Project Report

Chinki

May 20, 2017

Data Description:

Source: <https://archive.ics.uci.edu/ml/datasets/Auto+MPG> The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes. Attribute Information: 1. mpg: continuous 2. cylinders: multi-valued discrete 3. displacement: continuous 4. horsepower: continuous 5. weight: continuous 6. acceleration: continuous 7. model year: multi-valued discrete 8. origin: multi-valued discrete 9. car name: string (unique for each instance) I am taking MPG as y variable and other x variables. I will use regression technique to predict MPG. MPG is miles per galaons.

#Reading the data from the downloaded folder   
Car\_MPG=read.table("C:/Computational Statistics/3rd Quater/Regression/Project/auto-mpg.data.txt",na.strings = T)  
#Giving header to the data set  
colnames(Car\_MPG)=c("mpg","cylinders","displacement","horsepower","weight","acceleration","model","origin","car\_name")

#Getting structure of the dataset  
str(Car\_MPG)

## 'data.frame': 398 obs. of 9 variables:  
## $ mpg : num 18 15 18 16 17 15 14 14 14 15 ...  
## $ cylinders : int 8 8 8 8 8 8 8 8 8 8 ...  
## $ displacement: num 307 350 318 304 302 429 454 440 455 390 ...  
## $ horsepower : Factor w/ 94 levels "?","100.0","102.0",..: 17 35 29 29 24 42 47 46 48 40 ...  
## $ weight : num 3504 3693 3436 3433 3449 ...  
## $ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...  
## $ model : int 70 70 70 70 70 70 70 70 70 70 ...  
## $ origin : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ car\_name : Factor w/ 305 levels "amc ambassador brougham",..: 50 37 232 15 162 142 55 224 242 2 ...

# Horsepower has missing value so replacing ? into NA  
Car\_MPG$horsepower[Car\_MPG$horsepower=="?"]=NA  
  
#Checking for missing values  
table(is.na(Car\_MPG$horsepower))

##   
## FALSE TRUE   
## 392 6

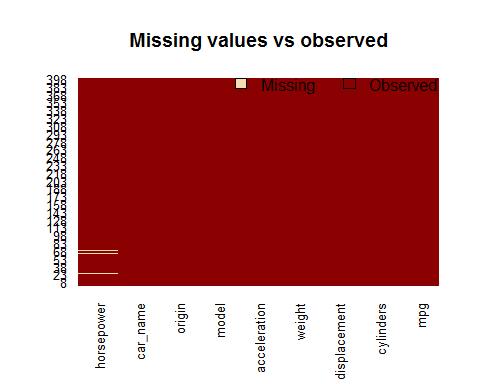
So in our data set, We have 6 missing values in the horsepower.We will deleted missing values form our dataset.

#Missing values representation  
library(Amelia)

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.4, built: 2015-12-05)  
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

missmap(Car\_MPG, main = "Missing values vs observed")



#Deleting row containing NA value  
Car\_MPG\_final=Car\_MPG[!(is.na(Car\_MPG$horsepower)) , ]

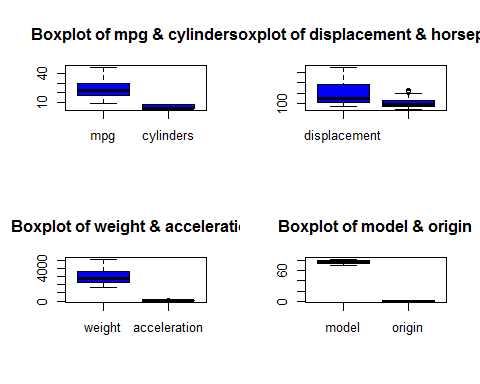
Car\_MPG\_final is a subset of car\_MPG without missing values.

#Converting factor to numeric  
Car\_MPG\_final$horsepower=as.numeric(as.character(Car\_MPG\_final$horsepower))  
str(Car\_MPG\_final$horsepower)

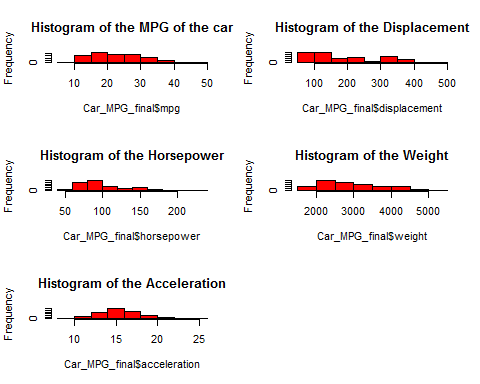
## num [1:392] 130 165 150 150 140 198 220 215 225 190 ...

Graphical Representation of the dataset

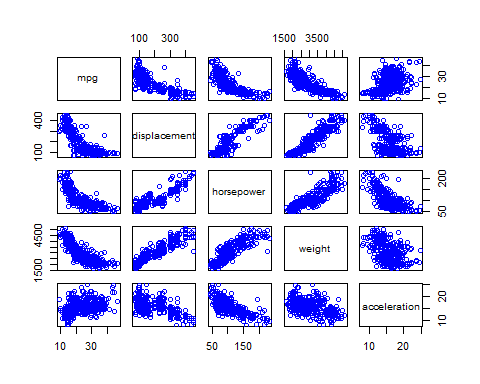
#Creating boxplot  
par(mfrow=c(2,2))  
boxplot(Car\_MPG\_final[c(1,2)],col="Blue",main="Boxplot of mpg & cylinders")  
boxplot(Car\_MPG\_final[c(3,4)],col="blue",main="Boxplot of displacement & horsepower")  
boxplot(Car\_MPG\_final[c(5,6)],col="blue",main="Boxplot of weight & acceleration")  
boxplot(Car\_MPG\_final[c(7,8)],col="blue",main="Boxplot of model & origin")

 From the boxplot,Weight, horsepower,mpg is symmetric and model and displacement is skewed.

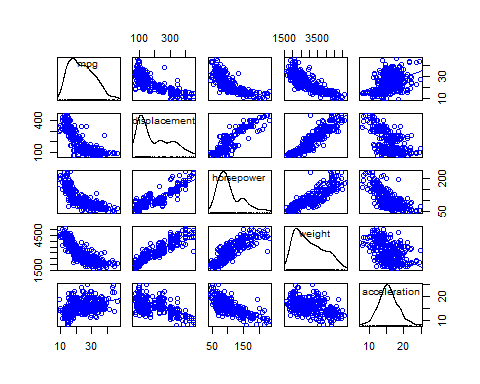
#Histograms of predicted variables   
par(mfrow=c(3,2))  
hist(Car\_MPG\_final$mpg,col="red",main = "Histogram of the MPG of the car")  
hist(Car\_MPG\_final$displacement,col="red",main = "Histogram of the Displacement")  
hist(Car\_MPG\_final$horsepower,col="red",main = "Histogram of the Horsepower")  
hist(Car\_MPG\_final$weight,col="red",main = "Histogram of the Weight")  
hist(Car\_MPG\_final$acceleration,col="red",main = "Histogram of the Acceleration")



#Scatterplot of data variable MPG with other predictors   
pairs(~mpg+displacement+horsepower+weight+acceleration,data = Car\_MPG\_final,col="blue")

 Observations from scatterplot \* Scatterplot of MPG and displacement is not in the linear trand, Inverse transformation can improve relationship between mpg and displacement \* Scatterplot of MPG and horsepower represent the two bunch of data,look like divided into 2 groups. \* Scatterplot of MPG and weight is not in the linear trand, Inverse transformation can improve relationship between mpg and weight \* Scatterplot of MPG and acceleration represent square function realtionship.

#Scatterplot matrix to observe correlation between responce & Predictor variables   
library(car)  
scatterplotMatrix(~mpg+displacement+horsepower+weight+acceleration,data = Car\_MPG\_final,ellipse=("FALSE"),smooth=F,col="blue")

 Displacement & weight are highly correlated negative correlation seem. horsepower & acceleration seems to have low correlation.

#Getting correlations  
cor(Car\_MPG\_final[,c(1,3,4,5,6)])

## mpg displacement horsepower weight acceleration  
## mpg 1.0000000 -0.8051269 -0.7784268 -0.8322442 0.4233285  
## displacement -0.8051269 1.0000000 0.8972570 0.9329944 -0.5438005  
## horsepower -0.7784268 0.8972570 1.0000000 0.8645377 -0.6891955  
## weight -0.8322442 0.9329944 0.8645377 1.0000000 -0.4168392  
## acceleration 0.4233285 -0.5438005 -0.6891955 -0.4168392 1.0000000

Displacemnet & weight is highly correlated with the mpg.

Lets start from AIC to selecte the correct model:

#Null function( with only intercept)  
null\_model=lm(mpg~1,data=Car\_MPG\_final)  
#Full model with all variables   
Full\_model=lm(mpg~weight+displacement+acceleration+horsepower+weight\*displacement+weight\*acceleration+weight\*horsepower+displacement\*acceleration+displacement\*horsepower+acceleration\*horsepower,data=Car\_MPG\_final)  
#Running step function  
step(null\_model, scope=list(lower=null\_model, upper=Full\_model),  
direction="forward")

## Start: AIC=1611.93  
## mpg ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + weight 1 16497.8 7321.2 1151.5  
## + displacement 1 15440.2 8378.8 1204.4  
## + horsepower 1 14433.1 9385.9 1248.9  
## + acceleration 1 4268.5 19550.5 1536.5  
## <none> 23819.0 1611.9  
##   
## Step: AIC=1151.49  
## mpg ~ weight  
##   
## Df Sum of Sq RSS AIC  
## + horsepower 1 327.39 6993.8 1135.6  
## + acceleration 1 168.34 7152.9 1144.4  
## + displacement 1 150.93 7170.3 1145.3  
## <none> 7321.2 1151.5  
##   
## Step: AIC=1135.56  
## mpg ~ weight + horsepower  
##   
## Df Sum of Sq RSS AIC  
## + weight:horsepower 1 1001.82 5992.0 1077.0  
## <none> 6993.8 1135.6  
## + displacement 1 13.82 6980.0 1136.8  
## + acceleration 1 0.01 6993.8 1137.6  
##   
## Step: AIC=1076.95  
## mpg ~ weight + horsepower + weight:horsepower  
##   
## Df Sum of Sq RSS AIC  
## + acceleration 1 34.064 5958.0 1076.7  
## <none> 5992.0 1077.0  
## + displacement 1 20.956 5971.1 1077.6  
##   
## Step: AIC=1076.72  
## mpg ~ weight + horsepower + acceleration + weight:horsepower  
##   
## Df Sum of Sq RSS AIC  
## + acceleration:horsepower 1 55.669 5902.3 1075.0  
## + displacement 1 34.026 5923.9 1076.5  
## <none> 5958.0 1076.7  
## + weight:acceleration 1 4.773 5953.2 1078.4  
##   
## Step: AIC=1075.04  
## mpg ~ weight + horsepower + acceleration + weight:horsepower +   
## horsepower:acceleration  
##   
## Df Sum of Sq RSS AIC  
## + weight:acceleration 1 105.14 5797.1 1070.0  
## + displacement 1 100.17 5802.1 1070.3  
## <none> 5902.3 1075.0  
##   
## Step: AIC=1069.99  
## mpg ~ weight + horsepower + acceleration + weight:horsepower +   
## horsepower:acceleration + weight:acceleration  
##   
## Df Sum of Sq RSS AIC  
## + displacement 1 96.982 5700.2 1065.4  
## <none> 5797.1 1070.0  
##   
## Step: AIC=1065.38  
## mpg ~ weight + horsepower + acceleration + displacement + weight:horsepower +   
## horsepower:acceleration + weight:acceleration  
##   
## Df Sum of Sq RSS AIC  
## + displacement:horsepower 1 59.077 5641.1 1063.3  
## <none> 5700.2 1065.4  
## + weight:displacement 1 19.683 5680.5 1066.0  
## + displacement:acceleration 1 17.867 5682.3 1066.2  
##   
## Step: AIC=1063.3  
## mpg ~ weight + horsepower + acceleration + displacement + weight:horsepower +   
## horsepower:acceleration + weight:acceleration + horsepower:displacement  
##   
## Df Sum of Sq RSS AIC  
## <none> 5641.1 1063.3  
## + weight:displacement 1 4.8842 5636.2 1065.0  
## + displacement:acceleration 1 0.0611 5641.0 1065.3

##   
## Call:  
## lm(formula = mpg ~ weight + horsepower + acceleration + displacement +   
## weight:horsepower + horsepower:acceleration + weight:acceleration +   
## horsepower:displacement, data = Car\_MPG\_final)  
##   
## Coefficients:  
## (Intercept) weight horsepower   
## 7.191e+01 -9.535e-03 -1.998e-01   
## acceleration displacement weight:horsepower   
## -6.977e-01 -5.591e-02 2.834e-05   
## horsepower:acceleration weight:acceleration horsepower:displacement   
## -5.666e-03 3.036e-04 3.317e-04

Low AIC model is mpg ~ weight + horsepower + acceleration + displacement + weight:horsepower + horsepower:acceleration + weight:acceleration + horsepower:displacement.

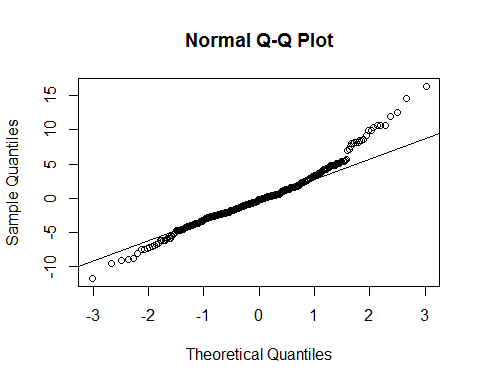
#Running multiple linear regression  
lm\_model1=lm(mpg ~ weight + horsepower + acceleration + displacement +   
 weight:horsepower + horsepower:acceleration + weight:acceleration +   
 horsepower:displacement ,data=Car\_MPG\_final)  
summary(lm\_model1)

##   
## Call:  
## lm(formula = mpg ~ weight + horsepower + acceleration + displacement +   
## weight:horsepower + horsepower:acceleration + weight:acceleration +   
## horsepower:displacement, data = Car\_MPG\_final)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.6700 -2.2650 -0.2052 1.7678 16.3084   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.191e+01 8.403e+00 8.558 2.8e-16 \*\*\*  
## weight -9.535e-03 4.537e-03 -2.102 0.03621 \*   
## horsepower -1.998e-01 6.418e-02 -3.114 0.00199 \*\*   
## acceleration -6.977e-01 3.958e-01 -1.763 0.07873 .   
## displacement -5.591e-02 2.044e-02 -2.736 0.00651 \*\*   
## weight:horsepower 2.834e-05 1.744e-05 1.625 0.10492   
## horsepower:acceleration -5.666e-03 4.304e-03 -1.317 0.18876   
## weight:acceleration 3.036e-04 1.881e-04 1.614 0.10733   
## horsepower:displacement 3.317e-04 1.656e-04 2.003 0.04591 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.838 on 383 degrees of freedom  
## Multiple R-squared: 0.7632, Adjusted R-squared: 0.7582   
## F-statistic: 154.3 on 8 and 383 DF, p-value: < 2.2e-16

Intercept,weight, horsepower,dispalcment and interaction between horsepower & displacement are significant. Rsquare is 75.82%.

Diagnostics of the model : Normalirty:

#Residuals of the model   
lm\_r1=residuals(lm\_model1)  
#Fitted value of the model  
lm\_f1=fitted.values(lm\_model1)  
#QQ plots   
qqnorm(lm\_r1)  
qqline(lm\_r1)

 More points are divered from the line so lets look formal test for normality

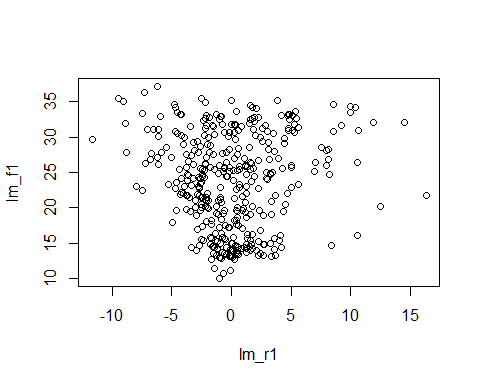
#Shapiro test for normality  
shapiro.test(lm\_r1)

##   
## Shapiro-Wilk normality test  
##   
## data: lm\_r1  
## W = 0.96424, p-value = 3.46e-08

As w is greater than 95%, We can assume normality is true for the given data set.

Constant Variance :

#Residuals Vrs fitted plot  
plot(lm\_r1,lm\_f1)

 Residuals vs fitted plot look like in a U shape. Lets do formal test

#Formal test for constant variance   
library(car)  
ncvTest(lm\_model1)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 58.69188 Df = 1 p = 1.844011e-14

As p value is less than 0.05,Reject null hypothesis and conclude that errors did not have constant variance.

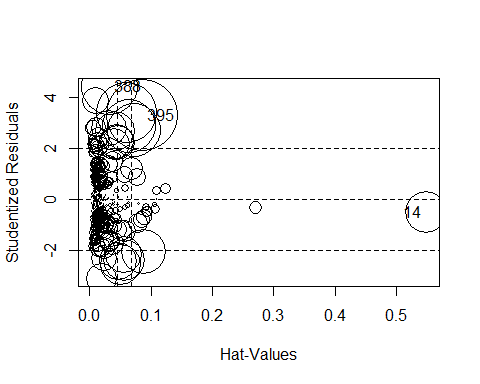
Outliers:

#Formal test for outliers  
outlierTest(lm\_model1)

## rstudent unadjusted p-value Bonferonni p  
## 388 4.424206 1.2642e-05 0.0049558

As p value is less than 0.004, outlier is significant.It is safe to remove this outlier from the data point.

#Influence plot of the model  
influencePlot(lm\_model1)



## StudRes Hat CookD  
## 14 -0.5294243 0.54840544 0.03789092  
## 388 4.4242059 0.03271793 0.07016078  
## 395 3.2872428 0.08682752 0.11131312

We are getting 3 outliers,so I am planning to delete this outlier.

#Deleting 3 outliers  
Car\_MPG\_final=Car\_MPG\_final[-c(14,388,395),]

As constant variance assumption is not valid for my dataset. Lets start with the transformation From the scatter plot shape, I am deciding my transformation.

# inverse transformation for weight & displacment, square for acceleration & sqaure root for horsepower  
Car\_MPG\_final=cbind(Car\_MPG\_final,weight1=1/Car\_MPG\_final$weight)  
Car\_MPG\_final=cbind(Car\_MPG\_final,acceleration1=Car\_MPG\_final$acceleration^2)  
Car\_MPG\_final=cbind(Car\_MPG\_final,displacement1=1/Car\_MPG\_final$displacement)   
Car\_MPG\_final=cbind(Car\_MPG\_final,horsepower1=sqrt(Car\_MPG\_final$horsepower))

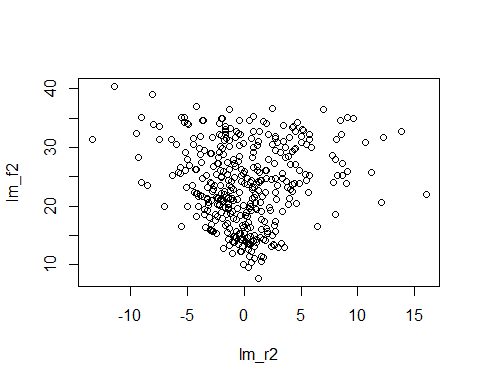
Lets us run model

#Regression model   
lm\_model2=lm(mpg ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
 weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
 horsepower1:displacement1,data=Car\_MPG\_final)  
summary(lm\_model2)

##   
## Call:  
## lm(formula = mpg ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
## weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
## horsepower1:displacement1, data = Car\_MPG\_final)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13.4088 -2.1737 -0.2889 1.8872 16.0240   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.202e+01 1.222e+01 0.984 0.32591   
## weight1 -2.295e+04 4.512e+04 -0.509 0.61133   
## horsepower1 -1.053e+00 9.193e-01 -1.146 0.25262   
## acceleration1 9.639e-02 3.211e-02 3.002 0.00286 \*\*  
## displacement1 4.589e+03 1.560e+03 2.942 0.00346 \*\*  
## weight1:horsepower1 8.418e+03 4.421e+03 1.904 0.05764 .   
## horsepower1:acceleration1 -5.914e-03 2.199e-03 -2.689 0.00747 \*\*  
## weight1:acceleration1 -1.335e+02 4.170e+01 -3.201 0.00149 \*\*  
## horsepower1:displacement1 -4.343e+02 1.610e+02 -2.697 0.00731 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.873 on 381 degrees of freedom  
## Multiple R-squared: 0.759, Adjusted R-squared: 0.754   
## F-statistic: 150 on 8 and 381 DF, p-value: < 2.2e-16

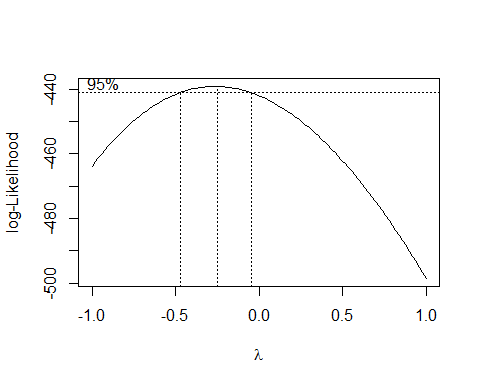
R square is almost same in this model as well. Diagnostics for this model:

#Residuals of this model  
lm\_r2=residuals(lm\_model2)  
#Fitted value of this model   
lm\_f2=fitted.values(lm\_model2)  
#Redisdual vs fitted plot  
plot(lm\_r2,lm\_f2)

 It is more concentrated towards o. Error variance is not costant. As if observe the shape its cone so lets try transformation of y variable.

I will decide transformation from the Boxcox.

#Boxcox transformation  
library(MASS)  
boxcox(Car\_MPG\_final$mpg~Car\_MPG\_final$weight1+Car\_MPG\_final$displacement+Car\_MPG\_final$acceleration+Car\_MPG\_final$weight1\*Car\_MPG\_final$displacement+Car\_MPG\_final$weight1\*Car\_MPG\_final$acceleration+Car\_MPG\_final$displacement\*Car\_MPG\_final$acceleration, lambda = seq(-1, 1, length = 20))

 From the graph it is somewhat between -0.5 and 0. So i will try both log and 1/sqrt(y)

#Transforming y variable  
Car\_MPG\_final=cbind(Car\_MPG\_final,mpg1=log(Car\_MPG\_final$mpg))  
Car\_MPG\_final=cbind(Car\_MPG\_final,mpg2=1/sqrt(Car\_MPG\_final$mpg))

Running Regression model

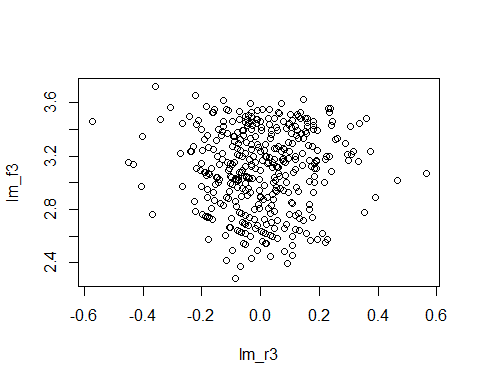
#Regression model with y transfromation as log   
lm\_model3=lm(mpg1 ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
 weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
 horsepower1:displacement1,data=Car\_MPG\_final)  
summary(lm\_model3)

##   
## Call:  
## lm(formula = mpg1 ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
## weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
## horsepower1:displacement1, data = Car\_MPG\_final)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.57366 -0.09236 -0.00851 0.09668 0.56365   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.478e+00 4.750e-01 7.322 1.46e-12 \*\*\*  
## weight1 -2.867e+03 1.754e+03 -1.634 0.103003   
## horsepower1 -1.252e-01 3.574e-02 -3.504 0.000512 \*\*\*  
## acceleration1 3.820e-03 1.248e-03 3.060 0.002368 \*\*   
## displacement1 1.617e+02 6.064e+01 2.667 0.007970 \*\*   
## weight1:horsepower1 5.437e+02 1.719e+02 3.163 0.001684 \*\*   
## horsepower1:acceleration1 -2.636e-04 8.549e-05 -3.084 0.002193 \*\*   
## weight1:acceleration1 -4.652e+00 1.621e+00 -2.869 0.004344 \*\*   
## horsepower1:displacement1 -1.517e+01 6.260e+00 -2.423 0.015841 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1506 on 381 degrees of freedom  
## Multiple R-squared: 0.8079, Adjusted R-squared: 0.8038   
## F-statistic: 200.3 on 8 and 381 DF, p-value: < 2.2e-16

R square is 80.38%.

Diagnostics Constant variance:

#Residuals of the model  
lm\_r3=residuals(lm\_model3)  
#Fitted value of the model  
lm\_f3=fitted.values(lm\_model3)  
#Plot of residuals and fitted value   
plot(lm\_r3,lm\_f3)

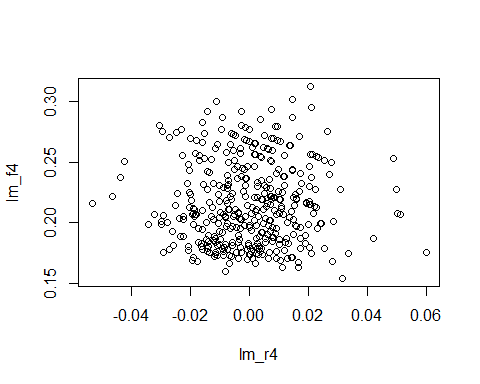
 still we are not good for the constant variance assumption. Lets try another transformation

#Regression model with y transfromation as i/sqrt(y)   
lm\_model4=lm(mpg2 ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
 weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
 horsepower1:displacement1,data=Car\_MPG\_final)  
summary(lm\_model4)

##   
## Call:  
## lm(formula = mpg2 ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
## weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
## horsepower1:displacement1, data = Car\_MPG\_final)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.053316 -0.010159 0.000082 0.009552 0.059980   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.170e-01 4.990e-02 2.344 0.019598 \*   
## weight1 4.387e+02 1.843e+02 2.380 0.017782 \*   
## horsepower1 1.843e-02 3.755e-03 4.908 1.37e-06 \*\*\*  
## acceleration1 -3.927e-04 1.311e-04 -2.995 0.002928 \*\*   
## displacement1 -1.564e+01 6.370e+00 -2.455 0.014544 \*   
## weight1:horsepower1 -7.031e+01 1.805e+01 -3.894 0.000116 \*\*\*  
## horsepower1:acceleration1 2.979e-05 8.981e-06 3.317 0.000999 \*\*\*  
## weight1:acceleration1 4.215e-01 1.703e-01 2.475 0.013754 \*   
## horsepower1:displacement1 1.466e+00 6.576e-01 2.230 0.026337 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01582 on 381 degrees of freedom  
## Multiple R-squared: 0.8214, Adjusted R-squared: 0.8177   
## F-statistic: 219.1 on 8 and 381 DF, p-value: < 2.2e-16

This model looks better than the previous model. Diagnostics: Normality:

#Residual of the model  
lm\_r4=residuals(lm\_model4)  
#Fitted value of the model  
lm\_f4=fitted.values(lm\_model4)  
#Residuals & fitted value plot  
plot(lm\_r4,lm\_f4)

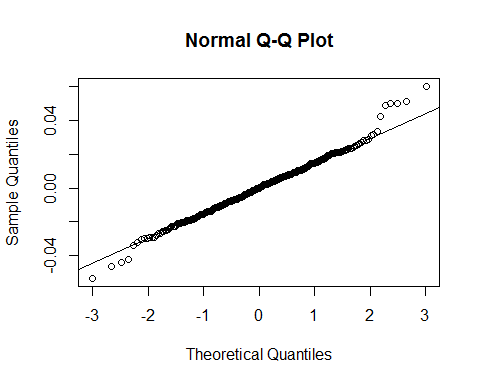
 plot is scattered enoght to conclude constant variance.

# Formal test for constant variance  
library(car)  
ncvTest(lm\_model4)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 0.02372449 Df = 1 p = 0.877588

As p-value is greter than 0.05, we are fail to reject null hypothesis and conclude that error has constant variance. Normality:

#Normality plot  
qqnorm(lm\_r4)  
qqline(lm\_r4)

 Normality seems ok. Lets try formal test for normality.

#Shapiro test  
shapiro.test(lm\_r4)

##   
## Shapiro-Wilk normality test  
##   
## data: lm\_r4  
## W = 0.9886, p-value = 0.003849

As w is greater than 95%, we will assume normality is satisfied. We have alredy worked with the outliers so we are good with the diagnostics. Model interpretation:

Lets start check model preformance: Lets create random sample

set.seed(123)  
train\_sample <- sample(398, 300)

Creating training and test model:

#Creating training and test sample  
Car\_MPG\_train=Car\_MPG\_final[train\_sample,]  
Car\_MPG\_test=Car\_MPG\_final[-train\_sample,]

Now we are going to train model on the training data.

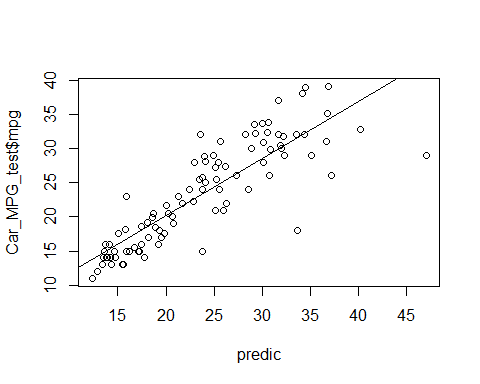
#Regression model on training data  
lm\_model5=lm(mpg2 ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
 weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
 horsepower1:displacement1,data=Car\_MPG\_train)  
summary(lm\_model5)

##   
## Call:  
## lm(formula = mpg2 ~ weight1 + horsepower1 + acceleration1 + displacement1 +   
## weight1:horsepower1 + horsepower1:acceleration1 + weight1:acceleration1 +   
## horsepower1:displacement1, data = Car\_MPG\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.054618 -0.010329 0.000633 0.009488 0.051582   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.269e-01 5.729e-02 2.216 0.027490 \*   
## weight1 5.407e+02 2.233e+02 2.421 0.016106 \*   
## horsepower1 1.820e-02 4.351e-03 4.182 3.85e-05 \*\*\*  
## acceleration1 -4.958e-04 1.512e-04 -3.279 0.001172 \*\*   
## displacement1 -2.190e+01 8.070e+00 -2.713 0.007067 \*\*   
## weight1:horsepower1 -8.256e+01 2.179e+01 -3.788 0.000185 \*\*\*  
## horsepower1:acceleration1 3.608e-05 1.040e-05 3.469 0.000604 \*\*\*  
## weight1:acceleration1 5.557e-01 1.945e-01 2.857 0.004593 \*\*   
## horsepower1:displacement1 2.067e+00 8.226e-01 2.513 0.012540 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01588 on 285 degrees of freedom  
## (6 observations deleted due to missingness)  
## Multiple R-squared: 0.8259, Adjusted R-squared: 0.821   
## F-statistic: 169 on 8 and 285 DF, p-value: < 2.2e-16

#Prediction based on our model  
prediction=predict(lm\_model5,Car\_MPG\_test)  
#As we transformed our y variable so getting accurate y variable   
predic=1/(prediction\*prediction)  
predic

## 2 4 5 6 12 13 15 16   
## 16.14978 17.42708 18.19772 13.64956 17.74561 17.11456 25.56711 21.68310   
## 20 23 24 31 36 37 39 40   
## 30.74163 24.07593 27.27426 25.46509 19.56567 20.76855 14.25849 13.54129   
## 52 62 65 66 68 75 78 84   
## 32.03620 25.94389 14.63038 14.73928 12.31100 14.27931 26.25017 30.07816   
## 88 90 99 102 107 110 112 118   
## 15.49044 15.92954 19.19307 21.31028 12.82956 25.12301 33.65620 47.04437   
## 119 124 126 138 139 143 146 152   
## 28.56476 20.66050 20.68570 13.41981 13.93422 37.23622 34.34683 36.68174   
## 155 159 160 163 164 170 172 176   
## 23.77388 13.65700 13.78072 17.19537 19.28251 20.64388 23.74451 35.11174   
## 188 190 194 203 216 220 227 229   
## 15.04754 16.71541 22.41854 19.87732 15.53704 25.25971 18.66294 18.94894   
## 230 234 237 238 239 247 250 253   
## 14.12431 32.31480 23.43487 31.84091 29.23072 40.24089 18.61618 18.04720   
## 254 261 270 277 283 289 297 299   
## 20.20742 17.41836 30.14631 20.08423 22.83190 15.80155 26.21151 15.87720   
## 304 307 312 314 322 329 332 339   
## 32.25042 23.97334 31.63628 22.99602 29.25779 28.84436 30.67707 25.15224   
## 341 344 345 346 347 348 353 356   
## 23.73985 36.86266 34.49842 36.81288 30.56873 31.71111 30.86954 30.05340   
## 360 371 372 374 384 385 391 396   
## 24.06575 25.63111 24.91110 22.48566 34.22558 33.52803 23.57866 28.27038

# Our prediction and test variable graph  
plot(Car\_MPG\_test$mpg~predic)  
abline(lm(Car\_MPG\_test$mpg~predic))

 Our prediction and test points are in a stright line, only we have some points here and there so I will conclude my prediction model is accurate.

MAE <- function(actual, predicted) {  
 mean(abs(actual - predicted))   
}  
MAE(Car\_MPG\_test$mpg,predic)

## [1] 2.748808