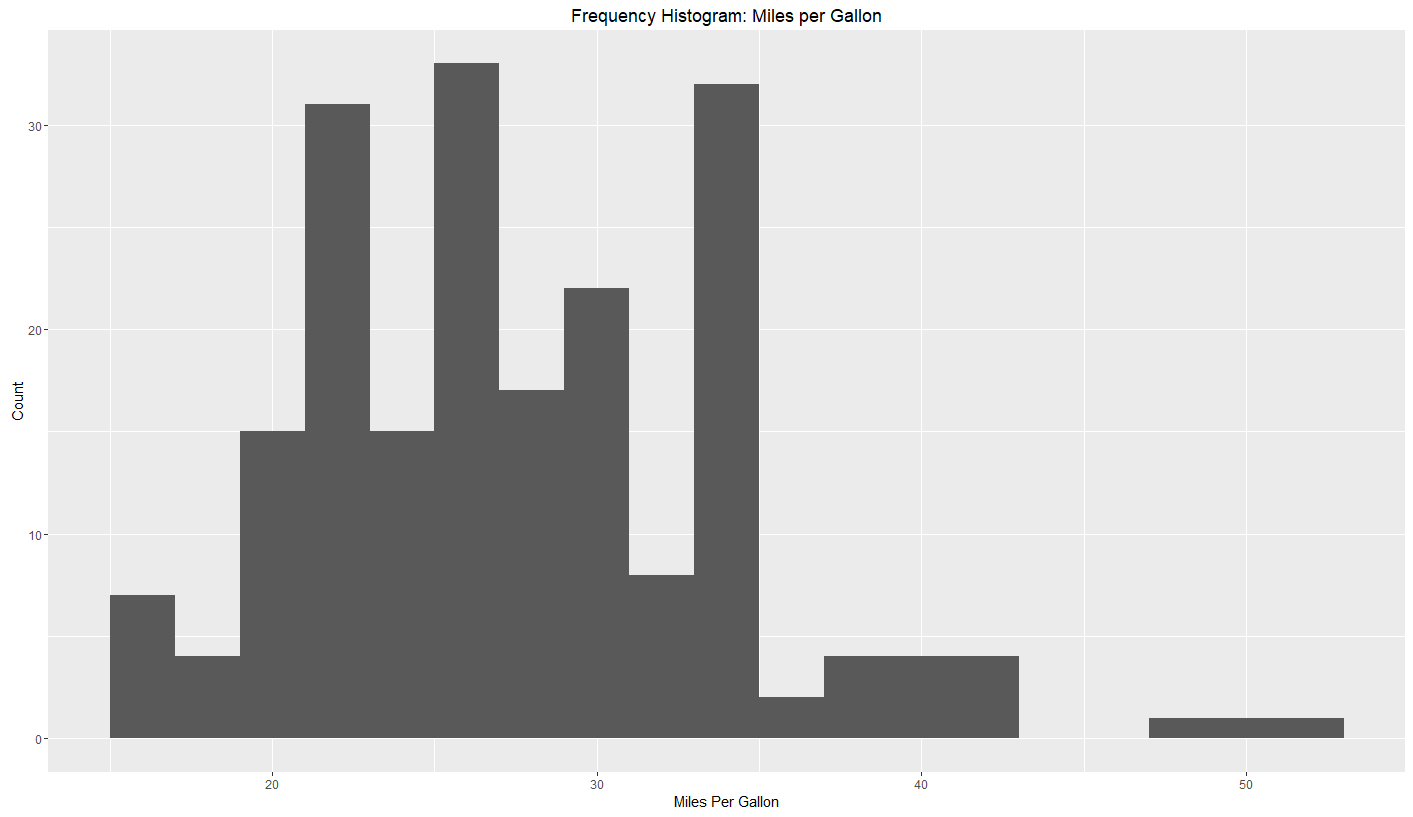
#so, let calculate the overall millage for the cars and do some analysis.

#Miles Per Gallon

completedData$mpg<- (completedData$city.mpg + completedData$highway.mpg)/2

qplot(completedData$mpg, xlab = 'Miles Per Gallon', ylab = 'Count', binwidth = 2,

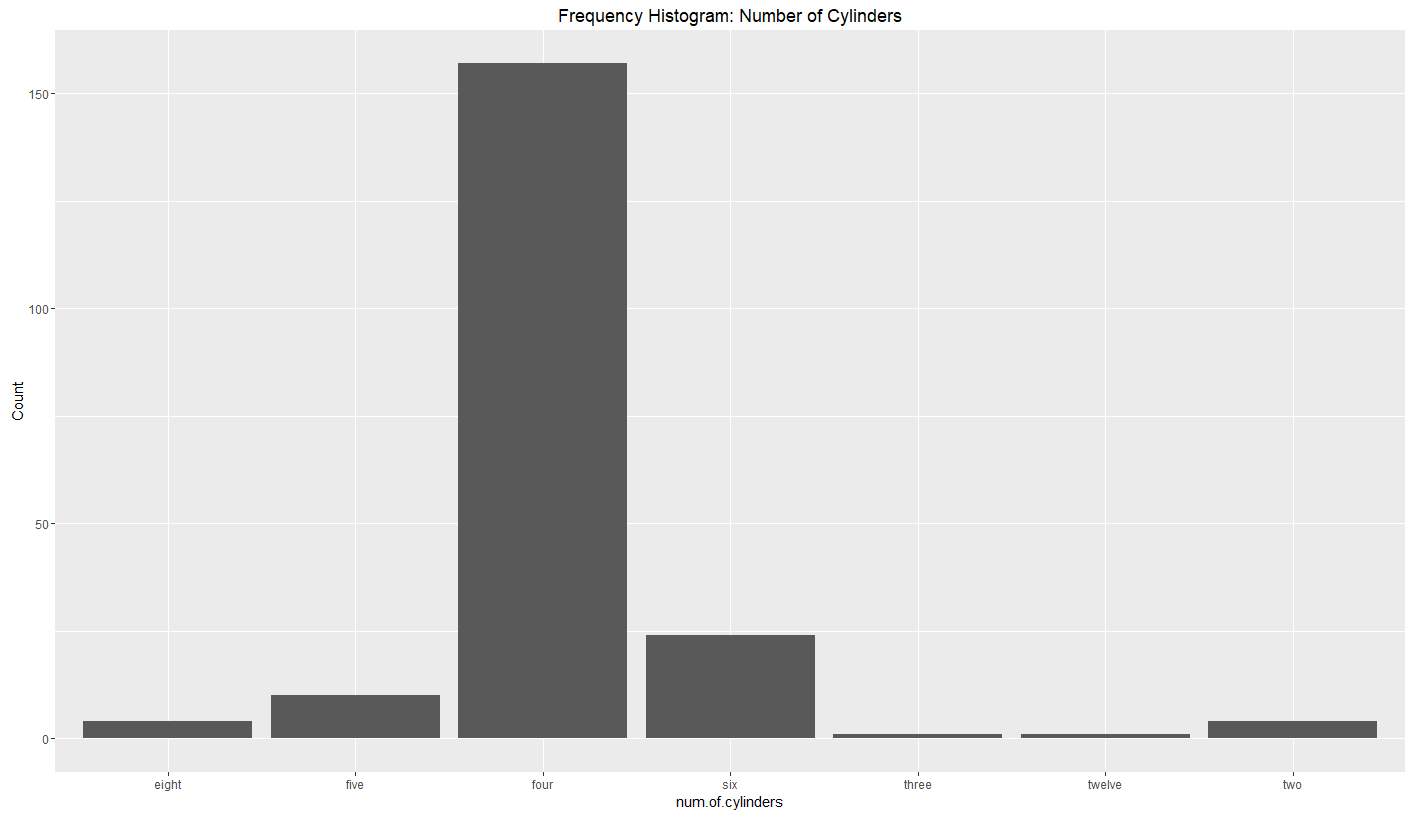
main='Frequency Histogram: Miles per Gallon')



#num.of.cylinders

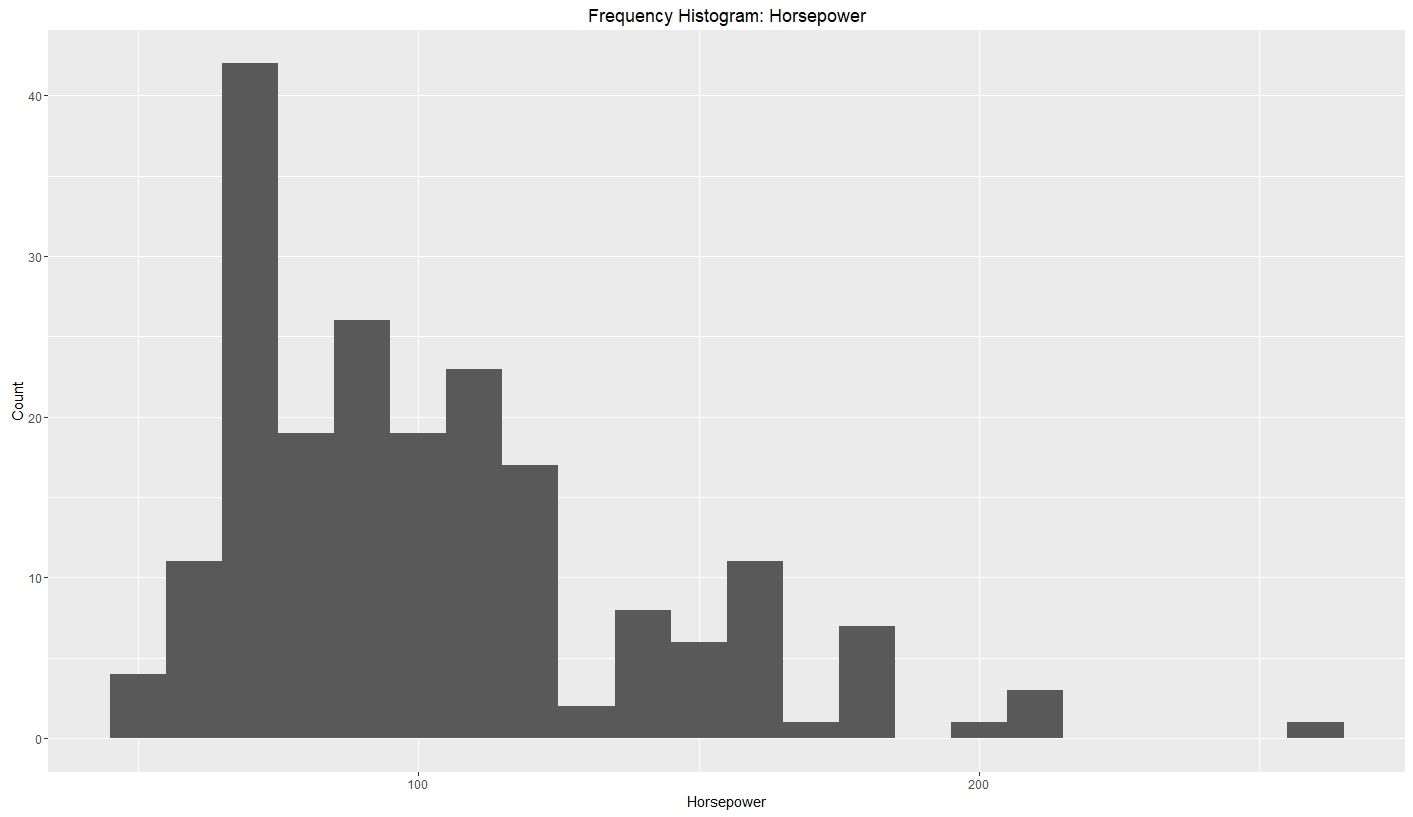
qplot(completedData$num.of.cylinders, xlab = 'num.of.cylinders', ylab = 'Count',

main='Frequency Histogram: Number of Cylinders')

 # Horsepower

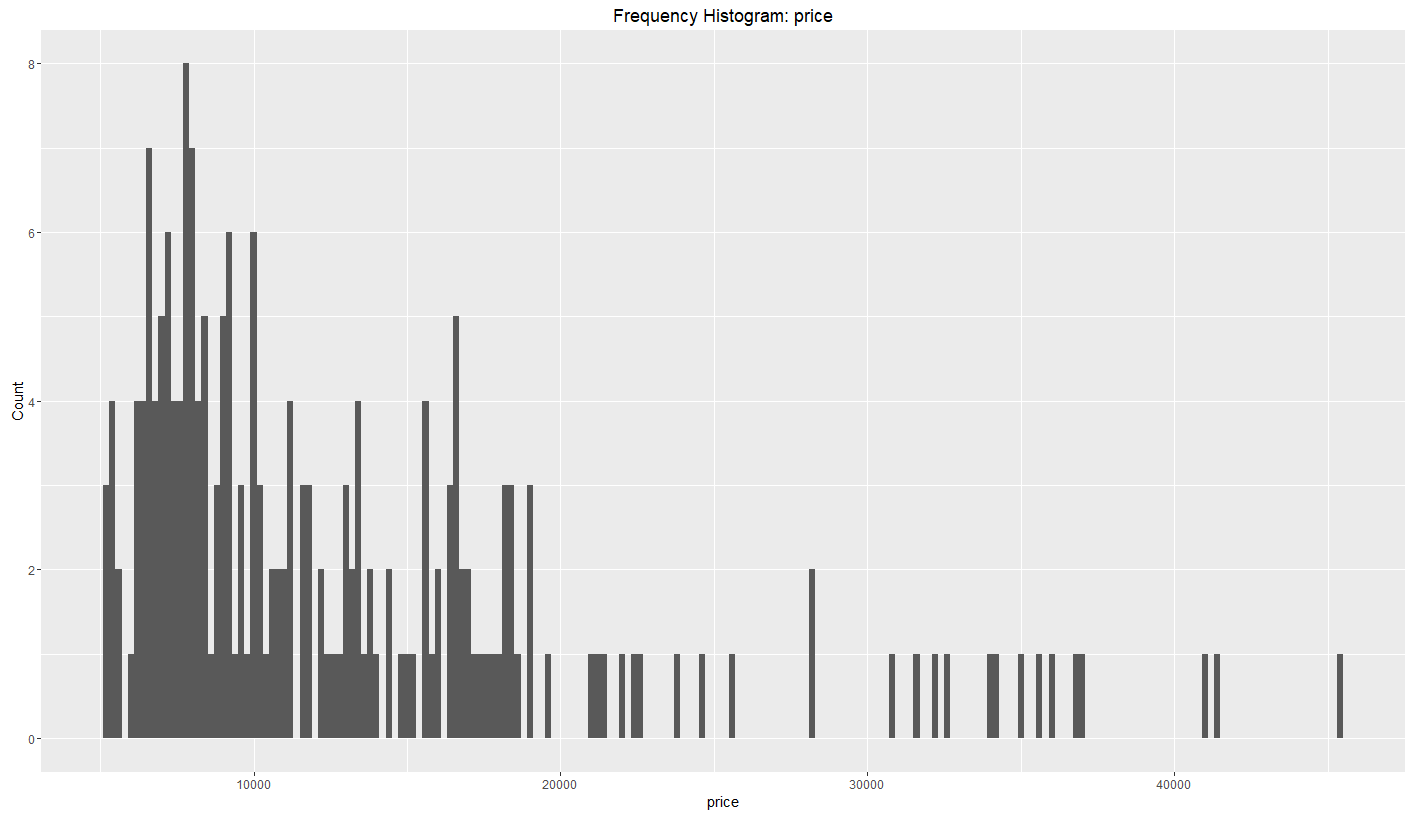
qplot(completedData$horsepower, xlab = 'Horsepower', ylab = 'Count', binwidth = 10,

main='Frequency Histogram: Horsepower')

 # price

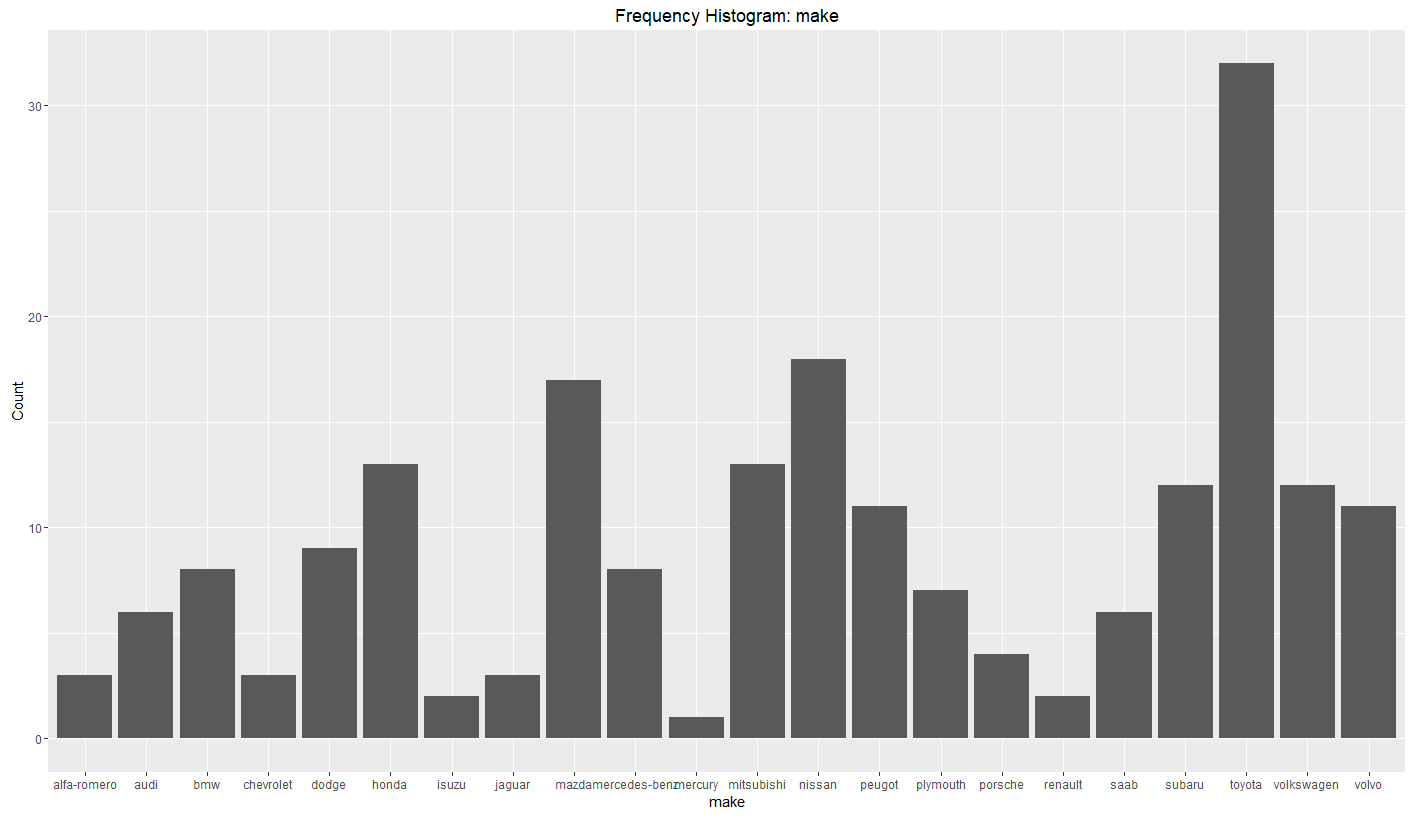
qplot(completedData$price, xlab = 'price', ylab = 'Count', binwidth = 200,

main='Frequency Histogram: price')



#make

qplot(completedData$make, xlab = 'make'**, ylab = 'Count', main='Frequency Histogram: make')**



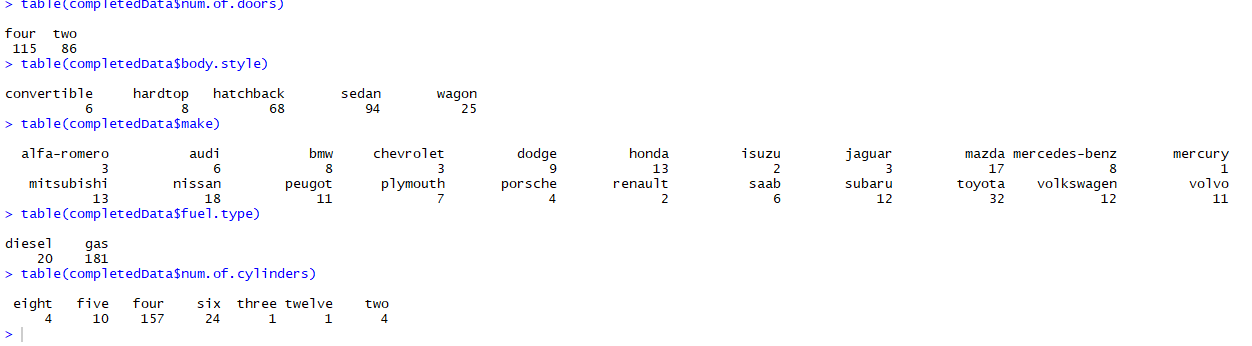
table(completedData$num.of.doors)

table(completedData$body.style)

table(completedData$make)

table(completedData$fuel.type)

table(completedData$num.of.cylinders)



#The results show us the number of cars with three and twelve cylinders are only 1. Which is very low that will affect the performance of the model. However we will keep them for the analysis.

#lets see the distribution of cars considering the body style and fuel type

ggplot(completedData, aes(x = fuel.type, fill = body.style)) +

geom\_bar() +

facet\_wrap(~make) +

ggtitle(" cars distrubution with respect to body style and fuel type") +

xlab("fuel.type") +

ylab("Total Count") +

labs(fill = "body.style")

# ONE important notice here that the majority of the cars are gas (sedan and hatchback) are the most common in the market



#lets see if we can use the num.of.cylinders to understand any pattern.

ggplot(completedData, aes(x = fuel.type, fill = num.of.cylinders)) +

geom\_bar() +

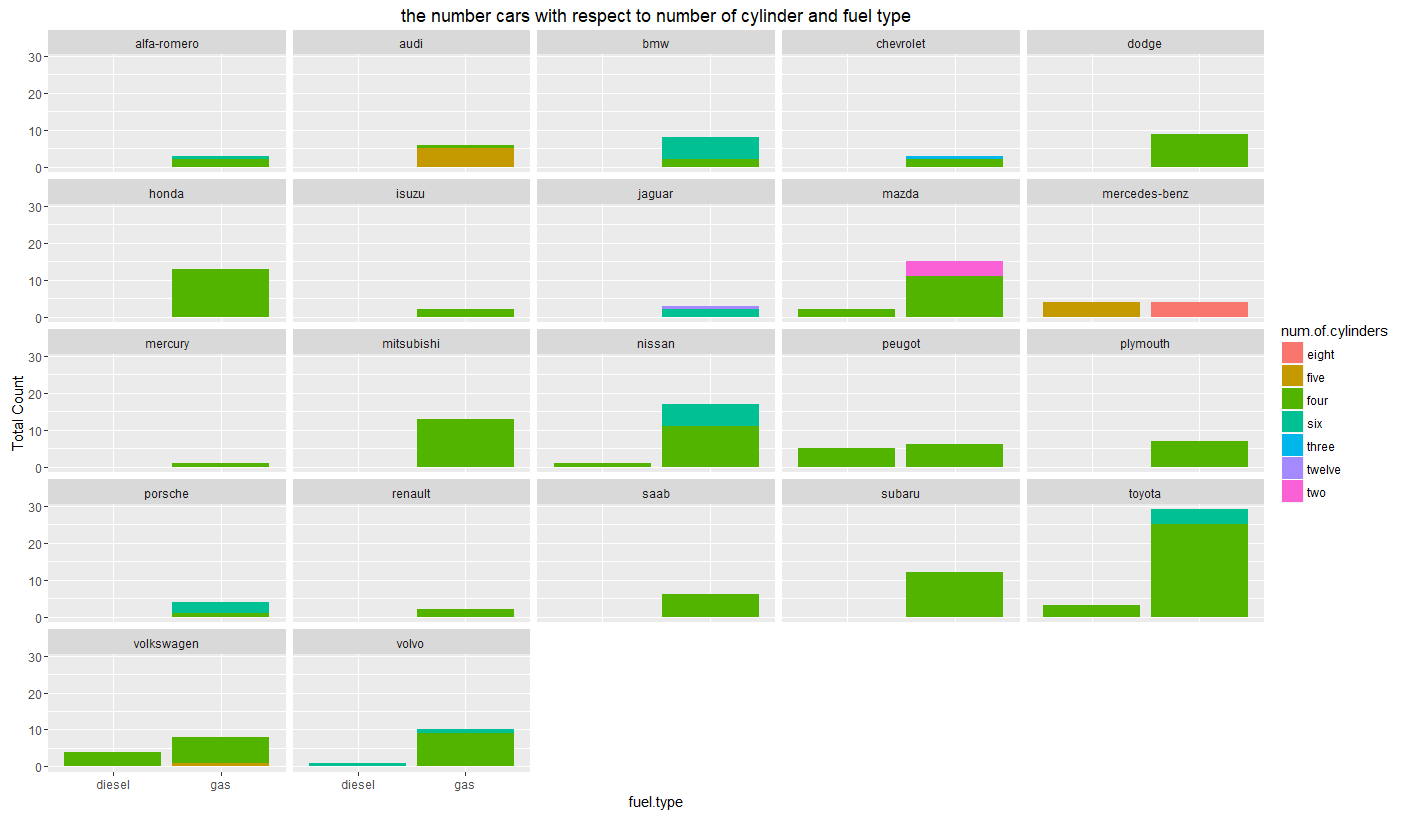
facet\_wrap(~make) +

ggtitle("the number cars with respect to number of cylinder and fuel type") +

xlab("fuel.type") +

ylab("Total Count") +

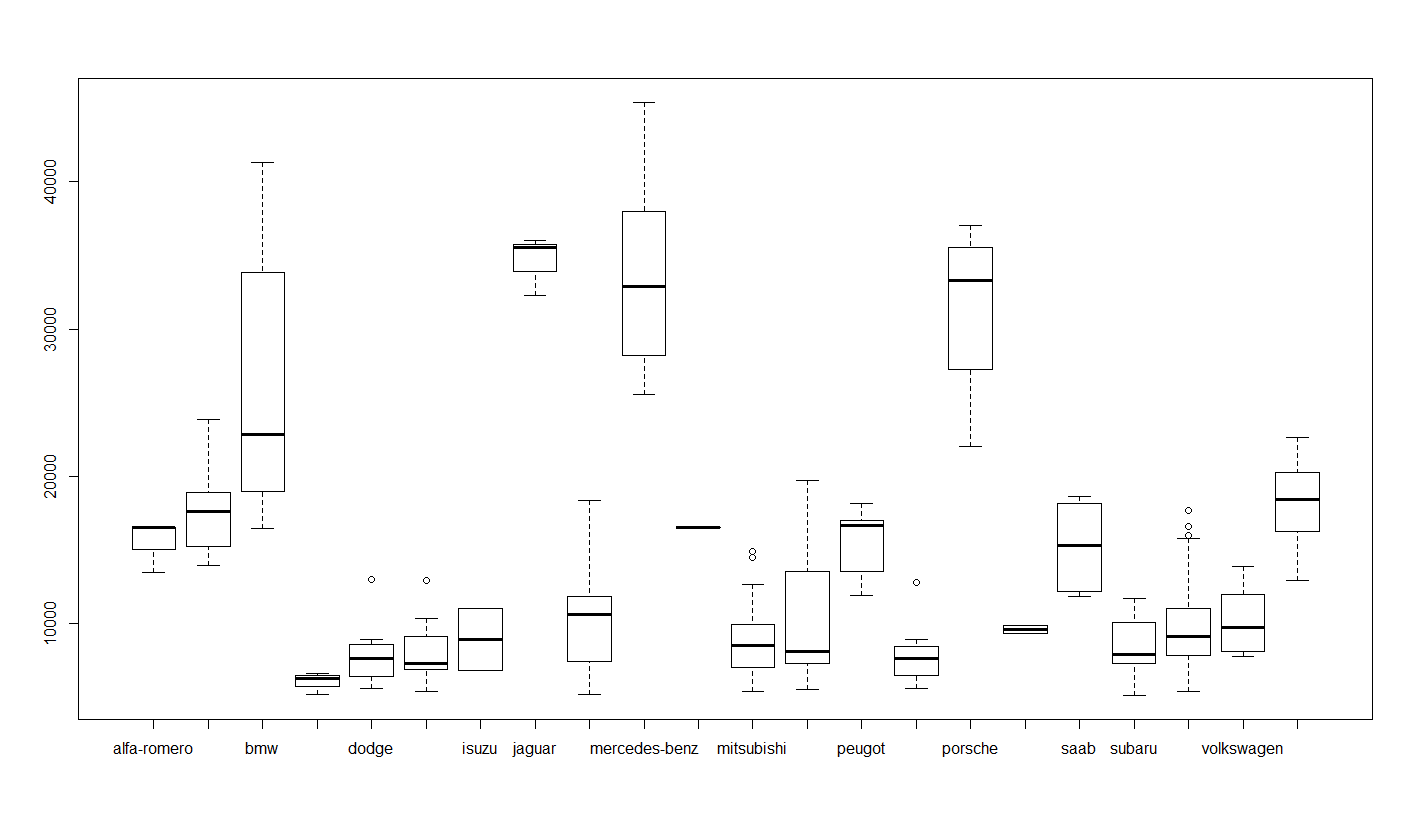
labs(fill = "num.of.cylinders")



#interesting we can see that the majority of the cars are having four cylinder

#as we can see toyota, Honda and mitsubishi maybe due to there economy in term of fuel consumption

boxplot(completedData$price ~ completedData$make)



# the boxplot shows us that the majority of the car prices are below 20000

#Plotting the cars brand with the millage

top\_1 <- completedData %>%

group\_by(make, body.style, mpg) %>%

summarize(total = sum(price)) %>%

top\_n(1)

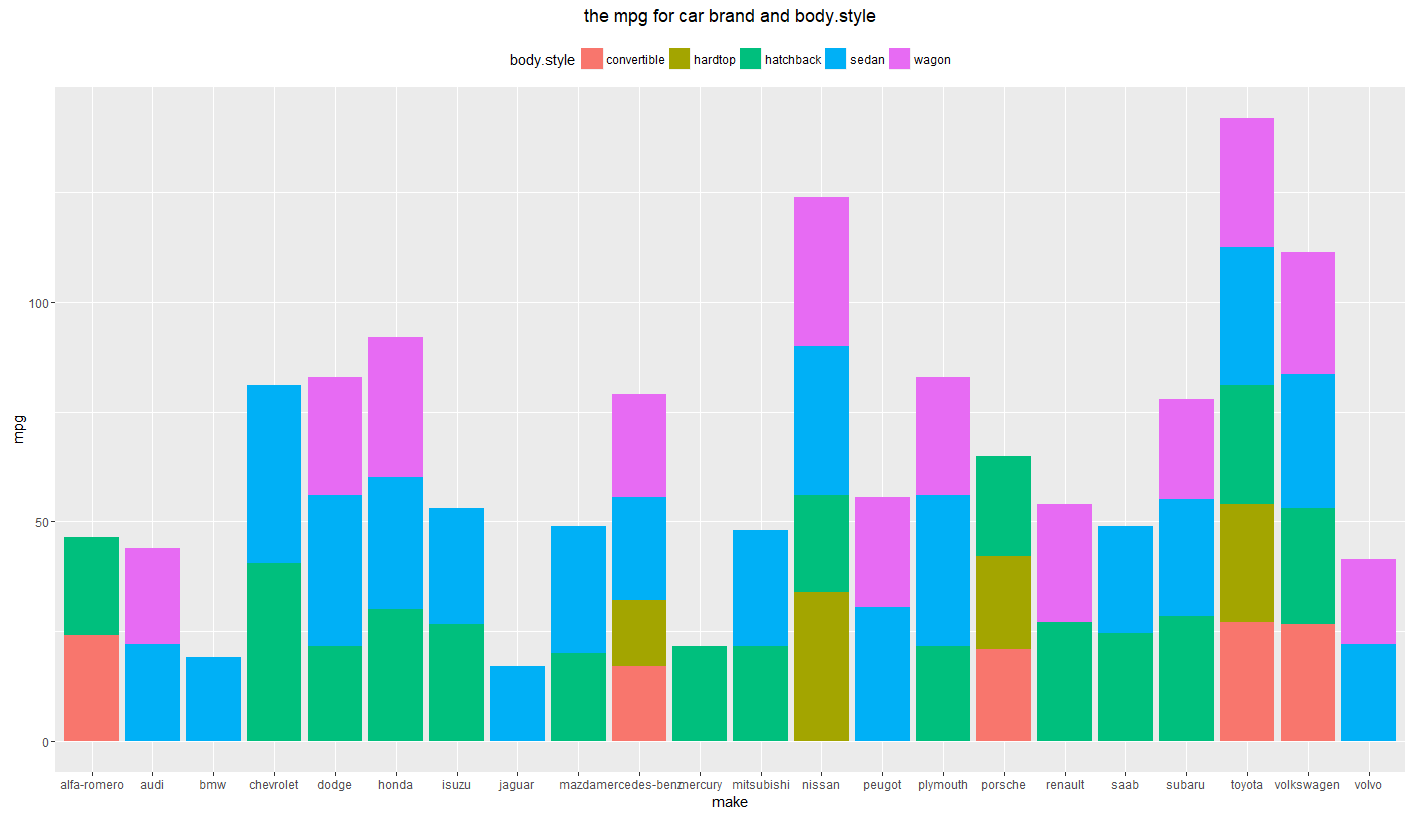
#we can see the car with the highest millage are toyota and vlokswagen

ggplot(top\_1, aes(make, mpg, fill = body.style)) +

geom\_bar(stat = "identity") +

ggtitle("the mpg for car brand and body.style") +

theme(legend.position = "top")

# the figure indicate that the hatchback and sedan car have the lowest mpg among between the cars

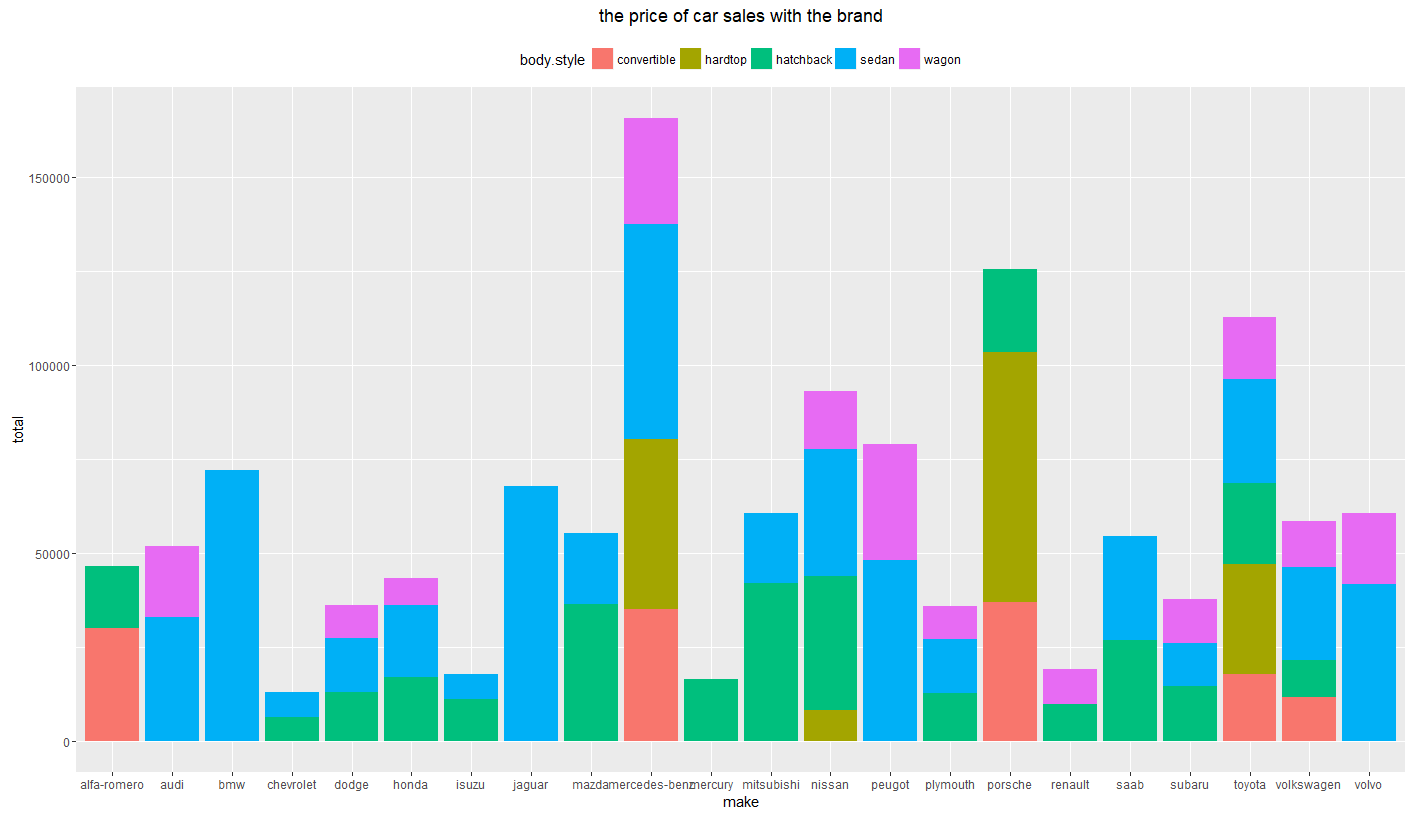
# let's see the overall price for the car

ggplot(top\_1, aes(make, total, fill = body.style)) +

geom\_bar(stat = "identity") +

ggtitle("the price of car sales with the brand") +

theme(legend.position = "top")



#the buyers also consider the price of the cars you can see the price of hatchback and sedan lower comparing to others to other cars

#so i assume that the cars with lower millage have high price let's see the relationship of the cars there.

ggplot(data = completedData, aes(x = price, y = mpg)) +

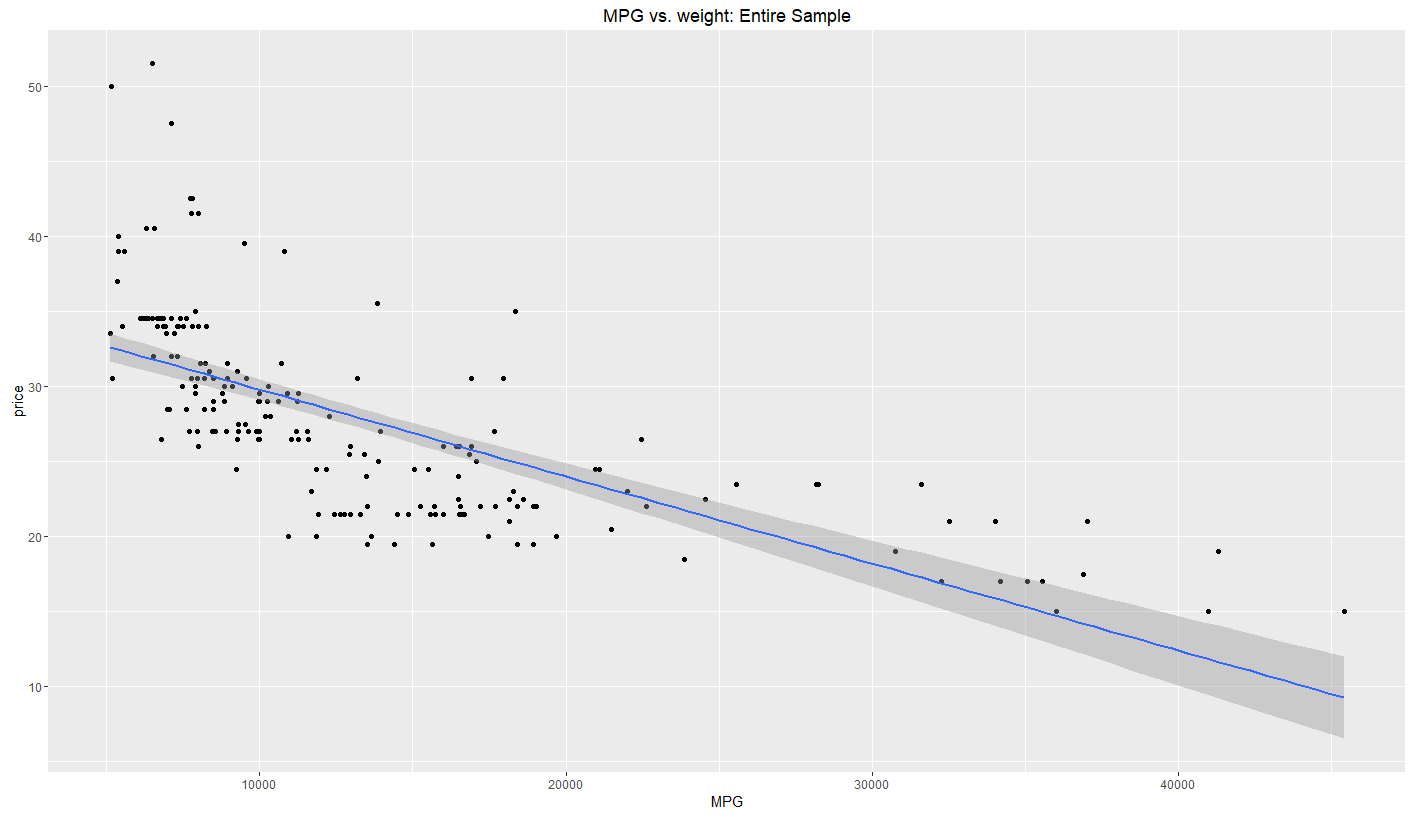
geom\_point() +

geom\_smooth(method='lm') +

xlab('MPG') +

ylab('price') +

ggtitle('MPG vs. weight: Entire Sample')

#cool then my assumption is correct which is when the mpg is low the price is high

#then we can say that the most profitable cars to sell in the market are sedan and hatchback.

#lets try to check the distrbution of car and the symboling affect the car price

ggplot(data = completedData, aes(x = symboling, y = price)) +

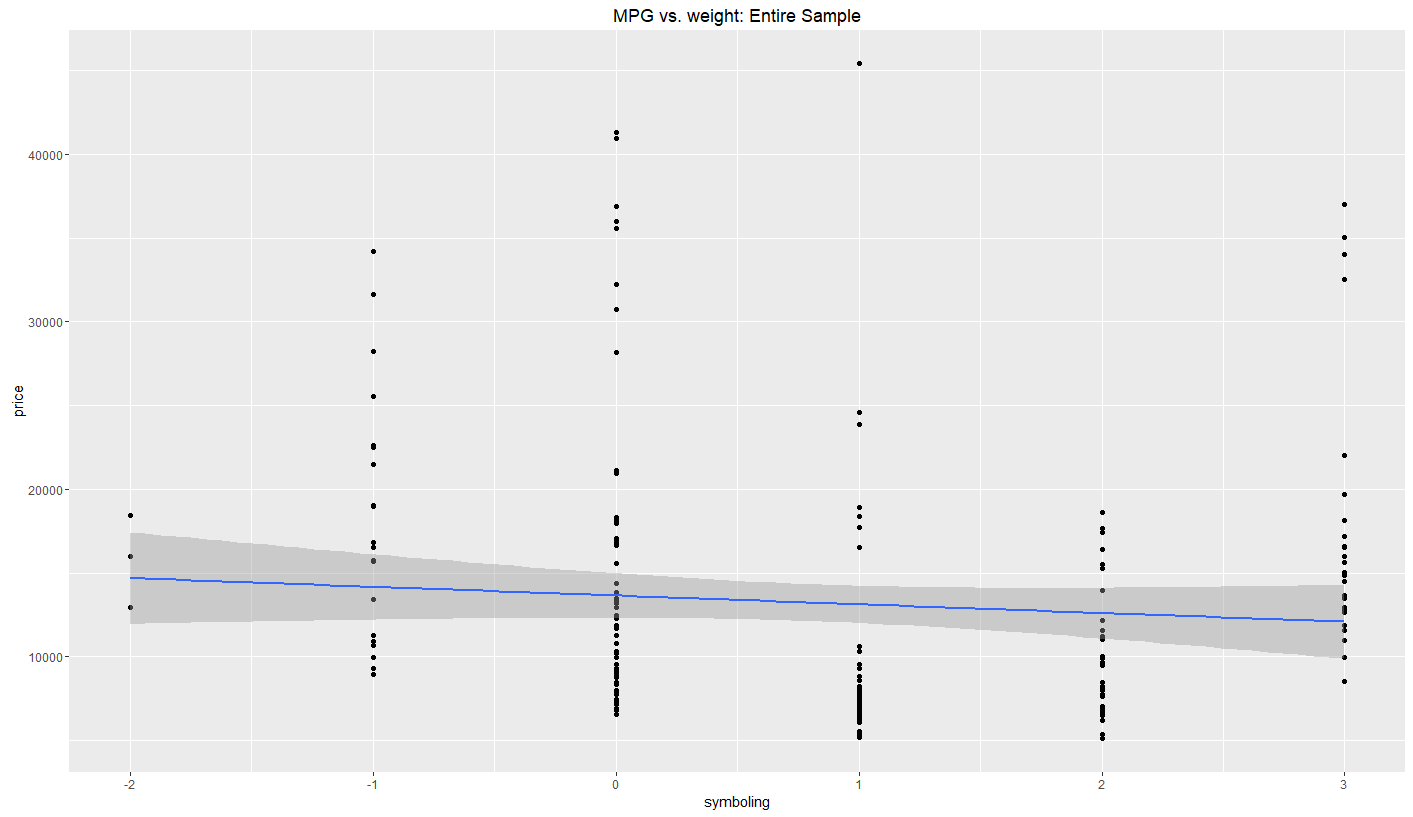
geom\_point() +

geom\_smooth(method='lm') +

xlab('symboling') +

ylab('price') +

ggtitle('MPG vs. weight: Entire Sample')



# there is positive relationship between the cars and risk variable, we can see that the majority of the cars are risky. Which is make since because the cars are used so the risk is high.

Part 3: Data Modeling

# # split the data into training and testing 80/20

set.seed(1234)

ind<- sample(2, nrow(fulldata), replace=T , prob = c(0.8,0.2))

train <- fulldata[ind==1,]

test <- fulldata[ind==2,]

#let’s hypothesize that price are related to other variables

# null hypothesis (H0). Here, our null hypothesis is that price and other variables are not related.

#H1: There is some relationship between price and other variables.

lm<- lm(price~.,data=train)

summary(lm)

Call:

lm(formula = price ~ ., data = train)

Residuals:

Min 1Q Median 3Q Max

-2703.2 -699.1 0.0 748.7 2790.6

Coefficients: (5 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) -2.803e+03 1.947e+04 -0.144 0.885799

symboling -2.520e+02 2.446e+02 -1.031 0.305172

normalized.losses -3.442e-01 8.072e+00 -0.043 0.966068

makeaudi 3.552e+03 2.355e+03 1.508 0.134594

makebmw 7.828e+03 2.391e+03 3.274 0.001446 \*\*

makechevrolet -2.013e+03 2.209e+03 -0.911 0.364294

makedodge -3.647e+03 1.910e+03 -1.909 0.058989 .

makehonda 6.998e+02 2.056e+03 0.340 0.734320

makeisuzu -1.857e+03 2.149e+03 -0.864 0.389352

makejaguar -2.613e+02 2.419e+03 -0.108 0.914185

makemazda 3.219e+02 1.754e+03 0.183 0.854787

makemercedes-benz 2.220e+03 2.205e+03 1.007 0.316404

makemercury -1.625e+03 2.859e+03 -0.568 0.570955

makemitsubishi -3.271e+03 1.908e+03 -1.714 0.089542 .

makenissan 9.271e+02 1.823e+03 0.509 0.612103

makepeugot -1.705e+04 4.633e+03 -3.680 0.000373 \*\*\*

makeplymouth -2.613e+03 1.868e+03 -1.399 0.164882

makeporsche 1.998e+04 2.792e+03 7.158 1.24e-10 \*\*\*

makerenault 7.244e+02 2.181e+03 0.332 0.740505

makesaab 5.530e+03 2.328e+03 2.375 0.019383 \*

makesubaru 1.116e+03 2.251e+03 0.496 0.621225

maketoyota -1.406e+03 1.636e+03 -0.859 0.392267

makevolkswagen 2.087e+02 1.844e+03 0.113 0.910100

makevolvo 6.175e+01 2.391e+03 0.026 0.979444

fuel.typegas -1.042e+04 6.802e+03 -1.532 0.128628

aspirationturbo 2.121e+03 7.875e+02 2.694 0.008247 \*\*

num.of.doorstwo -6.323e+02 4.847e+02 -1.304 0.194987

body.stylehardtop -3.034e+03 1.202e+03 -2.525 0.013102 \*

body.stylehatchback -2.811e+03 1.146e+03 -2.453 0.015842 \*

body.stylesedan -2.648e+03 1.206e+03 -2.196 0.030324 \*

body.stylewagon -2.692e+03 1.306e+03 -2.061 0.041796 \*

drive.wheelsfwd -1.230e+00 9.448e+02 -0.001 0.998964

drive.wheelsrwd 1.981e+03 1.211e+03 1.636 0.104904

engine.locationrear NA NA NA NA

wheel.base 2.831e+02 8.350e+01 3.391 0.000989 \*\*\*

length -1.805e+02 4.833e+01 -3.734 0.000309 \*\*\*

width 7.712e+02 2.292e+02 3.364 0.001079 \*\*

height -3.329e+02 1.432e+02 -2.325 0.022015 \*

curb.weight 6.801e+00 1.527e+00 4.453 2.16e-05 \*\*\*

engine.typel 1.165e+04 4.831e+03 2.411 0.017688 \*

engine.typeohc -2.354e+01 1.112e+03 -0.021 0.983153

engine.typeohcf NA NA NA NA

engine.typeohcv -3.110e+03 1.104e+03 -2.816 0.005825 \*\*

engine.typerotor 1.673e+04 6.483e+03 2.580 0.011279 \*

num.of.cylindersfive 9.836e+02 3.350e+03 0.294 0.769667

num.of.cylindersfour 5.249e+03 4.251e+03 1.235 0.219763

num.of.cylinderssix -9.907e+02 2.740e+03 -0.362 0.718451

num.of.cylindersthree NA NA NA NA

num.of.cylinderstwelve -8.325e+03 4.978e+03 -1.672 0.097500 .

num.of.cylinderstwo NA NA NA NA

engine.size 1.414e+02 3.306e+01 4.277 4.25e-05 \*\*\*

fuel.system2bbl 3.348e+03 1.272e+03 2.632 0.009785 \*\*

fuel.system4bbl -2.373e+03 2.489e+03 -0.953 0.342603

fuel.systemidi NA NA NA NA

fuel.systemmpfi 2.599e+03 1.361e+03 1.910 0.058932 .

fuel.systemspdi 1.802e+03 1.731e+03 1.041 0.300124

fuel.systemspfi 3.091e+03 2.498e+03 1.237 0.218725

bore -1.073e+04 2.901e+03 -3.700 0.000348 \*\*\*

stroke -3.054e+03 9.482e+02 -3.221 0.001712 \*\*

compression.ratio -6.035e+02 5.005e+02 -1.206 0.230660

horsepower 2.239e+00 2.532e+01 0.088 0.929697

peak.rpm 1.153e+00 5.965e-01 1.934 0.055910 .

city.mpg -5.739e+01 1.169e+02 -0.491 0.624591

highway.mpg 9.833e+01 1.014e+02 0.970 0.334216

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1362 on 103 degrees of freedom

Multiple R-squared: 0.9798, Adjusted R-squared: 0.9684

F-statistic: 86.19 on 58 and 103 DF, p-value: < 2.2e-16

# by looking at the summary of the model we can see that the model have significant value with some parameters. So we have evidence to reject H0

# is Our multiple R2 value is also a little higher than our adjusted R2.

#R2 and R2adj. The closer these values are to 1, the better the model explains the label. and the p-values are smaller than 0.05, we reject H0. There to be a relationship between price and some variables in the model

#In the other hand, there are many parameters are no significant on the model, it would be better if we attempt to build another model with the parameters that have a significant value to the model

**# Model improvement**

# i have picked up parameters that have a highly correspondent to our model.

lm2<- lm(price ~ make + aspiration + body.style + wheel.base + length + width + height + curb.weight + num.of.cylinders + engine.size + peak.rpm + fuel.type + engine.location,data=train)

summary(lm2)

Call:

lm(formula = price ~ make + aspiration + body.style + wheel.base +

length + width + height + curb.weight + num.of.cylinders +

engine.size + peak.rpm + fuel.type + engine.location, data = train)

Residuals:

Min 1Q Median 3Q Max

-3605.9 -838.4 -1.5 742.2 4603.2

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) -3.791e+04 1.414e+04 -2.681 0.008367 \*\*

makeaudi -1.097e+03 1.805e+03 -0.608 0.544417

makebmw 5.592e+03 1.380e+03 4.054 8.95e-05 \*\*\*

makechevrolet -4.685e+03 1.711e+03 -2.738 0.007120 \*\*

makedodge -5.852e+03 1.467e+03 -3.989 0.000114 \*\*\*

makehonda -4.618e+03 1.436e+03 -3.215 0.001672 \*\*

makeisuzu -3.648e+03 1.629e+03 -2.240 0.026929 \*

makejaguar 2.672e+03 1.910e+03 1.399 0.164497

makemazda -2.585e+03 1.314e+03 -1.967 0.051495 .

makemercedes-benz 1.296e+03 2.102e+03 0.617 0.538570

makemercury -4.491e+03 2.125e+03 -2.113 0.036623 \*

makemitsubishi -6.349e+03 1.325e+03 -4.793 4.72e-06 \*\*\*

makenissan -3.446e+03 1.235e+03 -2.791 0.006106 \*\*

makepeugot -6.862e+03 1.579e+03 -4.347 2.89e-05 \*\*\*

makeplymouth -4.757e+03 1.446e+03 -3.289 0.001316 \*\*

makeporsche 1.766e+04 1.748e+03 10.105 < 2e-16 \*\*\*

makerenault -4.514e+03 1.668e+03 -2.706 0.007795 \*\*

makesaab 6.380e+01 1.498e+03 0.043 0.966096

makesubaru -4.489e+03 1.344e+03 -3.339 0.001119 \*\*

maketoyota -4.114e+03 1.209e+03 -3.403 0.000906 \*\*\*

makevolkswagen -3.689e+03 1.338e+03 -2.757 0.006739 \*\*

makevolvo -3.018e+03 1.557e+03 -1.939 0.054872 .

aspirationturbo 1.838e+03 4.979e+02 3.692 0.000335 \*\*\*

body.stylehardtop -2.688e+03 1.176e+03 -2.286 0.023990 \*

body.stylehatchback -2.345e+03 1.091e+03 -2.149 0.033603 \*

body.stylesedan -1.532e+03 1.125e+03 -1.361 0.176089

body.stylewagon -1.553e+03 1.288e+03 -1.206 0.230069

wheel.base 3.237e+02 8.124e+01 3.985 0.000116 \*\*\*

length -1.700e+02 4.830e+01 -3.520 0.000609 \*\*\*

width 7.702e+02 2.277e+02 3.382 0.000970 \*\*\*

height -3.157e+02 1.223e+02 -2.580 0.011067 \*

curb.weight 5.795e+00 1.287e+00 4.503 1.56e-05 \*\*\*

num.of.cylindersfive -5.438e+03 1.637e+03 -3.323 0.001179 \*\*

num.of.cylindersfour -4.874e+03 2.157e+03 -2.260 0.025630 \*

num.of.cylinderssix -4.932e+03 2.055e+03 -2.400 0.017920 \*

num.of.cylindersthree -1.435e+02 3.044e+03 -0.047 0.962491

num.of.cylinderstwelve -4.463e+03 2.889e+03 -1.545 0.125074

num.of.cylinderstwo -3.813e+03 2.699e+03 -1.413 0.160324

engine.size 2.794e+01 1.453e+01 1.923 0.056835 .

peak.rpm 1.091e+00 4.246e-01 2.569 0.011404 \*

fuel.typegas 7.457e+01 6.434e+02 0.116 0.907932

engine.locationrear NA NA NA NA

---

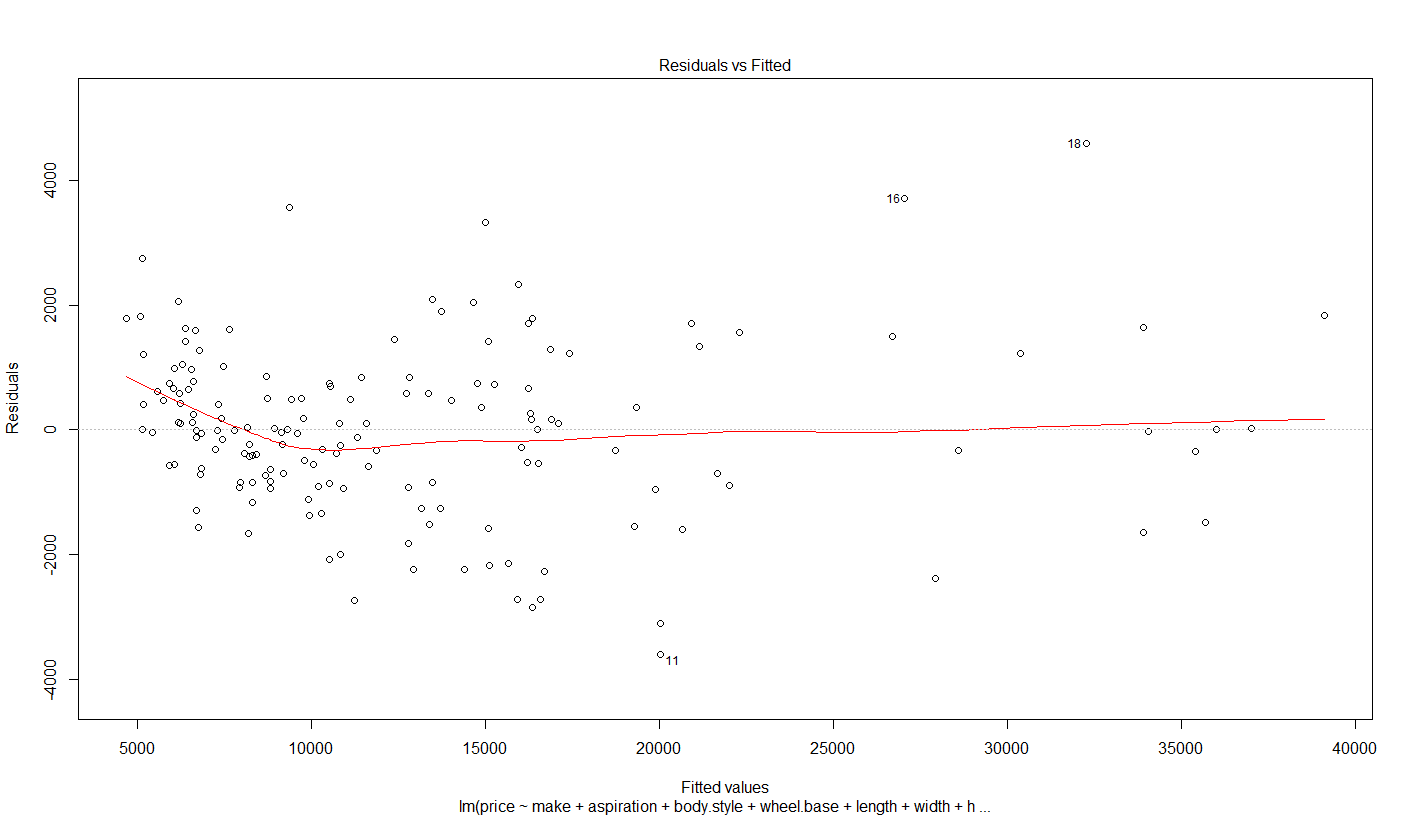
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

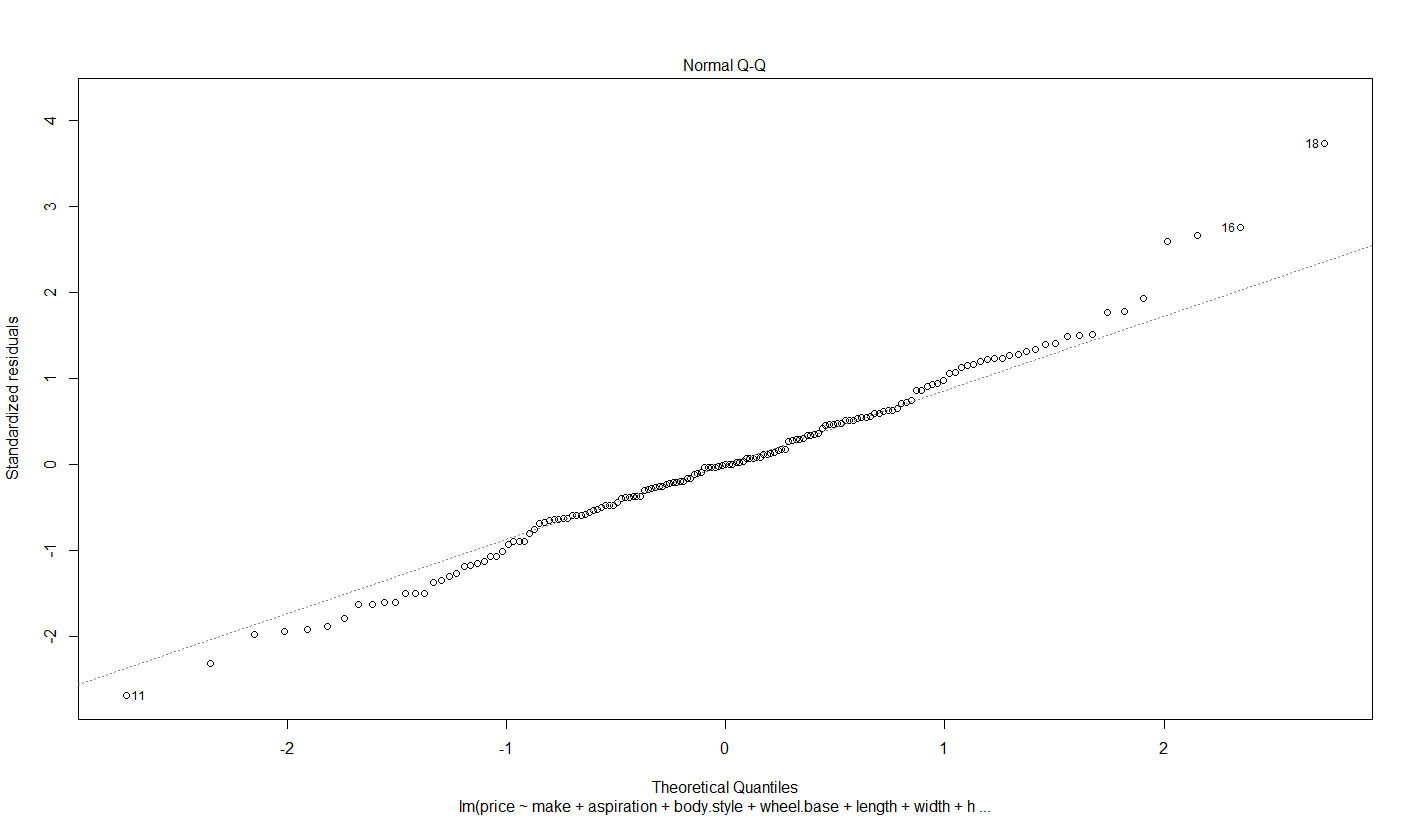
Residual standard error: 1562 on 121 degrees of freedom

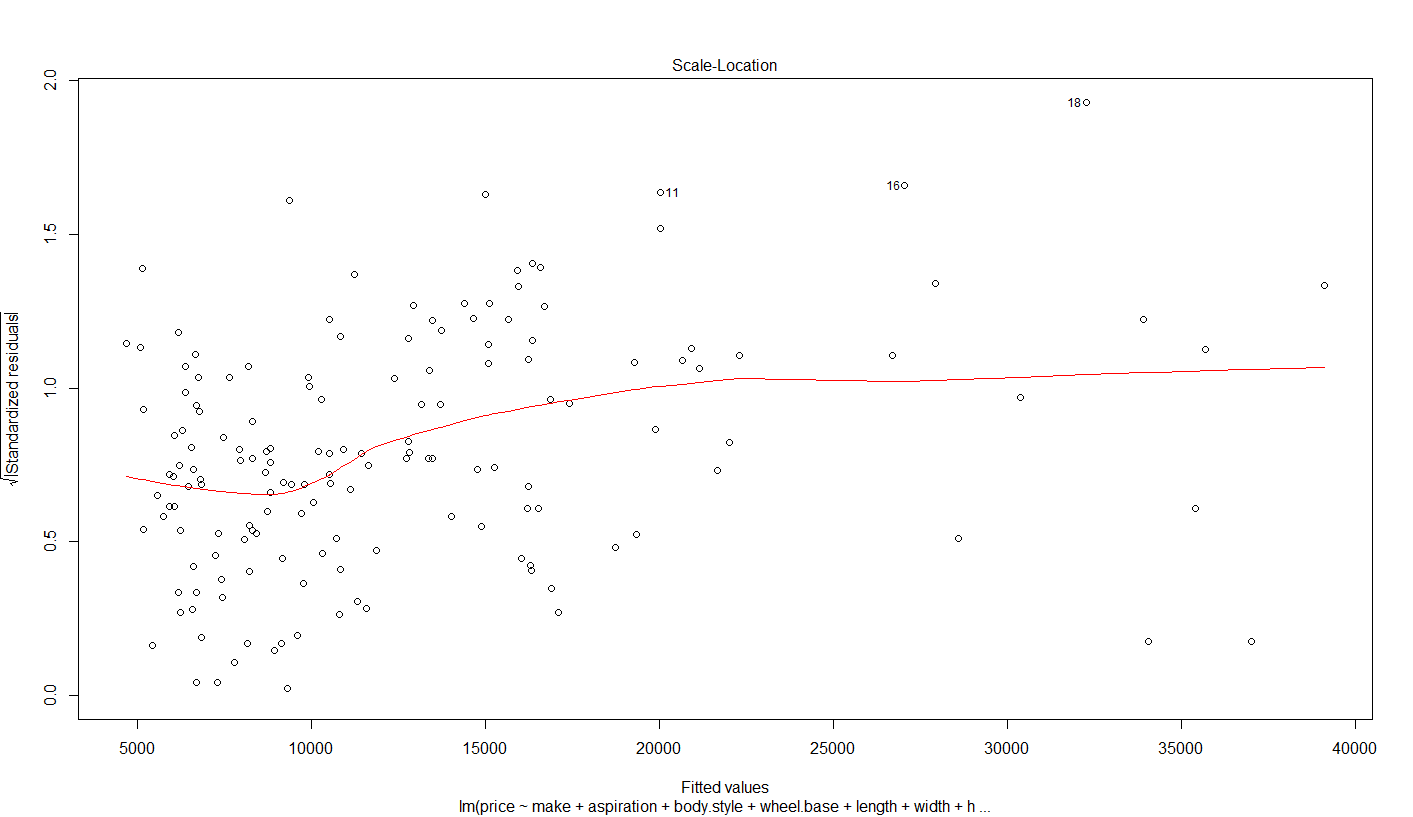
Multiple R-squared: 0.9688, Adjusted R-squared: 0.9585

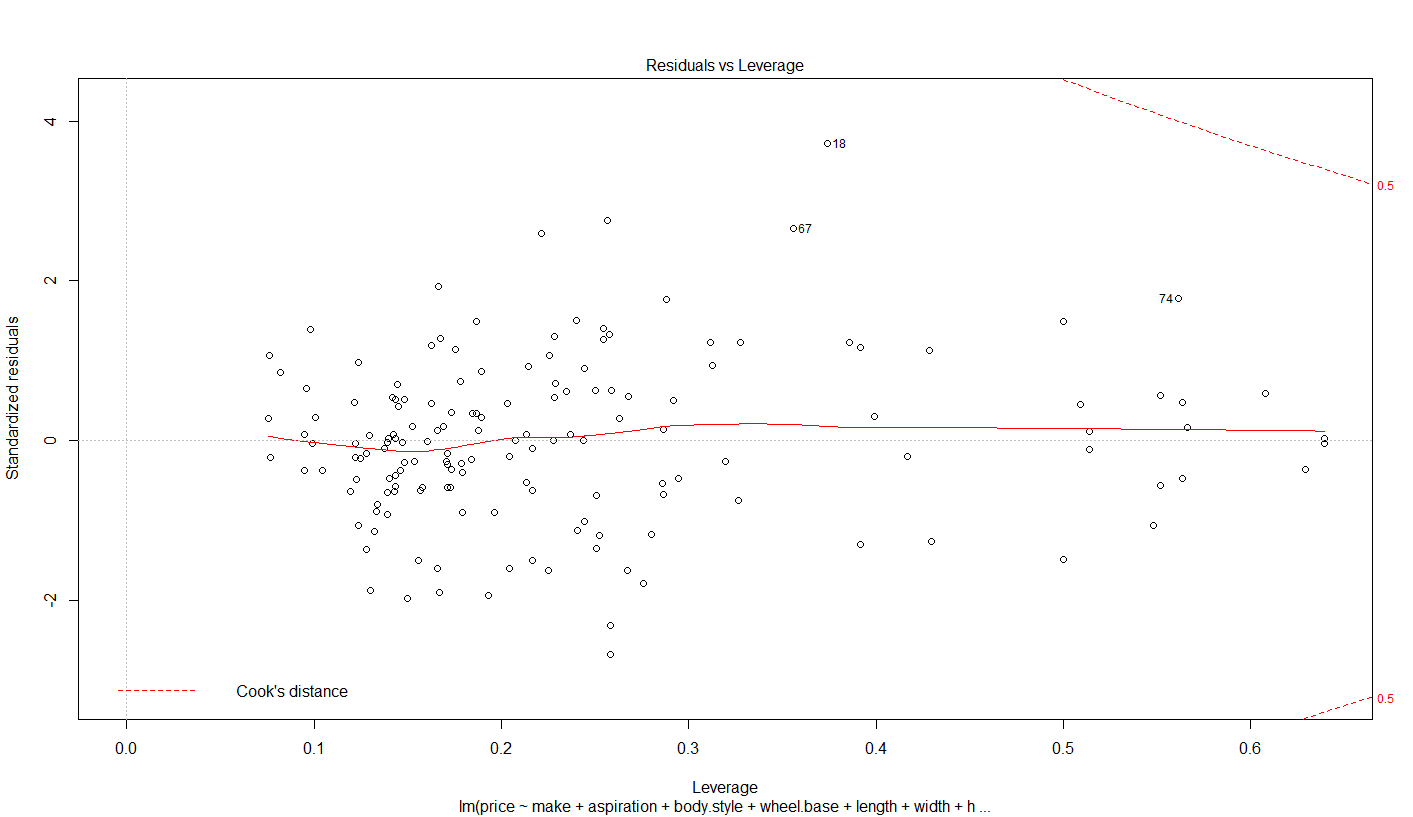
F-statistic: 93.99 on 40 and 121 DF, p-value: < 2.2e-16

plot(lm2)









#based on the obtained graphs we can see that:

# graph1 - the distribution of residuals have some changes with the fitted price values.

# graph2 - the normal Q-Q residuals are almost having straight line and appears to be close to Normally distributed

# graph3 - fitted smoothing regression line indicates that the distribution of residuals have some changes with the predicted values

# graph4 - there are outlier on the data which might affect the performance of the model. The errors and outliers in the data can have greater or lesser effect, depending on how extreme they are and their placement with respect to the other data.

#from the figure we can see there are some outlier points 11, 16, 18

Predlm <- predict(lm2, test)

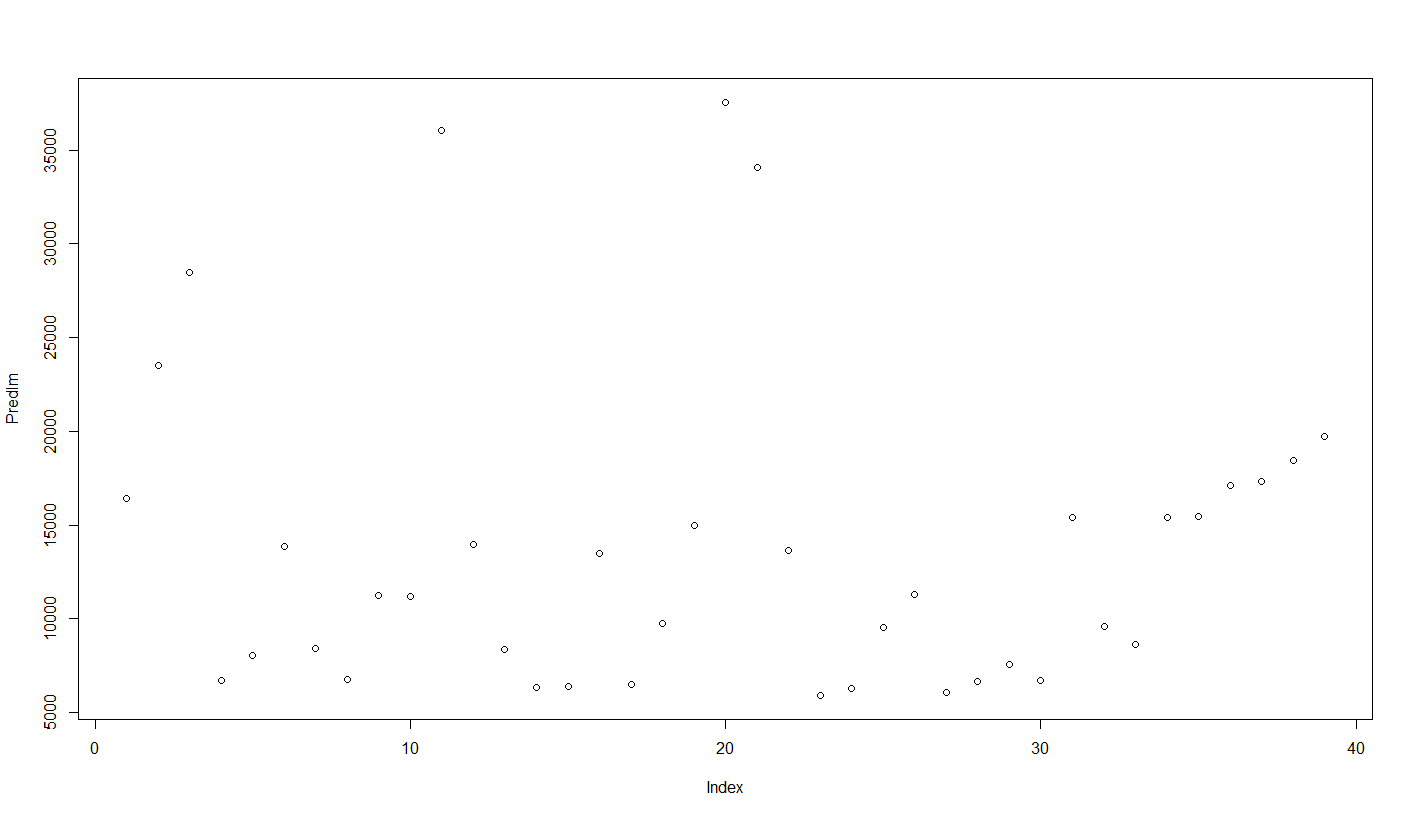
summary(Predlm)

> summary(Predlm)

Min. 1st Qu. Median Mean 3rd Qu. Max.

5919 7155 11230 13660 15920 37550

plot(Predlm)



**Improvement 2: Step-wise regression**

#improve the model again

#Cook's distance is measured in units of standard deviation. which is used to compare the differences between the mean without using data points.

#Step-wise regression is used to tackle this issue.

Start: AIC=2382.96

price ~ symboling + normalized.losses + make + fuel.type + aspiration +

num.of.doors + body.style + drive.wheels + engine.location +

wheel.base + length + width + height + curb.weight + engine.type +

num.of.cylinders + engine.size + fuel.system + bore + stroke +

compression.ratio + horsepower + peak.rpm + city.mpg + highway.mpg

Step: AIC=2382.96

price ~ symboling + normalized.losses + make + fuel.type + aspiration +

num.of.doors + body.style + drive.wheels + wheel.base + length +

width + height + curb.weight + engine.type + num.of.cylinders +

engine.size + fuel.system + bore + stroke + compression.ratio +

horsepower + peak.rpm + city.mpg + highway.mpg

Step: AIC=2382.96

price ~ symboling + normalized.losses + make + aspiration + num.of.doors +

body.style + drive.wheels + wheel.base + length + width +

height + curb.weight + engine.type + num.of.cylinders + engine.size +

fuel.system + bore + stroke + compression.ratio + horsepower +

peak.rpm + city.mpg + highway.mpg

Df Sum of Sq RSS AIC

- normalized.losses 1 3376 191207980 2381.0

- horsepower 1 14520 191219125 2381.0

- city.mpg 1 447222 191651827 2381.3

- highway.mpg 1 1747392 192951997 2382.4

- symboling 1 1971457 193176061 2382.6

<none> 191204605 2383.0

- compression.ratio 1 2699027 193903631 2383.2

- num.of.doors 1 3158761 194363366 2383.6

- body.style 4 12986803 204191408 2385.6

- peak.rpm 1 6940384 198144988 2386.7

- drive.wheels 2 11319876 202524480 2388.3

- height 1 10037593 201242197 2389.2

- engine.type 2 14897785 206102390 2391.1

- aspiration 1 13471580 204676185 2392.0

- fuel.system 6 30575076 221779680 2395.0

- stroke 1 19254447 210459052 2396.5

- width 1 21010812 212215416 2397.8

- wheel.base 1 21343694 212548299 2398.1

- bore 1 25419561 216624166 2401.2

- length 1 25880019 217084624 2401.5

- num.of.cylinders 4 40285016 231489620 2405.9

- engine.size 1 33952371 225156976 2407.4

- curb.weight 1 36805165 228009769 2409.5

- make 19 326029261 517233866 2506.2

Step: AIC=2380.97

price ~ symboling + make + aspiration + num.of.doors + body.style +

drive.wheels + wheel.base + length + width + height + curb.weight +

engine.type + num.of.cylinders + engine.size + fuel.system +

bore + stroke + compression.ratio + horsepower + peak.rpm +

city.mpg + highway.mpg

Df Sum of Sq RSS AIC

- horsepower 1 17379 191225359 2379.0

- city.mpg 1 445959 191653939 2379.3

- highway.mpg 1 1744017 192951998 2380.4

- symboling 1 2094736 193302716 2380.7

<none> 191207980 2381.0

- compression.ratio 1 2695717 193903697 2381.2

- num.of.doors 1 3251594 194459574 2381.7

+ normalized.losses 1 3376 191204605 2383.0

- body.style 4 13264651 204472631 2383.8

- peak.rpm 1 7022302 198230283 2384.8

- height 1 10592523 201800504 2387.7

- drive.wheels 2 13654246 204862227 2388.1

- engine.type 2 15028739 206236720 2389.2

- aspiration 1 13532168 204740148 2390.0

- fuel.system 6 30612768 221820748 2393.0

- stroke 1 19276450 210484430 2394.5

- width 1 21129244 212337225 2395.9

- wheel.base 1 21345977 212553958 2396.1

- bore 1 25745410 216953390 2399.4

- length 1 25916907 217124887 2399.6

- num.of.cylinders 4 40315075 231523055 2404.0

- engine.size 1 34679358 225887338 2406.0

- curb.weight 1 36855118 228063098 2407.5

- make 19 356879193 548087173 2513.6

Step: AIC=2378.98

price ~ symboling + make + aspiration + num.of.doors + body.style +

drive.wheels + wheel.base + length + width + height + curb.weight +

engine.type + num.of.cylinders + engine.size + fuel.system +

bore + stroke + compression.ratio + peak.rpm + city.mpg +

highway.mpg

Df Sum of Sq RSS AIC

- city.mpg 1 470489 191695848 2377.4

- highway.mpg 1 1790198 193015558 2378.5

- symboling 1 2080925 193306284 2378.7

<none> 191225359 2379.0

- compression.ratio 1 3045456 194270815 2379.5

- num.of.doors 1 3305674 194531033 2379.8

+ horsepower 1 17379 191207980 2381.0

+ normalized.losses 1 6235 191219125 2381.0

- body.style 4 13317101 204542461 2381.9

- peak.rpm 1 8838209 200063568 2384.3

- height 1 10641627 201866986 2385.8

- drive.wheels 2 14669433 205894793 2387.0

- engine.type 2 15038754 206264114 2387.2

- stroke 1 19457181 210682540 2392.7

- fuel.system 6 34355812 225581171 2393.8

- wheel.base 1 21356475 212581834 2394.1

- aspiration 1 21804580 213029940 2394.5

- width 1 22199614 213424974 2394.8

- length 1 26005000 217230360 2397.6

- bore 1 26017506 217242865 2397.7

- num.of.cylinders 4 42736461 233961820 2403.7

- curb.weight 1 37393220 228618580 2405.9

- engine.size 1 43221604 234446964 2410.0

- make 19 380298855 571524214 2518.3

Step: AIC=2377.38

price ~ symboling + make + aspiration + num.of.doors + body.style +

drive.wheels + wheel.base + length + width + height + curb.weight +

engine.type + num.of.cylinders + engine.size + fuel.system +

bore + stroke + compression.ratio + peak.rpm + highway.mpg

Df Sum of Sq RSS AIC

- highway.mpg 1 1956023 193651871 2377.0

- symboling 1 1971476 193667324 2377.0

<none> 191695848 2377.4

- num.of.doors 1 3498377 195194225 2378.3

- compression.ratio 1 3589335 195285183 2378.4

+ city.mpg 1 470489 191225359 2379.0

+ horsepower 1 41909 191653939 2379.3

+ normalized.losses 1 6203 191689645 2379.4

- body.style 4 12946283 204642131 2380.0

- peak.rpm 1 9255096 200950944 2383.0

- height 1 11279210 202975058 2384.6

- engine.type 2 14709711 206405559 2385.4

- drive.wheels 2 14933498 206629346 2385.5

- stroke 1 19260283 210956131 2390.9

- fuel.system 6 34828351 226524199 2392.4

- aspiration 1 21363591 213059439 2392.5

- width 1 21878096 213573944 2392.9

- wheel.base 1 22533184 214229033 2393.4

- bore 1 25632198 217328046 2395.7

- length 1 26619877 218315725 2396.4

- num.of.cylinders 4 42810706 234506554 2402.0

- curb.weight 1 40315080 232010928 2406.3

- engine.size 1 42840828 234536676 2408.1

- make 19 402443483 594139332 2522.6

Step: AIC=2377.02

price ~ symboling + make + aspiration + num.of.doors + body.style +

drive.wheels + wheel.base + length + width + height + curb.weight +

engine.type + num.of.cylinders + engine.size + fuel.system +

bore + stroke + compression.ratio + peak.rpm

Df Sum of Sq RSS AIC

- symboling 1 2021225 195673096 2376.7

<none> 193651871 2377.0

+ highway.mpg 1 1956023 191695848 2377.4

- compression.ratio 1 3095816 196747687 2377.6

- num.of.doors 1 3643787 197295658 2378.0

+ city.mpg 1 636313 193015558 2378.5

- body.style 4 11786725 205438596 2378.6

+ horsepower 1 42087 193609784 2379.0

+ normalized.losses 1 15 193651856 2379.0

- peak.rpm 1 7361939 201013810 2381.1

- height 1 11536763 205188634 2384.4

- engine.type 2 15712104 209363975 2385.7

- drive.wheels 2 15924143 209576014 2385.8

- aspiration 1 19465718 213117589 2390.5

- stroke 1 20464699 214116570 2391.3

- wheel.base 1 20766977 214418848 2391.5

- fuel.system 6 35065601 228717472 2392.0

- width 1 25506091 219157962 2395.1

- length 1 28337063 221988934 2397.2

- bore 1 31391856 225043727 2399.4

- curb.weight 1 38380390 232032261 2404.3

- num.of.cylinders 4 54411912 248063783 2409.1

- engine.size 1 47781744 241433615 2410.8

- make 19 410190692 603842563 2523.3

Step: AIC=2376.71

price ~ make + aspiration + num.of.doors + body.style + drive.wheels +

wheel.base + length + width + height + curb.weight + engine.type +

num.of.cylinders + engine.size + fuel.system + bore + stroke +

compression.ratio + peak.rpm

Df Sum of Sq RSS AIC

<none> 195673096 2376.7

+ symboling 1 2021225 193651871 2377.0

+ highway.mpg 1 2005772 193667324 2377.0

- compression.ratio 1 3566088 199239183 2377.6

- body.style 4 11187687 206860783 2377.7

+ city.mpg 1 736515 194936581 2378.1

+ normalized.losses 1 79754 195593342 2378.6

+ horsepower 1 15657 195657439 2378.7

- peak.rpm 1 7091573 202764669 2380.5

- num.of.doors 1 7126147 202799243 2380.5

- height 1 10176142 205849238 2382.9

- engine.type 2 14441430 210114526 2384.2

- drive.wheels 2 14568362 210241457 2384.3

- aspiration 1 18459382 214132478 2389.3

- stroke 1 19073169 214746265 2389.8

- fuel.system 6 34608138 230281234 2391.1

- width 1 24370711 220043807 2393.7

- wheel.base 1 25473953 221147049 2394.5

- length 1 27376197 223049293 2395.9

- bore 1 30275946 225949042 2398.0

- curb.weight 1 38493576 234166672 2403.8

- num.of.cylinders 4 52921985 248595081 2407.5

- engine.size 1 45760528 241433624 2408.8

- make 19 410195723 605868819 2521.8

The summary of the results present how the stepAIC built were its remove all the variables that have lowest value.

In the step1 model , the engine.location  has lowest AIC.

In the step 2model : the fuel.type  is also removed.

The last step: the model remove all none

We also notice that: this model suggest more feautres comparing to first lm model such as (engine.type, fuel.system, bore, stroke, compression.ratio and highway.mpg)

Let's look at the summary

summary(lm.step)

Call:

lm(formula = price ~ make + aspiration + num.of.doors + body.style +

drive.wheels + wheel.base + length + width + height + curb.weight +

engine.type + num.of.cylinders + engine.size + fuel.system +

bore + stroke + compression.ratio + peak.rpm, data = train)

Residuals:

Min 1Q Median 3Q Max

-2489.3 -640.4 0.0 785.6 2764.5

Coefficients: (3 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.248e+04 1.538e+04 -0.812 0.418787

makeaudi 3.045e+03 2.029e+03 1.500 0.136463

makebmw 8.008e+03 2.120e+03 3.777 0.000260 \*\*\*

makechevrolet -1.707e+03 1.971e+03 -0.866 0.388271

makedodge -3.586e+03 1.771e+03 -2.025 0.045386 \*

makehonda 5.860e+02 1.965e+03 0.298 0.766116

makeisuzu -2.046e+03 2.060e+03 -0.994 0.322641

makejaguar -2.989e+02 2.063e+03 -0.145 0.885043

makemazda 2.441e+02 1.627e+03 0.150 0.881027

makemercedes-benz 2.278e+03 2.122e+03 1.073 0.285521

makemercury -1.420e+03 2.535e+03 -0.560 0.576470

makemitsubishi -3.439e+03 1.741e+03 -1.975 0.050851 .

makenissan 9.322e+02 1.626e+03 0.573 0.567741

makepeugot -1.798e+04 4.204e+03 -4.277 4.10e-05 \*\*\*

makeplymouth -2.610e+03 1.696e+03 -1.539 0.126793

makeporsche 2.060e+04 2.394e+03 8.603 6.80e-14 \*\*\*

makerenault 7.390e+02 2.023e+03 0.365 0.715608

makesaab 5.182e+03 2.015e+03 2.572 0.011479 \*

makesubaru 9.230e+02 2.141e+03 0.431 0.667248

maketoyota -1.406e+03 1.501e+03 -0.936 0.351118

makevolkswagen -5.475e+00 1.638e+03 -0.003 0.997339

makevolvo 5.170e+02 2.216e+03 0.233 0.815978

aspirationturbo 1.906e+03 5.971e+02 3.192 0.001851 \*\*

num.of.doorstwo -8.619e+02 4.346e+02 -1.983 0.049880 \*

body.stylehardtop -2.736e+03 1.152e+03 -2.375 0.019306 \*

body.stylehatchback -2.431e+03 1.085e+03 -2.240 0.027107 \*

body.stylesedan -2.177e+03 1.130e+03 -1.926 0.056733 .

body.stylewagon -2.265e+03 1.253e+03 -1.809 0.073295 .

drive.wheelsfwd 2.739e+02 8.976e+02 0.305 0.760842

drive.wheelsrwd 2.208e+03 1.133e+03 1.949 0.053900 .

wheel.base 2.914e+02 7.771e+01 3.750 0.000286 \*\*\*

length -1.836e+02 4.722e+01 -3.887 0.000175 \*\*\*

width 7.920e+02 2.160e+02 3.668 0.000382 \*\*\*

height -3.170e+02 1.338e+02 -2.370 0.019565 \*

curb.weight 6.660e+00 1.445e+00 4.609 1.11e-05 \*\*\*

engine.typel 1.219e+04 4.509e+03 2.704 0.007964 \*\*

engine.typeohc -1.592e+02 9.537e+02 -0.167 0.867702

engine.typeohcf NA NA NA NA

engine.typeohcv -3.011e+03 1.067e+03 -2.822 0.005689 \*\*

engine.typerotor 1.708e+04 5.950e+03 2.872 0.004916 \*\*

num.of.cylindersfive 1.560e+03 3.174e+03 0.491 0.624082

num.of.cylindersfour 5.881e+03 3.939e+03 1.493 0.138368

num.of.cylinderssix -6.099e+02 2.435e+03 -0.250 0.802739

num.of.cylindersthree NA NA NA NA

num.of.cylinderstwelve -7.181e+03 3.678e+03 -1.952 0.053523 .

num.of.cylinderstwo NA NA NA NA

engine.size 1.396e+02 2.777e+01 5.026 2.00e-06 \*\*\*

fuel.system2bbl 3.134e+03 1.210e+03 2.590 0.010923 \*

fuel.system4bbl -2.945e+03 2.324e+03 -1.267 0.207774

fuel.systemidi 1.092e+04 6.449e+03 1.694 0.093230 .

fuel.systemmpfi 2.347e+03 1.249e+03 1.879 0.062974 .

fuel.systemspdi 1.542e+03 1.648e+03 0.935 0.351627

fuel.systemspfi 2.825e+03 2.426e+03 1.164 0.246828

bore -1.096e+04 2.681e+03 -4.088 8.40e-05 \*\*\*

stroke -2.970e+03 9.153e+02 -3.245 0.001566 \*\*

compression.ratio -6.503e+02 4.635e+02 -1.403 0.163500

peak.rpm 9.668e-01 4.887e-01 1.978 0.050428 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1346 on 108 degrees of freedom

Multiple R-squared: 0.9793, Adjusted R-squared: 0.9692

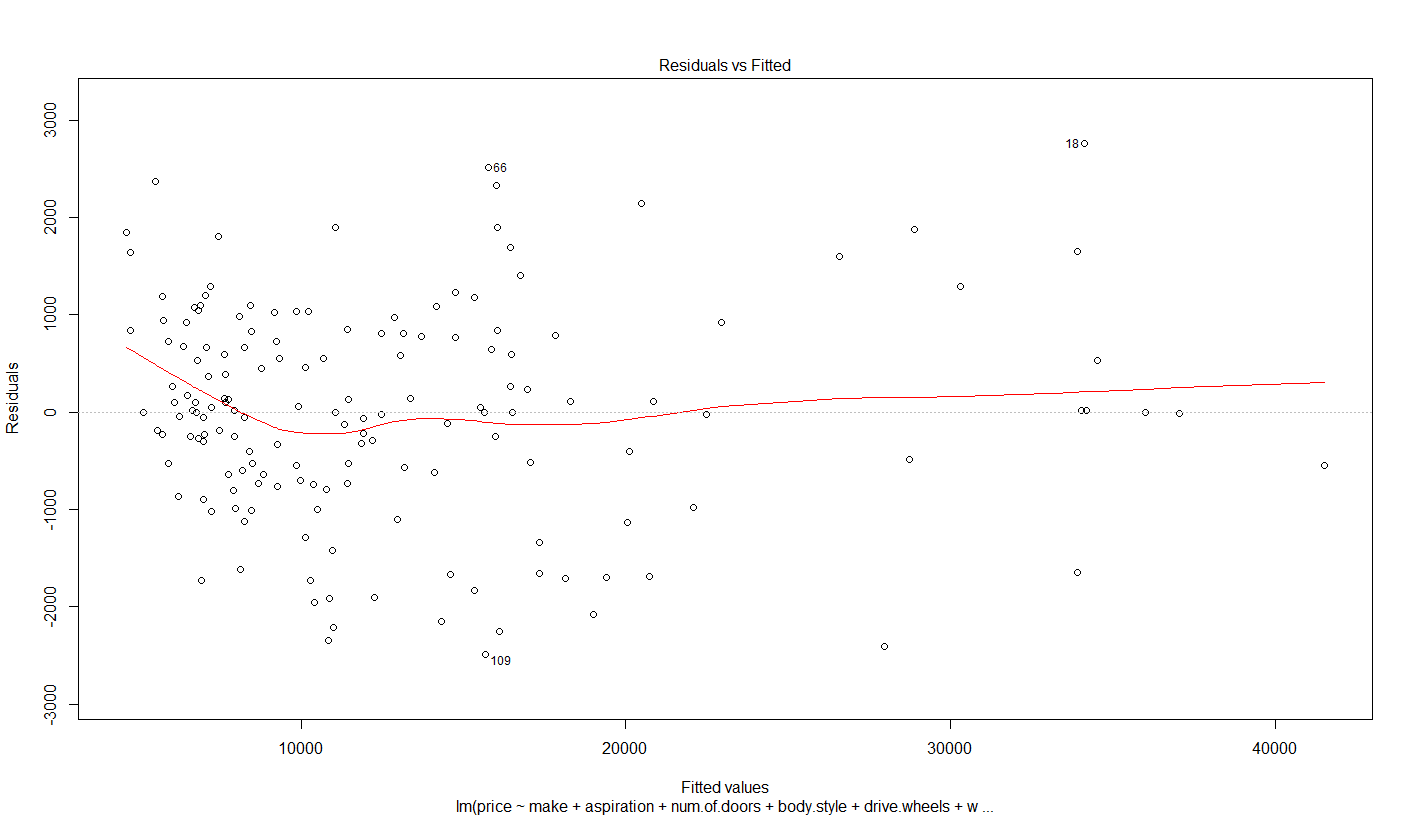
F-statistic: 96.59 on 53 and 108 DF, p-value: < 2.2e-16

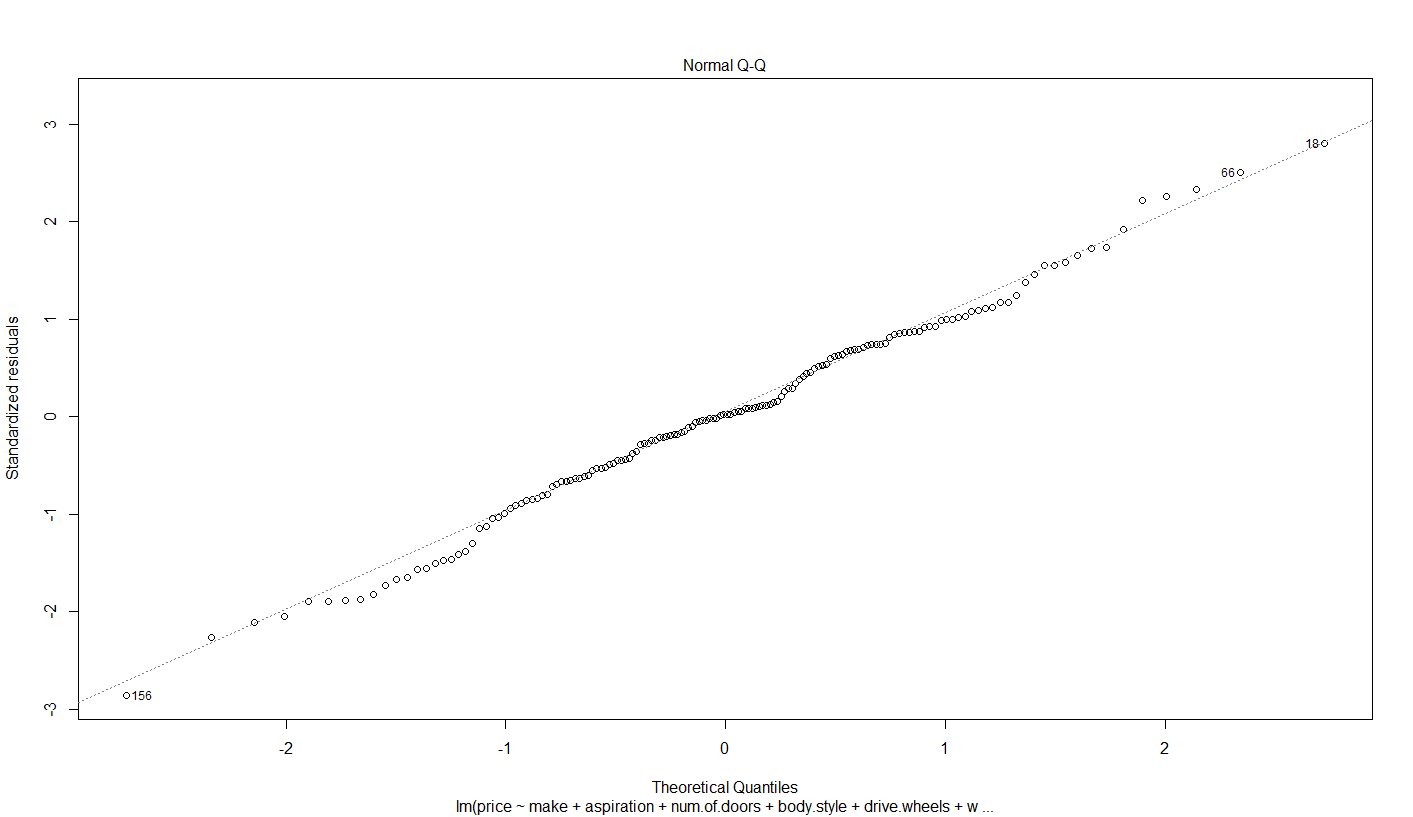
# the result analysis indicates that there are 18 variables have been used whch are :

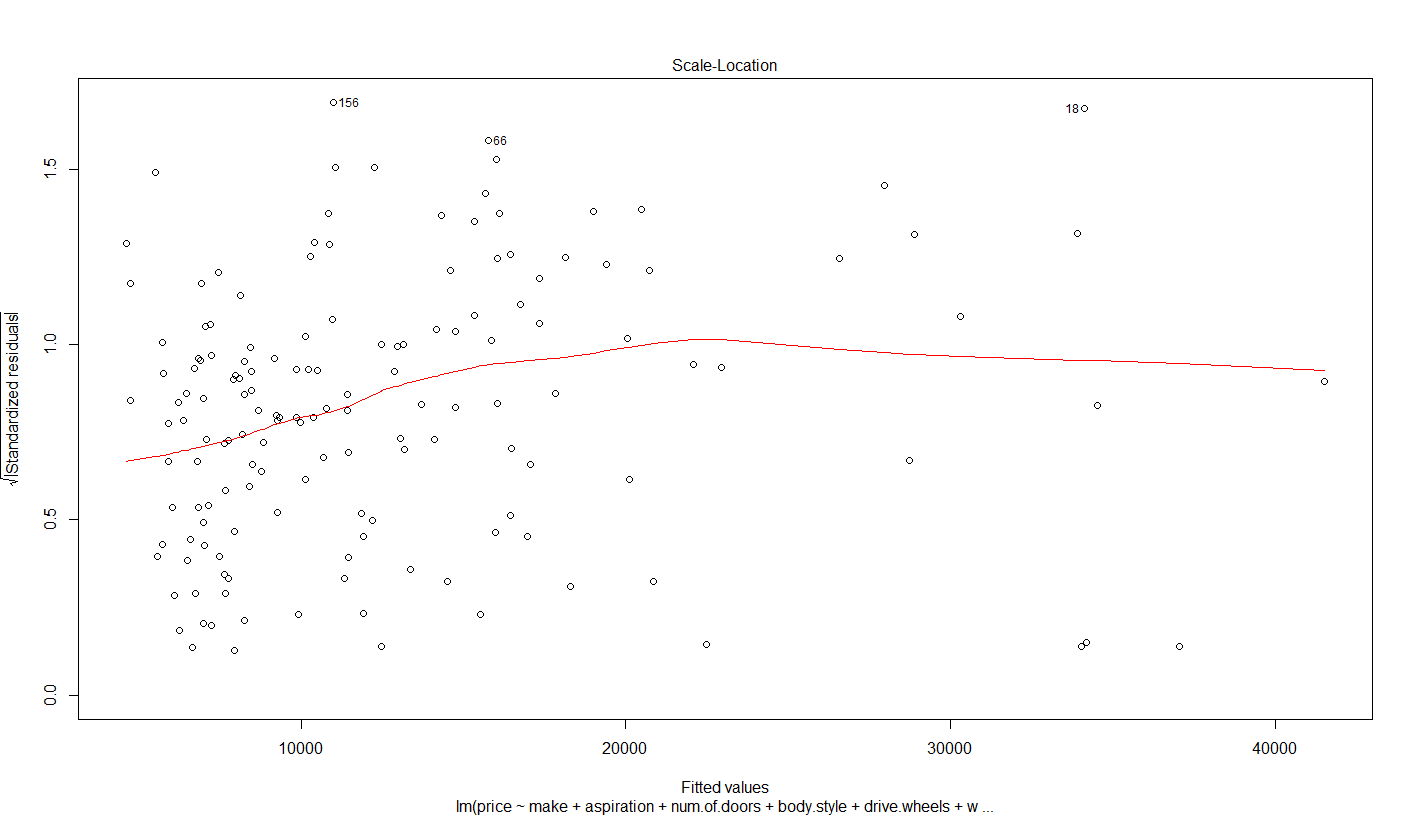
#make + aspiration + num.of.doors + body.style + drive.wheels + wheel.base + length + width + height + curb.weight + engine.type + num.of.cylinders + engine.size + fuel.system + bore + stroke + compression.ratio + peak.rpm)

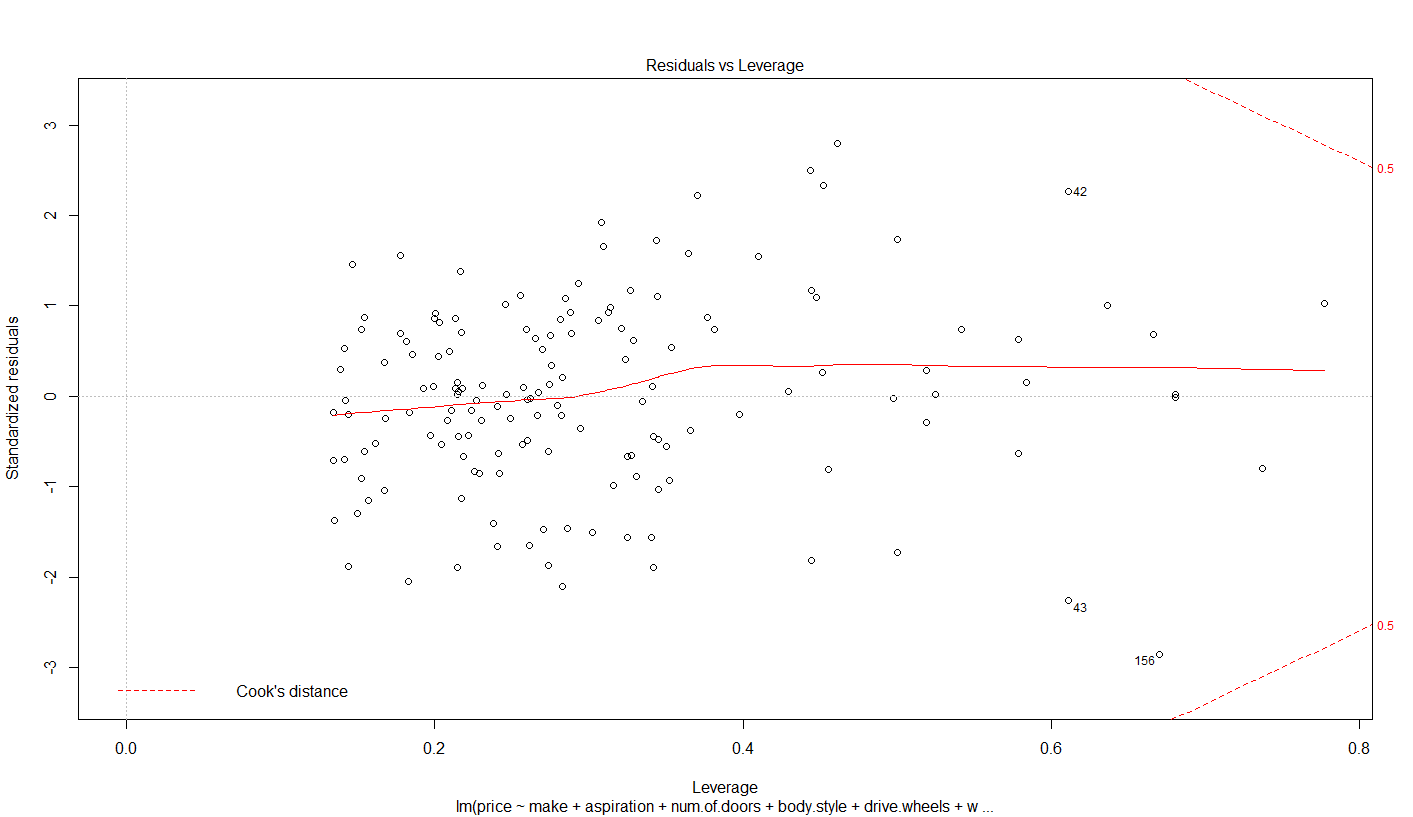
#by looking at the P-value of F-statistic we can see that the result have been improved comparing with the previous Lm model . Also the value of R2adj of this model is slightly better than that of the linear model earlier.

plot(lm.step)









Based on the obtained results, we can see that QQ plot seems to be normal. The residuals fits on the model. In the other side, the other plots have some outliers and it's not really well.

**Summary and Conclusion**

In this document, we attempt to investigate which model is the best to predict the price of automobile data. There are three models used, the first model used all the variables but the results indicated that the model have many parameters that do not fit on the model. The second model was built by selecting the features that fits the model. However, we found there are outliers on the model. The third model, we used step wise regression to select the best variables. The stepAIC model successfully selected 18 variables and enhance the performance.

The step wise is a very powerful and we need to re-build the model several times and update the model variables. One of the limitations, the variables have many variation which requires a method to handle the variances. In addition, there is a need to re-weight the variables.

* One highly recommend approach is using a regularization such as Singular Value Decomposition SVD to provide better fitting model.