Regression\_Assignment\_2

Karthick~Rahul

2 March 2018

Problem Statement

The data file auto.data contains information on

1. miles per gallon (mpg)
2. number of engine cylinders
3. engine displacement in cubic inches
4. engine horsepower
5. body weight in lbs.
6. acceleration
7. last two digits of the year of the model
8. country of origin (coded as 1 for American, 2 for European and 3 for Japanese)
9. and the name or make of 397 passenger cars

The dependent variable of interest i.e. the model sought, is that for a car’s mileage (mpg), relating it to the remaining independent variables. Among these independent variables, we have no interest in year and name of the cars.

Visualization

It might be impossible to graphically display the simultaneous effects of all the independent variables on the dependent variable in a single image. With the aid of a minimum number of graphical images, displaying the simultaneous effects of as many of the independent variables as possible, briefly explain how the mileage of a car (mpg) is related to which of the independent variables.

library(dplyr)  
library(car)  
library(corrplot)  
library(MASS)  
library(lmtest)

#Dowloading the entire dataset  
df <- read.table("auto.data",header = TRUE)  
df$year <- NULL  
df$name <- NULL  
str(df)

## 'data.frame': 397 obs. of 7 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 : num 130 165 150 150 140 198 220 215 225 190 ...  
## $ weight : num 3504 3693 3436 3433 3449 ...  
## $ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...  
## $ origin : int 1 1 1 1 1 1 1 1 1 1 ...

head(df)

## mpg cylinders displacement horsepower weight acceleration origin  
## 1 18 8 307 130 3504 12.0 1  
## 2 15 8 350 165 3693 11.5 1  
## 3 18 8 318 150 3436 11.0 1  
## 4 16 8 304 150 3433 12.0 1  
## 5 17 8 302 140 3449 10.5 1  
## 6 15 8 429 198 4341 10.0 1

print("Total variables in the model")

## [1] "Total variables in the model"

colnames(df)

## [1] "mpg" "cylinders" "displacement" "horsepower"   
## [5] "weight" "acceleration" "origin"

print("Only two variables of the seven seem to take levels : number of cylinders and Origin")

## [1] "Only two variables of the seven seem to take levels : number of cylinders and Origin"

print("summary on the cylinders ")

## [1] "summary on the cylinders "

table(df$cylinder)

##   
## 3 4 5 6 8   
## 4 203 3 84 103

print("summary of Origin of the cars ")

## [1] "summary of Origin of the cars "

table(df$origin)

##   
## 1 2 3   
## 248 70 79

We observed that there were 5 rows in the data that had NA values for horsepower. Since this is less 2% of the total rows of data, we can safely omit these values from the dataset.

df <- na.omit(df)  
str(df)

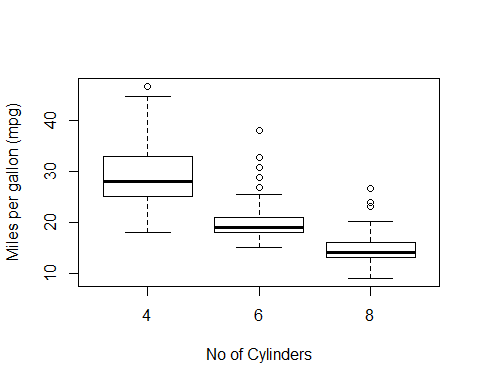
## 'data.frame': 392 obs. of 7 variables:  
## $ mpg : num 18 15 18 16 17 15 14 14 14 15 ...  
## $ cylinders : int 8 8 8 8 8 8 8 8 8 8 ...  
## $ displacement: num 307 350 318 304 302 429 454 440 455 390 ...  
## $ horsepower : num 130 165 150 150 140 198 220 215 225 190 ...  
## $ weight : num 3504 3693 3436 3433 3449 ...  
## $ acceleration: num 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...  
## $ origin : int 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "na.action")=Class 'omit' Named int [1:5] 33 127 331 337 355  
## .. ..- attr(\*, "names")= chr [1:5] "33" "127" "331" "337" ...

Since 3- and 5-Cylinder Vehicles have only 4 and 3 rows of data, we can club them together into the 4-Cylinder category of vehicles.

#considering the cylinder at only three levels  
for ( i in 1:nrow(df)){   
 if (df$cylinders[i] == 3){  
 df$cylinders[i] = 4  
 }  
 if (df$cylinders[i] ==5){  
 df$cylinders[i] = 4  
 }  
}  
#Final table cylinders summary before encoding  
table(df$cylinder)

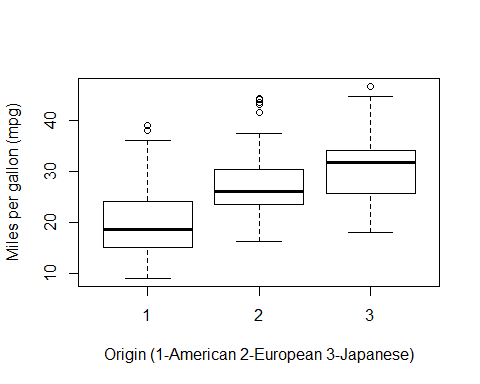
##   
## 4 6 8   
## 206 83 103

boxplot(df$mpg~df$cylinders, xlab = "No of Cylinders", ylab = "Miles per gallon (mpg)")



We know that increasing the number of Cylinders in a Car increases its pulling capacity and fuel consumption. This means that increasing the number of Cylinders in a Car decreases its mileage. The above boxplot confirms our suspicion.

boxplot(df$mpg~df$origin, xlab = "Origin (1-American 2-European 3-Japanese)", ylab = "Miles per gallon (mpg)")

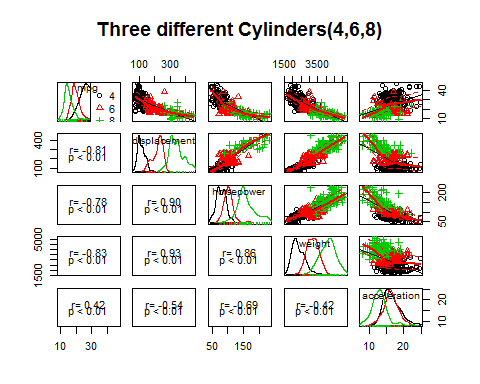
 Conventional wisdom in Cars would tell us that the Japanese car makers usually focus on mileage of the car (higher fuel efficiency) and their cars are known for their efficient performance. SImilarly, European Manufacturers focus on riding comfort and at the same time mileage, without trying to sacrifice one for the other. US Car manufacturers consider cars as luxury goods and the focus is on comfort over mileage.

As can be seen from the above boxplot, we can conclude that the data is in accordance with our assumption regarding miles per gallon (mpg) for cars of 3 different origins.

To observe the relationship between the other quantitative varialbes and mpg, we can check the scatterplot matrix below and observe the values, separately colored for Cylinder Types and Country of Origin.

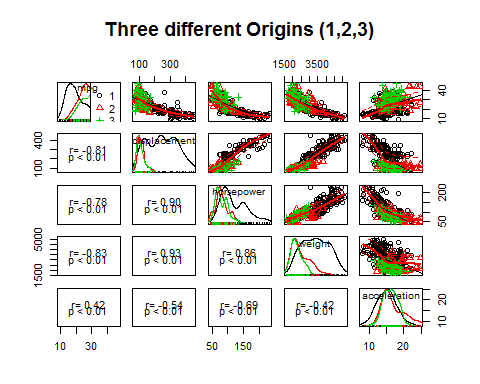
panel.cor <- function(x, y, digits = 2, cex.cor, ...)  
{  
 usr <- par("usr"); on.exit(par(usr))  
 par(usr = c(0, 1, 0, 1))  
 # correlation coefficient  
 r <- cor(x, y)  
 txt <- format(c(r, 0.123456789), digits = digits)[1]  
 txt <- paste("r= ", txt, sep = "")  
 text(0.5, 0.6, txt)  
  
 # p-value calculation  
 p <- cor.test(x, y, method = "pearson")$p.value  
 txt2 <- format(c(p, 0.123456789), digits = digits)[1]  
 txt2 <- paste("p= ", txt2, sep = "")  
 if(p<0.01) txt2 <- paste("p ", "< 0.01", sep = "")  
 text(0.5, 0.4, txt2)  
}

scatterplotMatrix(~mpg+displacement+horsepower+weight+acceleration|cylinders , data=df,main="Three different Cylinders(4,6,8)",lower.panel = panel.cor)

 Observing the scatter of points for the 3 different Cylinder numbers, we can see that each of them appears distinct for each of the quantitative values shown above.

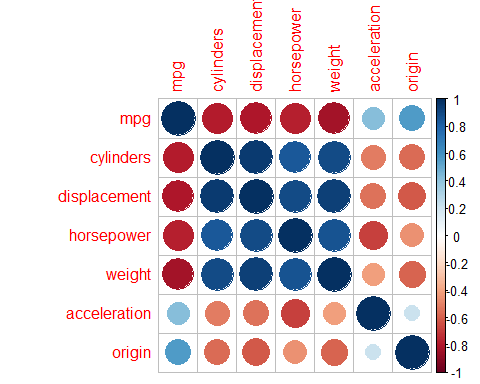
For example, considering the plot for mpg vs displacement, we can observe that 4 Cylinder cars appear to have higher mpg for lower displacement values as compared to 6 cylinder cars.

scatterplotMatrix(~mpg+displacement+horsepower+weight+acceleration|origin , data=df,main="Three different Origins (1,2,3)",lower.panel = panel.cor)

 Observing the scatter of points for the 3 origin countries, we can see that American made Cars(1) are distinct from Japanese(3) and European(2) made Cars, both of which appear to be dispersed in similar regions of the plot.

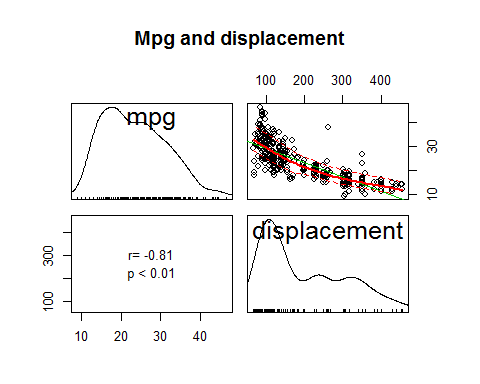
We can view the correlation between different quantitative variables in the dataset using the following visualization.

rv <- cor(df, use = "complete.obs")  
corrplot(rv)

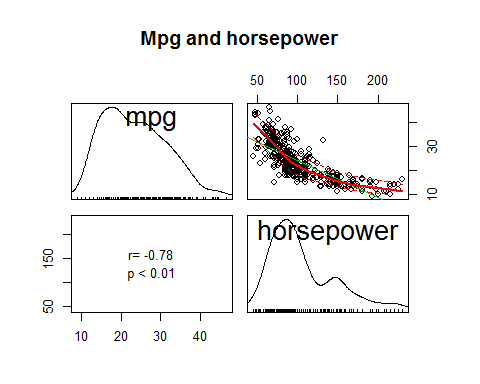
 The size of the circle and the intensity of the shade of color are the indicators of correlation between two variables.

**Inferences from the various Plots**

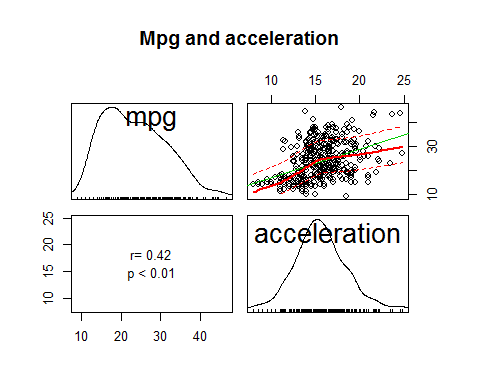
scatterplotMatrix(~mpg+displacement , data=df,main="Mpg and displacement",lower.panel = panel.cor)

 We can observe a slightly non-linear relationship between Displacement and mpg. They are negatively correlated (-0.805) and it is significant.

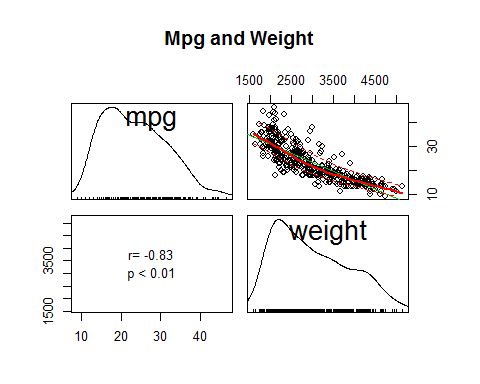
scatterplotMatrix(~mpg+horsepower , data=df,main="Mpg and horsepower",lower.panel = panel.cor)

 Again we observe a slightly non-linear relationship between mpg and horsepower with a correlation of (-0.78) which is significant.

scatterplotMatrix(~mpg+acceleration , data=df,main="Mpg and acceleration",lower.panel = panel.cor)

 There doesn’t seem to be a proper relationship between mpg and acceleration. This can be further investigated.

scatterplotMatrix(~mpg+weight , data=df,main="Mpg and Weight",lower.panel = panel.cor)

 As can be seen from the above plot, Weight seems to have a linear relationship with mpg with a correlation of -0.83 which is significant.

Model Building

***After accounting for the non-linear effects (if any) and the interactions (if any), present among the independent variables, as revealed through the preliminary visualization step above (or even otherwise, if your prior engineering knowledge compels you to do so), build the “best possible” regression model you can, according to the AIC criterion. You should automate this model building process with the aid of the R function stepAIC (residing in the library MASS), a successful deployment of which requires preliminary experimentation with a few “small” and a few “large” initial models, between which the “best” model is chosen; but must write one paragraph clearly describing the underlying algorithm, and more importantly its statistical logic, for selecting the “best possible” model.***

Step 1 : We need to encode the categorical variables (Cylinders and Origin) for the regression function. It is done as follows:

df\_mod <- df  
  
for ( i in 1:nrow(df\_mod)){  
 if (df\_mod$cylinders[i] == 4){  
 df\_mod$cy\_dum1[i] = 0  
 df\_mod$cy\_dum2[i] = 0  
 }  
 if (df\_mod$cylinders[i] == 6){  
 df\_mod$cy\_dum1[i] = 1  
 df\_mod$cy\_dum2[i] = 0  
 }  
 if (df\_mod$cylinders[i] == 8){  
 df\_mod$cy\_dum1[i] = 0  
 df\_mod$cy\_dum2[i] = 1  
 }  
 if (df\_mod$origin[i] == 1){  
 df\_mod$or\_dum1[i] = 0  
 df\_mod$or\_dum2[i] = 0  
 }  
 if (df\_mod$origin[i] == 2){  
 df\_mod$or\_dum1[i] = 1  
 df\_mod$or\_dum2[i] = 0  
 }  
 if (df\_mod$origin[i] == 3){  
 df\_mod$or\_dum1[i] = 0  
 df\_mod$or\_dum2[i] = 1  
 }  
}  
   
df\_mod$cylinders <- NULL  
df\_mod$origin <- NULL  
#head(filter(df\_mod,cy\_dum1 == 1))  
head(df\_mod)

## mpg displacement horsepower weight acceleration cy\_dum1 cy\_dum2 or\_dum1  
## 1 18 307 130 3504 12.0 0 1 0  
## 2 15 350 165 3693 11.5 0 1 0  
## 3 18 318 150 3436 11.0 0 1 0  
## 4 16 304 150 3433 12.0 0 1 0  
## 5 17 302 140 3449 10.5 0 1 0  
## 6 15 429 198 4341 10.0 0 1 0  
## or\_dum2  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 0

We can fit a First Order Regression Model as follows:

mlrm <- lm(mpg~displacement+horsepower+weight+acceleration+cy\_dum1+cy\_dum2+or\_dum1+or\_dum2,data =df\_mod)  
summary(mlrm)

##   
## Call:  
## lm(formula = mpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2, data = df\_mod)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.9528 -2.3332 -0.5641 1.8038 15.4492   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 43.3845353 2.4259825 17.883 < 2e-16 \*\*\*  
## displacement 0.0145414 0.0089989 1.616 0.106938   
## horsepower -0.0830534 0.0165523 -5.018 8.02e-07 \*\*\*  
## weight -0.0042321 0.0007811 -5.418 1.06e-07 \*\*\*  
## acceleration -0.0319607 0.1183801 -0.270 0.787318   
## cy\_dum1 -4.2807663 0.8843062 -4.841 1.88e-06 \*\*\*  
## cy\_dum2 -2.4627907 1.5657038 -1.573 0.116554   
## or\_dum1 0.4022300 0.6829354 0.589 0.556227   
## or\_dum2 2.4457120 0.6750867 3.623 0.000331 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.989 on 383 degrees of freedom  
## Multiple R-squared: 0.7441, Adjusted R-squared: 0.7388   
## F-statistic: 139.2 on 8 and 383 DF, p-value: < 2.2e-16

As can be seen from the summary, only Horsepower, weight, cy\_dum1 and or\_dum2 are significant when considering a full model.

full\_linear\_model <- lm(formula = mpg ~ displacement + horsepower +weight + acceleration +cy\_dum1 +cy\_dum2 + or\_dum2 +or\_dum1 ,data =df\_mod )  
  
#setting up the steps for MLRM   
multistep <- stepAIC(full\_linear\_model,direction = "backward", scope=list(upper=~.,lower=~1), trace = FALSE)  
summary(multistep)

##   
## Call:  
## lm(formula = mpg ~ displacement + horsepower + weight + cy\_dum1 +   
## cy\_dum2 + or\_dum2, data = df\_mod)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.962 -2.328 -0.558 1.730 15.507   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 43.0454898 1.3918626 30.927 < 2e-16 \*\*\*  
## displacement 0.0130895 0.0085030 1.539 0.12453   
## horsepower -0.0791885 0.0130697 -6.059 3.26e-09 \*\*\*  
## weight -0.0042919 0.0006855 -6.261 1.02e-09 \*\*\*  
## cy\_dum1 -4.3110327 0.8778757 -4.911 1.34e-06 \*\*\*  
## cy\_dum2 -2.4026858 1.5569953 -1.543 0.12361   
## or\_dum2 2.2435637 0.5804864 3.865 0.00013 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.981 on 385 degrees of freedom  
## Multiple R-squared: 0.7438, Adjusted R-squared: 0.7399   
## F-statistic: 186.3 on 6 and 385 DF, p-value: < 2.2e-16

multistep$anova

## Stepwise Model Path   
## Analysis of Deviance Table  
##   
## Initial Model:  
## mpg ~ displacement + horsepower + weight + acceleration + cy\_dum1 +   
## cy\_dum2 + or\_dum2 + or\_dum1  
##   
## Final Model:  
## mpg ~ displacement + horsepower + weight + cy\_dum1 + cy\_dum2 +   
## or\_dum2  
##   
##   
## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 383 6094.707 1093.614  
## 2 - acceleration 1 1.159920 384 6095.866 1091.689  
## 3 - or\_dum1 1 5.397741 385 6101.264 1090.036

As can be seen from the results of the stepAIC function, the Final First Order Model obtained is as follows:

mpg ~ displacement + horsepower + weight + cy\_dum1 + cy\_dum2 + or\_dum2

Note : stepAIC selects a model based on reduction in AIC values and not p-values. Therefore, it is possible that the model may contain certain insignificant terms.

We can check the results of building a Model based on the results of AIC function :

simple\_linear <- lm(formula = mpg ~ displacement + horsepower + weight + cy\_dum1 +   
 cy\_dum2 + or\_dum2, data = df\_mod)  
summary(simple\_linear)

##   
## Call:  
## lm(formula = mpg ~ displacement + horsepower + weight + cy\_dum1 +   
## cy\_dum2 + or\_dum2, data = df\_mod)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.962 -2.328 -0.558 1.730 15.507   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 43.0454898 1.3918626 30.927 < 2e-16 \*\*\*  
## displacement 0.0130895 0.0085030 1.539 0.12453   
## horsepower -0.0791885 0.0130697 -6.059 3.26e-09 \*\*\*  
## weight -0.0042919 0.0006855 -6.261 1.02e-09 \*\*\*  
## cy\_dum1 -4.3110327 0.8778757 -4.911 1.34e-06 \*\*\*  
## cy\_dum2 -2.4026858 1.5569953 -1.543 0.12361   
## or\_dum2 2.2435637 0.5804864 3.865 0.00013 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.981 on 385 degrees of freedom  
## Multiple R-squared: 0.7438, Adjusted R-squared: 0.7399   
## F-statistic: 186.3 on 6 and 385 DF, p-value: < 2.2e-16

AIC(simple\_linear)

## [1] 2204.484

Since we are basing our model only on Akaike Information Criteria and not the significance of the terms, we let the insignificant terms remain in the Model.

Eliminating the insignificant values (based on the p-values from the above results), we can build the next model as follows:

simple\_linear <- lm(formula = mpg ~ horsepower + weight + cy\_dum1 +   
 or\_dum2, data = df\_mod)  
summary(simple\_linear)

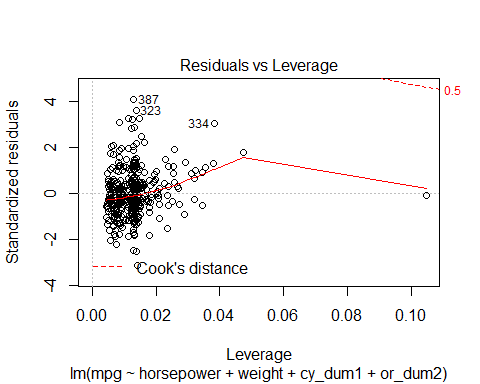
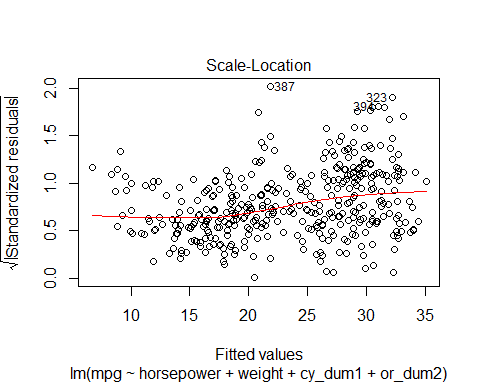
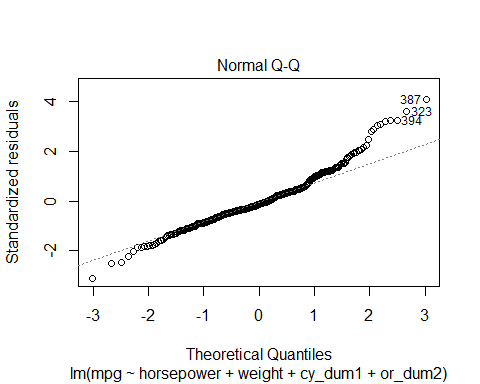
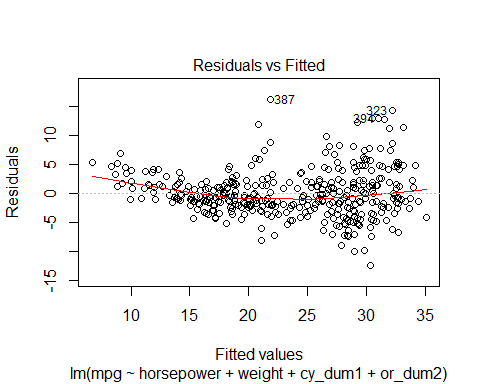
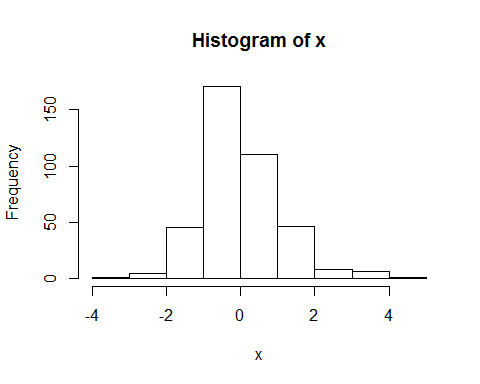
##   
## Call:  
## lm(formula = mpg ~ horsepower + weight + cy\_dum1 + or\_dum2, data = df\_mod)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.3375 -2.3004 -0.5498 1.8704 16.1863   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 43.6431383 0.8925331 48.898 < 2e-16 \*\*\*  
## horsepower -0.0740746 0.0110633 -6.696 7.58e-11 \*\*\*  
## weight -0.0040980 0.0005295 -7.739 8.82e-14 \*\*\*  
## cy\_dum1 -3.1777240 0.5263160 -6.038 3.67e-09 \*\*\*  
## or\_dum2 2.0651413 0.5689777 3.630 0.000322 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.985 on 387 degrees of freedom  
## Multiple R-squared: 0.742, Adjusted R-squared: 0.7393   
## F-statistic: 278.2 on 4 and 387 DF, p-value: < 2.2e-16

AIC(simple\_linear)

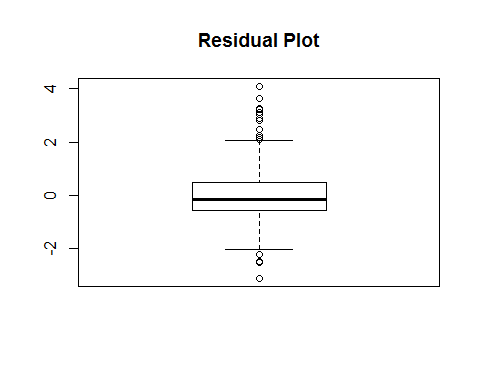
## [1] 2203.376

Now that we have a Fitted First Order Model, we can do a residual analysis to see if the Residuals stand up to the assumptions of Homoscedasticity and Normality.

resAnalysis(simple\_linear)



## Method P.value  
## 1 Shapiro-Wilk normality test 1.507085e-08  
## 2 Anderson-Darling normality test 7.217212e-10  
## 3 Cramer-von Mises normality test 6.833572e-08  
## 4 Lilliefors (Kolmogorov-Smirnov) normality test 9.751788e-07  
## 5 Shapiro-Francia normality test 6.537642e-08  
##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 23.556, df = 4, p-value = 9.804e-05  
##   
## [1] "Outliers in the data :"



## 72 109 112 113 245 310 323   
## -2.520420 -2.487864 -3.117838 -2.236721 2.814473 3.092914 3.626676   
## 325 326 327 328 330 334 361   
## 2.161177 3.221343 3.256378 2.480668 2.888906 3.037559 2.229952   
## 387 391 394   
## 4.088112 2.087624 3.265185

As can be seen from the Residual Analysis, both Normality and Heteroscedasticity assumptions are violated. This implies, we need to transform the variables in our model to try and achieve Normality and Heteroscedasticity.

Before attempting that, we can try Fitting a Second Order Model before we proceed with transforming variables.

# Fitting square models after setting up the lower as constant number and upper as Square function of SLRM  
#First Order MLRM  
mlrm <- lm(mpg~displacement+horsepower+weight+acceleration+cy\_dum1+cy\_dum2+or\_dum1+or\_dum2,data =df\_mod)  
  
lowerModel <- lm(mpg ~ 1, data=df\_mod)  
upperModel <- lm(mpg ~ (.)^2, data = df\_mod)  
multistep2 <- stepAIC(mlrm,direction = "both", scope=list(upper= upperModel,lower= lowerModel), trace= FALSE)  
summary(multistep2)

##   
## Call:  
## lm(formula = mpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:horsepower +   
## horsepower:cy\_dum1 + displacement:or\_dum2 + horsepower:or\_dum2 +   
## acceleration:or\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration +   
## acceleration:cy\_dum1 + displacement:cy\_dum2, data = df\_mod)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.2779 -1.9237 -0.2878 1.6992 19.1632   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.116e+01 7.139e+00 7.166 4.14e-12 \*\*\*  
## displacement -5.516e-02 1.554e-02 -3.550 0.000434 \*\*\*  
## horsepower -5.084e-03 8.345e-02 -0.061 0.951454   
## weight -1.626e-03 8.649e-04 -1.880 0.060866 .   
## acceleration -1.288e-02 3.399e-01 -0.038 0.969791   
## cy\_dum1 -2.174e+01 7.141e+00 -3.045 0.002493 \*\*   
## cy\_dum2 -3.183e+01 8.438e+00 -3.772 0.000188 \*\*\*  
## or\_dum1 -1.600e+01 3.771e+00 -4.243 2.78e-05 \*\*\*  
## or\_dum2 4.560e+00 2.626e+00 1.736 0.083309 .   
## displacement:horsepower 2.105e-04 1.343e-04 1.567 0.117898   
## horsepower:cy\_dum1 1.029e-01 3.582e-02 2.873 0.004303 \*\*   
## displacement:or\_dum2 1.204e-01 2.726e-02 4.415 1.32e-05 \*\*\*  
## horsepower:or\_dum2 -2.028e-01 4.246e-02 -4.776 2.57e-06 \*\*\*  
## acceleration:or\_dum1 8.895e-01 2.226e-01 3.995 7.79e-05 \*\*\*  
## acceleration:cy\_dum2 1.619e+00 3.549e-01 4.561 6.91e-06 \*\*\*  
## horsepower:acceleration -1.047e-02 4.035e-03 -2.595 0.009818 \*\*   
## acceleration:cy\_dum1 6.544e-01 3.046e-01 2.148 0.032318 \*   
## displacement:cy\_dum2 3.111e-02 2.221e-02 1.401 0.162134   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.532 on 374 degrees of freedom  
## Multiple R-squared: 0.8041, Adjusted R-squared: 0.7952   
## F-statistic: 90.31 on 17 and 374 DF, p-value: < 2.2e-16

Based on AIC, a second order Model is given as follows:

mpg ~ displacement + horsepower + weight + acceleration + cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:horsepower + horsepower:cy\_dum1 + displacement:or\_dum2 + horsepower:or\_dum2 + acceleration:or\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration + acceleration:cy\_dum1 + displacement:cy\_dum2

The 2nd Order Model can be viewed as follows:

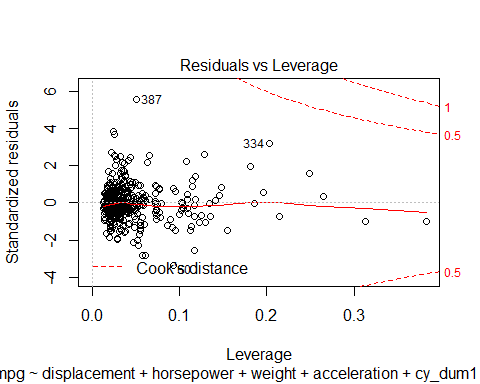
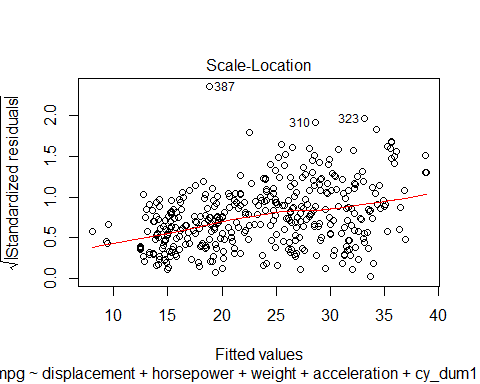
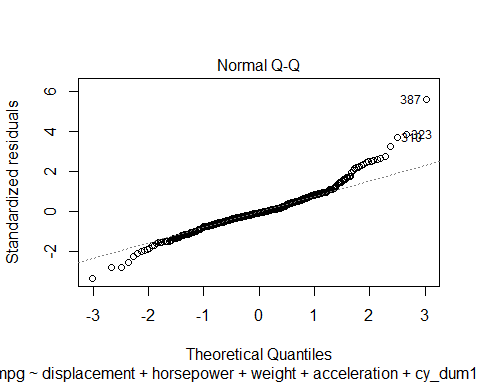
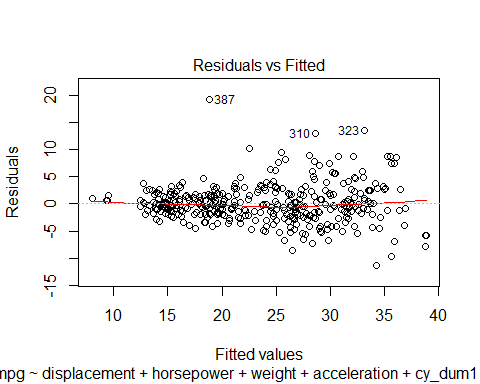
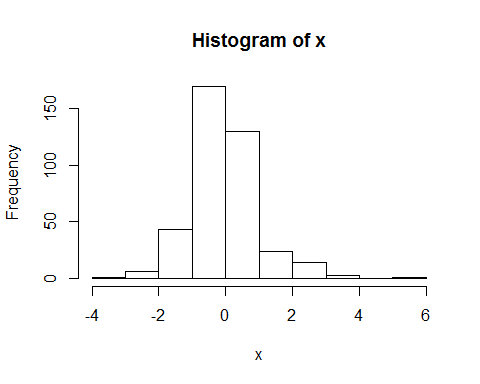
second\_order\_model <- multistep2  
summary(second\_order\_model)

##   
## Call:  
## lm(formula = mpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:horsepower +   
## horsepower:cy\_dum1 + displacement:or\_dum2 + horsepower:or\_dum2 +   
## acceleration:or\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration +   
## acceleration:cy\_dum1 + displacement:cy\_dum2, data = df\_mod)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.2779 -1.9237 -0.2878 1.6992 19.1632   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.116e+01 7.139e+00 7.166 4.14e-12 \*\*\*  
## displacement -5.516e-02 1.554e-02 -3.550 0.000434 \*\*\*  
## horsepower -5.084e-03 8.345e-02 -0.061 0.951454   
## weight -1.626e-03 8.649e-04 -1.880 0.060866 .   
## acceleration -1.288e-02 3.399e-01 -0.038 0.969791   
## cy\_dum1 -2.174e+01 7.141e+00 -3.045 0.002493 \*\*   
## cy\_dum2 -3.183e+01 8.438e+00 -3.772 0.000188 \*\*\*  
## or\_dum1 -1.600e+01 3.771e+00 -4.243 2.78e-05 \*\*\*  
## or\_dum2 4.560e+00 2.626e+00 1.736 0.083309 .   
## displacement:horsepower 2.105e-04 1.343e-04 1.567 0.117898   
## horsepower:cy\_dum1 1.029e-01 3.582e-02 2.873 0.004303 \*\*   
## displacement:or\_dum2 1.204e-01 2.726e-02 4.415 1.32e-05 \*\*\*  
## horsepower:or\_dum2 -2.028e-01 4.246e-02 -4.776 2.57e-06 \*\*\*  
## acceleration:or\_dum1 8.895e-01 2.226e-01 3.995 7.79e-05 \*\*\*  
## acceleration:cy\_dum2 1.619e+00 3.549e-01 4.561 6.91e-06 \*\*\*  
## horsepower:acceleration -1.047e-02 4.035e-03 -2.595 0.009818 \*\*   
## acceleration:cy\_dum1 6.544e-01 3.046e-01 2.148 0.032318 \*   
## displacement:cy\_dum2 3.111e-02 2.221e-02 1.401 0.162134   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.532 on 374 degrees of freedom  
## Multiple R-squared: 0.8041, Adjusted R-squared: 0.7952   
## F-statistic: 90.31 on 17 and 374 DF, p-value: < 2.2e-16

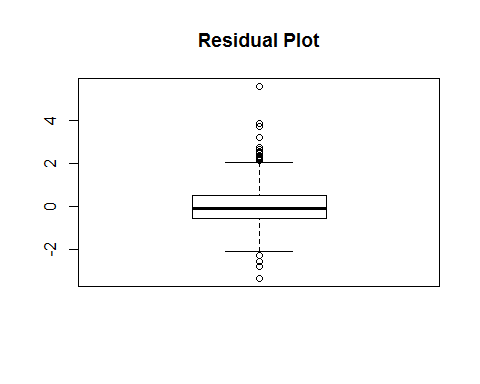
As can be seen, there are some insignificant terms in the 2nd Order Model because the selection of Variables was done on the basis on AIC and not p-value significance.

For now, we let the insignificant terms remain and do a Residual Analysis of the Model we have.

resAnalysis(second\_order\_model)



## Method P.value  
## 1 Shapiro-Wilk normality test 1.314359e-10  
## 2 Anderson-Darling normality test 5.641171e-12  
## 3 Cramer-von Mises normality test 9.232090e-09  
## 4 Lilliefors (Kolmogorov-Smirnov) normality test 3.988059e-07  
## 5 Shapiro-Francia normality test 7.737140e-10  
##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 56.415, df = 17, p-value = 4.052e-06  
##   
## [1] "Outliers in the data :"



## 20 60 103 112 145 245 248   
## -2.806000 -3.352132 -2.813753 -2.574605 -2.273533 2.173992 2.491665   
## 305 310 321 323 325 326 327   
## 2.132536 3.692161 2.565983 3.858415 2.486266 2.553930 2.241304   
## 328 330 334 336 358 370 387   
## 2.276093 2.450376 3.213920 2.721796 2.205315 2.328926 5.568382   
## 394   
## 2.635945

As can be seen from the test results above, for the 2nd Order Model also, Normality Tests and Breusch-Pagan Tests have failed indicating that Normality and Homoscedasticity Assumptions are violated.

We can try a Log Transformation on the mpg variable and then fitting a second Order Model.

1. Log Transformation

df\_mod2 <- df\_mod  
df\_mod2$lmpg <- log(df\_mod2$mpg)  
df\_mod2$mpg <- NULL  
head(df\_mod2)

## displacement horsepower weight acceleration cy\_dum1 cy\_dum2 or\_dum1  
## 1 307 130 3504 12.0 0 1 0  
## 2 350 165 3693 11.5 0 1 0  
## 3 318 150 3436 11.0 0 1 0  
## 4 304 150 3433 12.0 0 1 0  
## 5 302 140 3449 10.5 0 1 0  
## 6 429 198 4341 10.0 0 1 0  
## or\_dum2 lmpg  
## 1 0 2.890372  
## 2 0 2.708050  
## 3 0 2.890372  
## 4 0 2.772589  
## 5 0 2.833213  
## 6 0 2.708050

Fitting a second Order Model as follows:

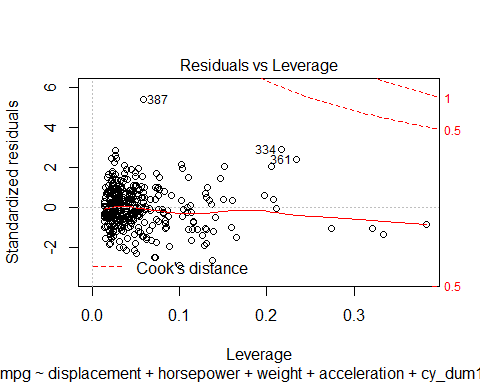
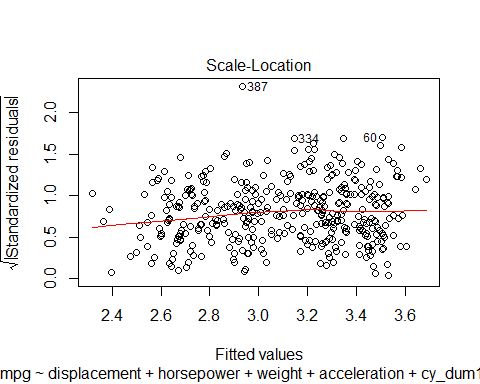
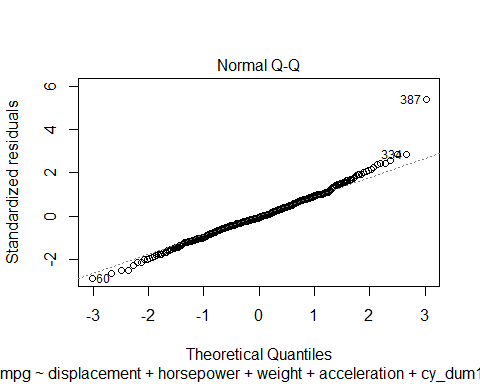
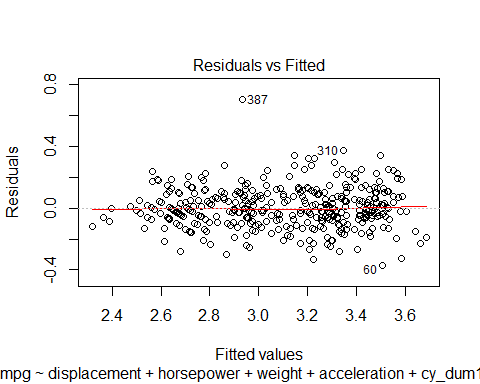
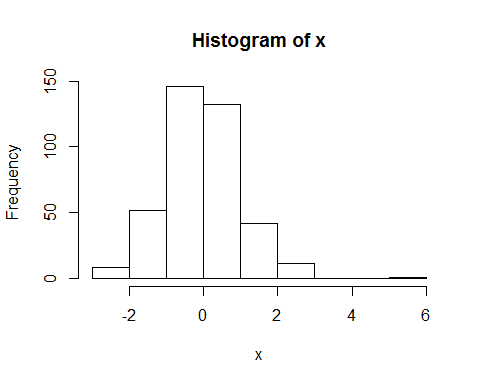
# Fitting square models after setting up the lower as constant number and upper as Square function of SLRM  
#First Order MLRM  
mlrm <- lm(lmpg~displacement+horsepower+weight+acceleration+cy\_dum1+cy\_dum2+or\_dum1+or\_dum2,data =df\_mod2)  
  
lowerModel <- lm(lmpg ~ 1, data=df\_mod2)  
upperModel <- lm(lmpg ~ (.)^2, data = df\_mod2)  
multistep2 <- stepAIC(mlrm,direction = "both", scope=list(upper= upperModel,lower= lowerModel), trace= FALSE)  
summary(multistep2)

##   
## Call:  
## lm(formula = lmpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:horsepower +   
## horsepower:cy\_dum1 + displacement:cy\_dum1 + acceleration:cy\_dum2 +   
## horsepower:acceleration + weight:acceleration + horsepower:or\_dum2 +   
## displacement:or\_dum2 + acceleration:or\_dum1 + acceleration:or\_dum2 +   
## acceleration:cy\_dum1, data = df\_mod2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.36888 -0.07831 -0.00819 0.07851 0.70490   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.423e+00 3.661e-01 12.079 < 2e-16 \*\*\*  
## displacement -6.965e-04 7.212e-04 -0.966 0.334739   
## horsepower 2.339e-03 2.989e-03 0.782 0.434530   
## weight -2.651e-04 1.034e-04 -2.564 0.010736 \*   
## acceleration -1.556e-02 1.934e-02 -0.804 0.421679   
## cy\_dum1 -6.454e-01 3.006e-01 -2.147 0.032427 \*   
## cy\_dum2 -1.060e+00 2.471e-01 -4.289 2.29e-05 \*\*\*  
## or\_dum1 -4.795e-01 1.536e-01 -3.121 0.001943 \*\*   
## or\_dum2 -3.796e-01 3.122e-01 -1.216 0.224819   
## displacement:horsepower 6.367e-06 3.853e-06 1.653 0.099243 .   
## horsepower:cy\_dum1 4.765e-03 1.390e-03 3.429 0.000674 \*\*\*  
## displacement:cy\_dum1 -1.409e-03 6.929e-04 -2.033 0.042772 \*   
## acceleration:cy\_dum2 6.045e-02 1.517e-02 3.984 8.17e-05 \*\*\*  
## horsepower:acceleration -6.383e-04 1.601e-04 -3.988 8.04e-05 \*\*\*  
## weight:acceleration 1.194e-05 6.877e-06 1.736 0.083457 .   
## horsepower:or\_dum2 -4.335e-03 1.751e-03 -2.475 0.013771 \*   
## displacement:or\_dum2 3.647e-03 1.081e-03 3.375 0.000816 \*\*\*  
## acceleration:or\_dum1 2.596e-02 9.196e-03 2.823 0.005019 \*\*   
## acceleration:or\_dum2 2.363e-02 1.357e-02 1.741 0.082531 .   
## acceleration:cy\_dum1 1.974e-02 1.344e-02 1.469 0.142743   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.135 on 372 degrees of freedom  
## Multiple R-squared: 0.8501, Adjusted R-squared: 0.8424   
## F-statistic: 111 on 19 and 372 DF, p-value: < 2.2e-16

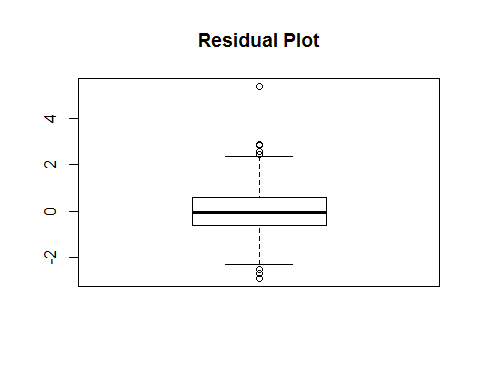
second\_order\_model\_log <- multistep2  
summary(second\_order\_model\_log)

##   
## Call:  
## lm(formula = lmpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:horsepower +   
## horsepower:cy\_dum1 + displacement:cy\_dum1 + acceleration:cy\_dum2 +   
## horsepower:acceleration + weight:acceleration + horsepower:or\_dum2 +   
## displacement:or\_dum2 + acceleration:or\_dum1 + acceleration:or\_dum2 +   
## acceleration:cy\_dum1, data = df\_mod2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.36888 -0.07831 -0.00819 0.07851 0.70490   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.423e+00 3.661e-01 12.079 < 2e-16 \*\*\*  
## displacement -6.965e-04 7.212e-04 -0.966 0.334739   
## horsepower 2.339e-03 2.989e-03 0.782 0.434530   
## weight -2.651e-04 1.034e-04 -2.564 0.010736 \*   
## acceleration -1.556e-02 1.934e-02 -0.804 0.421679   
## cy\_dum1 -6.454e-01 3.006e-01 -2.147 0.032427 \*   
## cy\_dum2 -1.060e+00 2.471e-01 -4.289 2.29e-05 \*\*\*  
## or\_dum1 -4.795e-01 1.536e-01 -3.121 0.001943 \*\*   
## or\_dum2 -3.796e-01 3.122e-01 -1.216 0.224819   
## displacement:horsepower 6.367e-06 3.853e-06 1.653 0.099243 .   
## horsepower:cy\_dum1 4.765e-03 1.390e-03 3.429 0.000674 \*\*\*  
## displacement:cy\_dum1 -1.409e-03 6.929e-04 -2.033 0.042772 \*   
## acceleration:cy\_dum2 6.045e-02 1.517e-02 3.984 8.17e-05 \*\*\*  
## horsepower:acceleration -6.383e-04 1.601e-04 -3.988 8.04e-05 \*\*\*  
## weight:acceleration 1.194e-05 6.877e-06 1.736 0.083457 .   
## horsepower:or\_dum2 -4.335e-03 1.751e-03 -2.475 0.013771 \*   
## displacement:or\_dum2 3.647e-03 1.081e-03 3.375 0.000816 \*\*\*  
## acceleration:or\_dum1 2.596e-02 9.196e-03 2.823 0.005019 \*\*   
## acceleration:or\_dum2 2.363e-02 1.357e-02 1.741 0.082531 .   
## acceleration:cy\_dum1 1.974e-02 1.344e-02 1.469 0.142743   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.135 on 372 degrees of freedom  
## Multiple R-squared: 0.8501, Adjusted R-squared: 0.8424   
## F-statistic: 111 on 19 and 372 DF, p-value: < 2.2e-16

resAnalysis(second\_order\_model\_log)



## Method P.value  
## 1 Shapiro-Wilk normality test 1.011055e-04  
## 2 Anderson-Darling normality test 2.023258e-02  
## 3 Cramer-von Mises normality test 1.666477e-02  
## 4 Lilliefors (Kolmogorov-Smirnov) normality test 1.045824e-01  
## 5 Shapiro-Francia normality test 6.835788e-05  
##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 38.465, df = 19, p-value = 0.005177  
##   
## [1] "Outliers in the data :"



## 20 60 103 112 310 323 334   
## -2.518436 -2.880304 -2.514948 -2.651366 2.837956 2.595093 2.870247   
## 336 370 387   
## 2.444199 2.438440 5.381128

The test for Normality and Heteroscedasticity still fails, so we can attempt to remove some Outliers from the data.

1. Removing the Outliers

df\_mod\_3 <- df\_mod2[c(-20, -60,-103, -112, -310,-323, -334,-336,-370,-387),]

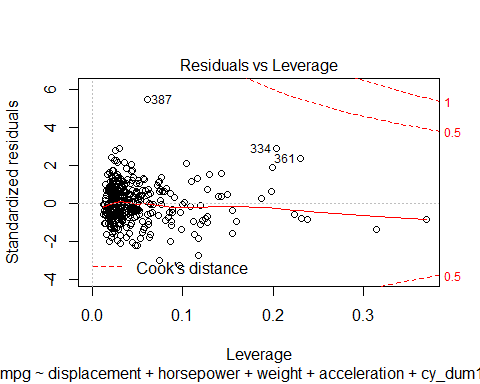
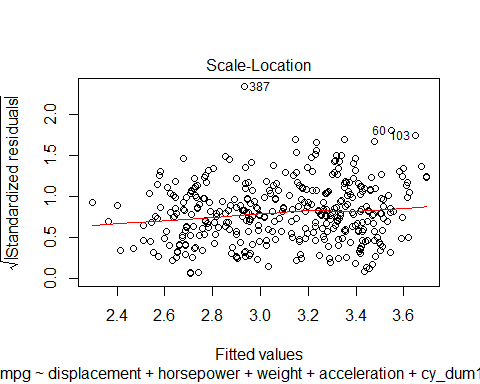
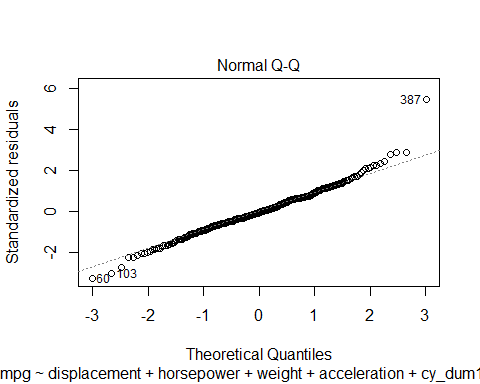
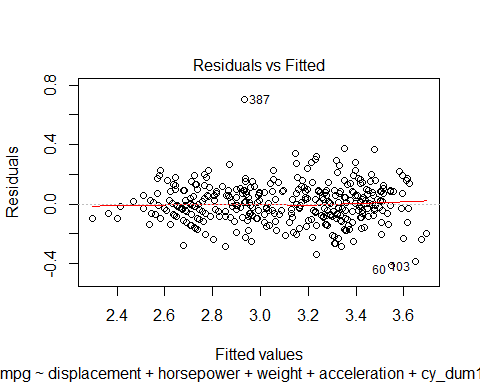
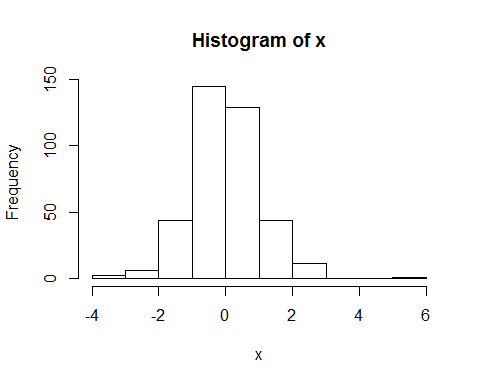
# Fitting square models after setting up the lower as constant number and upper as Square function of SLRM  
#First Order MLRM  
mlrm <- lm(lmpg~displacement+horsepower+weight+acceleration+cy\_dum1+cy\_dum2+or\_dum1+or\_dum2,data =df\_mod\_3)  
  
lowerModel <- lm(lmpg ~ 1, data=df\_mod\_3)  
upperModel <- lm(lmpg ~ (.)^2, data = df\_mod\_3)  
multistep3 <- stepAIC(mlrm,direction = "both", scope=list(upper= upperModel,lower= lowerModel), trace= FALSE)  
summary(multistep3)

##   
## Call:  
## lm(formula = lmpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:cy\_dum1 +   
## horsepower:cy\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration +   
## acceleration:or\_dum1 + horsepower:or\_dum2 + displacement:or\_dum2 +   
## acceleration:cy\_dum1 + horsepower:or\_dum1, data = df\_mod\_3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.41206 -0.07731 -0.00829 0.08221 0.70335   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.675e+00 1.675e-01 21.945 < 2e-16 \*\*\*  
## displacement 4.067e-04 4.897e-04 0.831 0.406786   
## horsepower 3.926e-03 1.855e-03 2.116 0.035022 \*   
## weight -1.038e-04 3.374e-05 -3.077 0.002247 \*\*   
## acceleration 1.974e-02 1.053e-02 1.875 0.061581 .   
## cy\_dum1 -6.511e-01 2.786e-01 -2.337 0.019994 \*   
## cy\_dum2 -1.231e+00 2.020e-01 -6.095 2.79e-09 \*\*\*  
## or\_dum1 -1.961e-01 2.090e-01 -0.939 0.348532   
## or\_dum2 1.945e-01 9.698e-02 2.005 0.045657 \*   
## displacement:cy\_dum1 -2.051e-03 6.886e-04 -2.978 0.003092 \*\*   
## horsepower:cy\_dum1 4.882e-03 1.368e-03 3.570 0.000406 \*\*\*  
## acceleration:cy\_dum2 6.620e-02 1.287e-02 5.145 4.39e-07 \*\*\*  
## horsepower:acceleration -5.960e-04 1.281e-04 -4.654 4.57e-06 \*\*\*  
## acceleration:or\_dum1 1.920e-02 9.037e-03 2.125 0.034295 \*   
## horsepower:or\_dum2 -6.474e-03 1.461e-03 -4.430 1.25e-05 \*\*\*  
## displacement:or\_dum2 3.458e-03 1.068e-03 3.239 0.001310 \*\*   
## acceleration:cy\_dum1 2.537e-02 1.144e-02 2.217 0.027214 \*   
## horsepower:or\_dum1 -1.987e-03 1.172e-03 -1.696 0.090663 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1328 on 364 degrees of freedom  
## Multiple R-squared: 0.853, Adjusted R-squared: 0.8462   
## F-statistic: 124.3 on 17 and 364 DF, p-value: < 2.2e-16

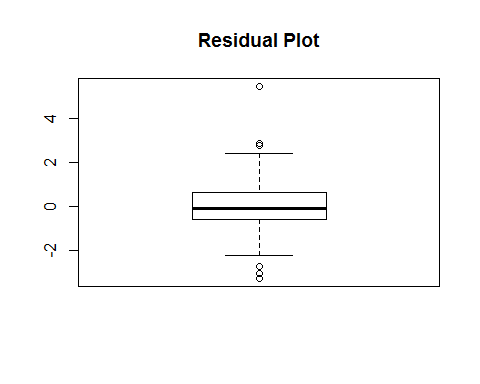
second\_order\_model\_log\_outlier1 <- multistep3  
summary(second\_order\_model\_log\_outlier1)

##   
## Call:  
## lm(formula = lmpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:cy\_dum1 +   
## horsepower:cy\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration +   
## acceleration:or\_dum1 + horsepower:or\_dum2 + displacement:or\_dum2 +   
## acceleration:cy\_dum1 + horsepower:or\_dum1, data = df\_mod\_3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.41206 -0.07731 -0.00829 0.08221 0.70335   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.675e+00 1.675e-01 21.945 < 2e-16 \*\*\*  
## displacement 4.067e-04 4.897e-04 0.831 0.406786   
## horsepower 3.926e-03 1.855e-03 2.116 0.035022 \*   
## weight -1.038e-04 3.374e-05 -3.077 0.002247 \*\*   
## acceleration 1.974e-02 1.053e-02 1.875 0.061581 .   
## cy\_dum1 -6.511e-01 2.786e-01 -2.337 0.019994 \*   
## cy\_dum2 -1.231e+00 2.020e-01 -6.095 2.79e-09 \*\*\*  
## or\_dum1 -1.961e-01 2.090e-01 -0.939 0.348532   
## or\_dum2 1.945e-01 9.698e-02 2.005 0.045657 \*   
## displacement:cy\_dum1 -2.051e-03 6.886e-04 -2.978 0.003092 \*\*   
## horsepower:cy\_dum1 4.882e-03 1.368e-03 3.570 0.000406 \*\*\*  
## acceleration:cy\_dum2 6.620e-02 1.287e-02 5.145 4.39e-07 \*\*\*  
## horsepower:acceleration -5.960e-04 1.281e-04 -4.654 4.57e-06 \*\*\*  
## acceleration:or\_dum1 1.920e-02 9.037e-03 2.125 0.034295 \*   
## horsepower:or\_dum2 -6.474e-03 1.461e-03 -4.430 1.25e-05 \*\*\*  
## displacement:or\_dum2 3.458e-03 1.068e-03 3.239 0.001310 \*\*   
## acceleration:cy\_dum1 2.537e-02 1.144e-02 2.217 0.027214 \*   
## horsepower:or\_dum1 -1.987e-03 1.172e-03 -1.696 0.090663 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1328 on 364 degrees of freedom  
## Multiple R-squared: 0.853, Adjusted R-squared: 0.8462   
## F-statistic: 124.3 on 17 and 364 DF, p-value: < 2.2e-16

resAnalysis(second\_order\_model\_log\_outlier1)



## Method P.value  
## 1 Shapiro-Wilk normality test 4.136651e-05  
## 2 Anderson-Darling normality test 2.084559e-02  
## 3 Cramer-von Mises normality test 2.723352e-02  
## 4 Lilliefors (Kolmogorov-Smirnov) normality test 4.425938e-02  
## 5 Shapiro-Francia normality test 2.552735e-05  
##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 30.431, df = 17, p-value = 0.0234  
##   
## [1] "Outliers in the data :"



## 60 103 112 310 323 334 387   
## -3.263965 -3.041646 -2.740132 2.874037 2.782103 2.878276 5.463787

Based on the Residual Analysis, we can see that the Normality and Heteroscedasticity Assumption still fails.

We can try removing one more set of Outliers as given by the Outlier Analysis above.

1. Removing Outliers for the second time

df\_mod\_4 <- df\_mod\_3[c(-60,-103, -112, -310,-323, -334,-387),]

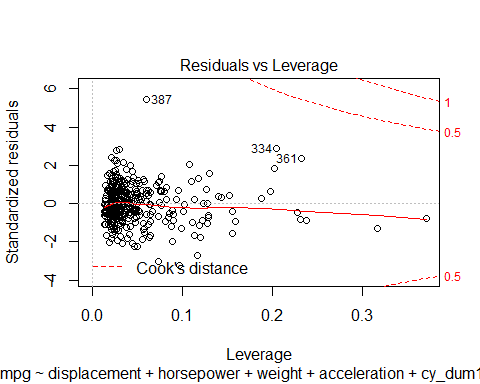
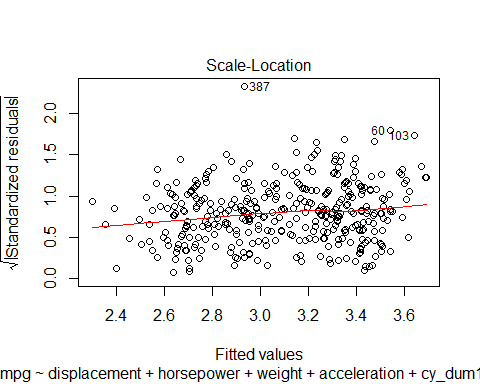
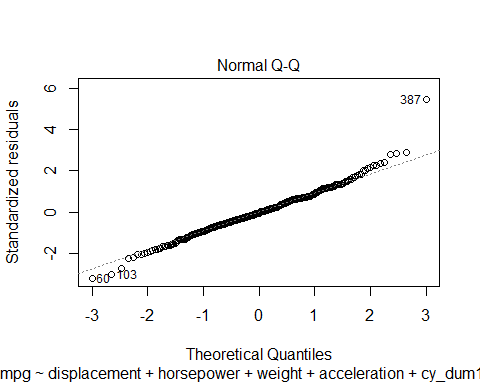
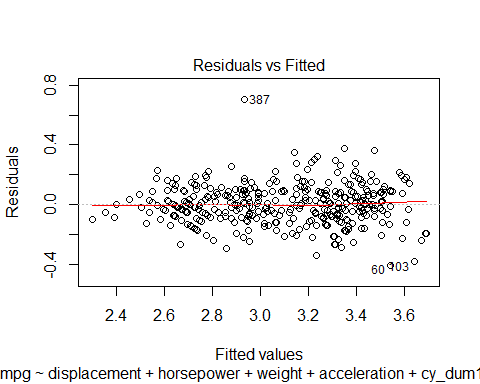
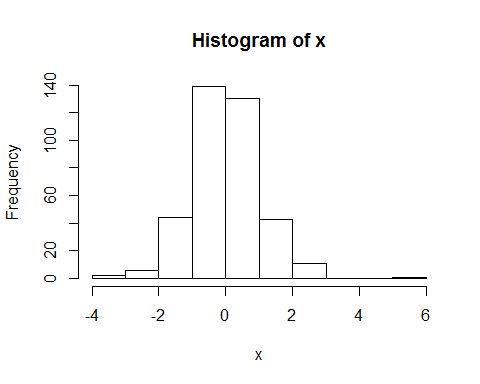
# Fitting square models after setting up the lower as constant number and upper as Square function of SLRM  
#First Order MLRM  
mlrm <- lm(lmpg~displacement+horsepower+weight+acceleration+cy\_dum1+cy\_dum2+or\_dum1+or\_dum2,data =df\_mod\_4)  
  
lowerModel <- lm(lmpg ~ 1, data=df\_mod\_4)  
upperModel <- lm(lmpg ~ (.)^2, data = df\_mod\_4)  
multistep4 <- stepAIC(mlrm,direction = "both", scope=list(upper= upperModel,lower= lowerModel), trace= FALSE)  
summary(multistep4)

##   
## Call:  
## lm(formula = lmpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:cy\_dum1 +   
## horsepower:cy\_dum1 + acceleration:cy\_dum2 + acceleration:or\_dum1 +   
## horsepower:acceleration + horsepower:or\_dum2 + displacement:or\_dum2 +   
## acceleration:cy\_dum1 + horsepower:or\_dum1, data = df\_mod\_4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.40749 -0.07877 -0.00512 0.08356 0.70327   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.736e+00 1.736e-01 21.526 < 2e-16 \*\*\*  
## displacement 5.249e-04 5.036e-04 1.042 0.297962   
## horsepower 2.996e-03 1.972e-03 1.519 0.129550   
## weight -1.061e-04 3.412e-05 -3.108 0.002036 \*\*   
## acceleration 1.578e-02 1.089e-02 1.450 0.148004   
## cy\_dum1 -6.269e-01 2.803e-01 -2.236 0.025949 \*   
## cy\_dum2 -1.194e+00 2.043e-01 -5.846 1.14e-08 \*\*\*  
## or\_dum1 -1.959e-01 2.100e-01 -0.933 0.351546   
## or\_dum2 1.988e-01 9.847e-02 2.019 0.044269 \*   
## displacement:cy\_dum1 -2.158e-03 6.969e-04 -3.096 0.002114 \*\*   
## horsepower:cy\_dum1 4.998e-03 1.378e-03 3.627 0.000328 \*\*\*  
## acceleration:cy\_dum2 6.323e-02 1.307e-02 4.839 1.94e-06 \*\*\*  
## acceleration:or\_dum1 1.936e-02 9.087e-03 2.130 0.033832 \*   
## horsepower:acceleration -5.434e-04 1.334e-04 -4.073 5.71e-05 \*\*\*  
## horsepower:or\_dum2 -6.318e-03 1.483e-03 -4.260 2.62e-05 \*\*\*  
## displacement:or\_dum2 3.321e-03 1.077e-03 3.083 0.002210 \*\*   
## acceleration:cy\_dum1 2.411e-02 1.156e-02 2.085 0.037739 \*   
## horsepower:or\_dum1 -1.985e-03 1.178e-03 -1.686 0.092650 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1333 on 358 degrees of freedom  
## Multiple R-squared: 0.8503, Adjusted R-squared: 0.8432   
## F-statistic: 119.6 on 17 and 358 DF, p-value: < 2.2e-16

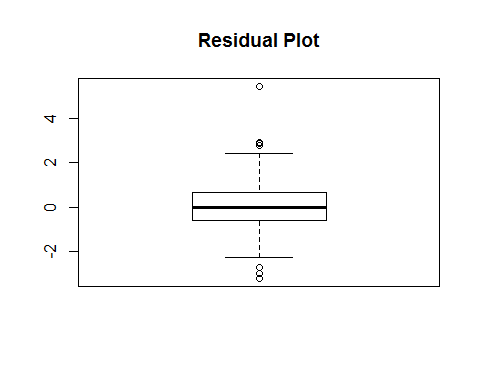
second\_order\_model\_log\_outlier2 <- multistep4  
summary(second\_order\_model\_log\_outlier2)

##   
## Call:  
## lm(formula = lmpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:cy\_dum1 +   
## horsepower:cy\_dum1 + acceleration:cy\_dum2 + acceleration:or\_dum1 +   
## horsepower:acceleration + horsepower:or\_dum2 + displacement:or\_dum2 +   
## acceleration:cy\_dum1 + horsepower:or\_dum1, data = df\_mod\_4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.40749 -0.07877 -0.00512 0.08356 0.70327   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.736e+00 1.736e-01 21.526 < 2e-16 \*\*\*  
## displacement 5.249e-04 5.036e-04 1.042 0.297962   
## horsepower 2.996e-03 1.972e-03 1.519 0.129550   
## weight -1.061e-04 3.412e-05 -3.108 0.002036 \*\*   
## acceleration 1.578e-02 1.089e-02 1.450 0.148004   
## cy\_dum1 -6.269e-01 2.803e-01 -2.236 0.025949 \*   
## cy\_dum2 -1.194e+00 2.043e-01 -5.846 1.14e-08 \*\*\*  
## or\_dum1 -1.959e-01 2.100e-01 -0.933 0.351546   
## or\_dum2 1.988e-01 9.847e-02 2.019 0.044269 \*   
## displacement:cy\_dum1 -2.158e-03 6.969e-04 -3.096 0.002114 \*\*   
## horsepower:cy\_dum1 4.998e-03 1.378e-03 3.627 0.000328 \*\*\*  
## acceleration:cy\_dum2 6.323e-02 1.307e-02 4.839 1.94e-06 \*\*\*  
## acceleration:or\_dum1 1.936e-02 9.087e-03 2.130 0.033832 \*   
## horsepower:acceleration -5.434e-04 1.334e-04 -4.073 5.71e-05 \*\*\*  
## horsepower:or\_dum2 -6.318e-03 1.483e-03 -4.260 2.62e-05 \*\*\*  
## displacement:or\_dum2 3.321e-03 1.077e-03 3.083 0.002210 \*\*   
## acceleration:cy\_dum1 2.411e-02 1.156e-02 2.085 0.037739 \*   
## horsepower:or\_dum1 -1.985e-03 1.178e-03 -1.686 0.092650 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1333 on 358 degrees of freedom  
## Multiple R-squared: 0.8503, Adjusted R-squared: 0.8432   
## F-statistic: 119.6 on 17 and 358 DF, p-value: < 2.2e-16

resAnalysis(second\_order\_model\_log\_outlier2)



## Method P.value  
## 1 Shapiro-Wilk normality test 0.0000548181  
## 2 Anderson-Darling normality test 0.0320187362  
## 3 Cramer-von Mises normality test 0.0445774665  
## 4 Lilliefors (Kolmogorov-Smirnov) normality test 0.0583250224  
## 5 Shapiro-Francia normality test 0.0000329422  
##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 29.762, df = 17, p-value = 0.02811  
##   
## [1] "Outliers in the data :"



## 60 103 112 310 323 334 387   
## -3.215913 -3.007434 -2.733877 2.856266 2.776920 2.910071 5.442648

As can be seen from the above results, the Normality Assumption is validated based on Lilliefors (Kolmogorov-Smirnov) normality test but the Homoscedasticity assumption is still not valid. Removing further Outliers doesn’t seem to affect the Breusch-Pagan Test results much.

Note : We tried Fitting a 3rd Order Model but none of the 3rd Order Terms were significant and stepAIC returned the same 2nd Order Model we have presented below.

Therefore, we present our Final Model as follows:

lmpg ~ displacement + horsepower + weight + acceleration + cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + horsepower:cy\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration + acceleration:or\_dum1 + horsepower:or\_dum2 + acceleration:cy\_dum1 + weight:or\_dum2 + displacement:cy\_dum2 + weight:acceleration

summary(second\_order\_model\_log\_outlier2)

##   
## Call:  
## lm(formula = lmpg ~ displacement + horsepower + weight + acceleration +   
## cy\_dum1 + cy\_dum2 + or\_dum1 + or\_dum2 + displacement:cy\_dum1 +   
## horsepower:cy\_dum1 + acceleration:cy\_dum2 + acceleration:or\_dum1 +   
## horsepower:acceleration + horsepower:or\_dum2 + displacement:or\_dum2 +   
## acceleration:cy\_dum1 + horsepower:or\_dum1, data = df\_mod\_4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.40749 -0.07877 -0.00512 0.08356 0.70327   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.736e+00 1.736e-01 21.526 < 2e-16 \*\*\*  
## displacement 5.249e-04 5.036e-04 1.042 0.297962   
## horsepower 2.996e-03 1.972e-03 1.519 0.129550   
## weight -1.061e-04 3.412e-05 -3.108 0.002036 \*\*   
## acceleration 1.578e-02 1.089e-02 1.450 0.148004   
## cy\_dum1 -6.269e-01 2.803e-01 -2.236 0.025949 \*   
## cy\_dum2 -1.194e+00 2.043e-01 -5.846 1.14e-08 \*\*\*  
## or\_dum1 -1.959e-01 2.100e-01 -0.933 0.351546   
## or\_dum2 1.988e-01 9.847e-02 2.019 0.044269 \*   
## displacement:cy\_dum1 -2.158e-03 6.969e-04 -3.096 0.002114 \*\*   
## horsepower:cy\_dum1 4.998e-03 1.378e-03 3.627 0.000328 \*\*\*  
## acceleration:cy\_dum2 6.323e-02 1.307e-02 4.839 1.94e-06 \*\*\*  
## acceleration:or\_dum1 1.936e-02 9.087e-03 2.130 0.033832 \*   
## horsepower:acceleration -5.434e-04 1.334e-04 -4.073 5.71e-05 \*\*\*  
## horsepower:or\_dum2 -6.318e-03 1.483e-03 -4.260 2.62e-05 \*\*\*  
## displacement:or\_dum2 3.321e-03 1.077e-03 3.083 0.002210 \*\*   
## acceleration:cy\_dum1 2.411e-02 1.156e-02 2.085 0.037739 \*   
## horsepower:or\_dum1 -1.985e-03 1.178e-03 -1.686 0.092650 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1333 on 358 degrees of freedom  
## Multiple R-squared: 0.8503, Adjusted R-squared: 0.8432   
## F-statistic: 119.6 on 17 and 358 DF, p-value: < 2.2e-16

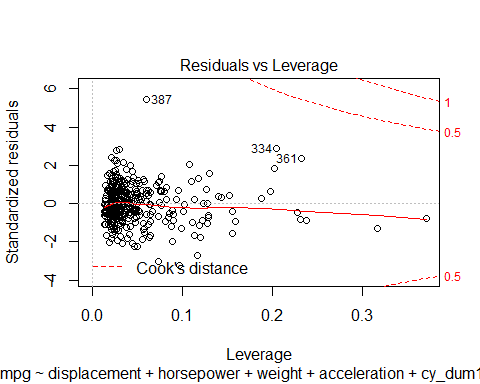
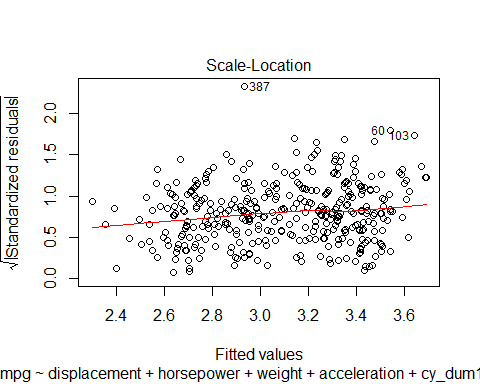
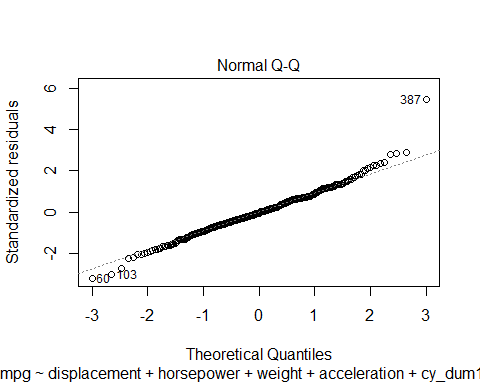
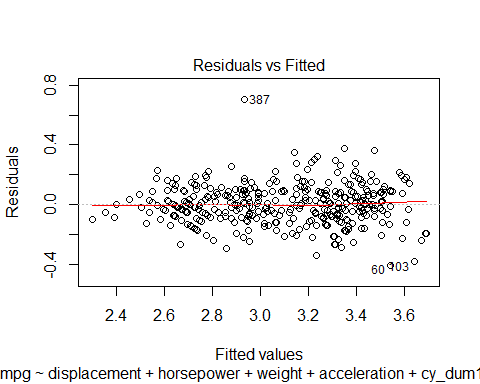
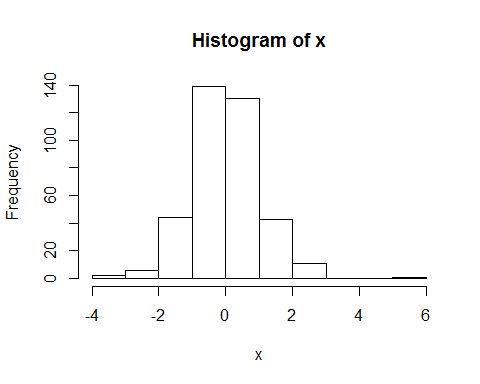
If our analysis was based SOLELY on Akaike Information Criteria, we would stop the Analysis here and let the insignificant terms remain.

Diagnostics

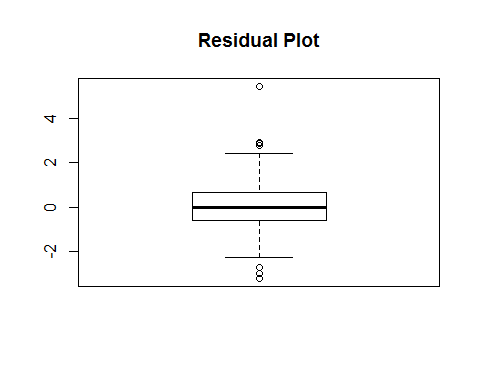
***Carry out a detailed diagnostic check through residual analysis of the fitted model, ensuring that the data fits the model well, conforming to all the underlying model assumptions.***

We have performed diagnostics (Residual Analysis) for the previous models in the previous section. We reproduce the Residual Analysis here again.

resAnalysis(second\_order\_model\_log\_outlier2)



## Method P.value  
## 1 Shapiro-Wilk normality test 0.0000548181  
## 2 Anderson-Darling normality test 0.0320187362  
## 3 Cramer-von Mises normality test 0.0445774665  
## 4 Lilliefors (Kolmogorov-Smirnov) normality test 0.0583250224  
## 5 Shapiro-Francia normality test 0.0000329422  
##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 29.762, df = 17, p-value = 0.02811  
##   
## [1] "Outliers in the data :"



## 60 103 112 310 323 334 387   
## -3.215913 -3.007434 -2.733877 2.856266 2.776920 2.910071 5.442648

As can be seen from the Normality Tests, the Normality Assumption is valid. The results of Breusch-Pagan Test shows that the residuals are not strictly homoscedastic. We take a look at the Residual Plot and we believe there is no visible pattern so we choose to accept the Model (despite it failing the Homoscedasticity Assumption).

(We believe that applying a different transformation like Box-Cox or successively removing outliers from the given data, we would be able to achieve Homoscedasticity in residuals.)

Conclusion

***Based on the fitted model and your graphical (as well as subjective engineering) understanding, explain how the different independent variables considered in this study, are “affecting” the gas mileage of different cars.***

The proposed model based on AIC :

lmpg ~ displacement + horsepower + weight + acceleration + cy\_dum1 + cy\_dum2 + or\_dum1 + horsepower:cy\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration + acceleration:or\_dum1 + displacement:cy\_dum2 + acceleration:cy\_dum1 + weight:cy\_dum2

The proposed model based on AIC after removing Insignificant terms :

lmpg ~ displacement + horsepower + cy\_dum1 + cy\_dum2 + or\_dum1 + horsepower:cy\_dum1 + acceleration:cy\_dum2 + horsepower:acceleration + acceleration:or\_dum1 + displacement:cy\_dum2 + acceleration:cy\_dum1

log(mpg) = 4.175 - 0.002(displacement) + 0.003(horsepower) - 0.988(cy\_dum1) - 1.581(cy\_dum2) - 0.672(or\_dum1) + 0.0043(horsepower x cy\_dum1) + 0.071(cy\_dum2 x acceleration) - 0.0005 (horsepower x acceleration) + 0.035(or\_dum1 x acceleration) + 0.002(displacement x cy\_dum2) + 0.034 (cy\_dum1 x acceleration)

We can make the following inferences from the Model we have :

1. Displacement has a linear effect on Log mpg values keeping in mind the linear effect of other terms.
2. Horsepower has a linear effect on Log mpg values keeping in mind the linear effect of other terms.
3. Each of the Cylinder values (4,6,8) has a different linear effect on log mpg values keeping in mind the linear effect of the other terms.
4. If the country of origin is Europe, the intercept varies for log mpg, keeping in mind the linear effects of the other terms.
5. The interaction between horsepower and number of cylinders implies that based on the number of cylinders, horsepower affects log mpg differently.
6. The interaction between acceleration and number of cylinders implies that based on the number of cylinders, the acceleration of a given car affects log mpg values differently.

7.The interaction between horsepower and acceleration implies that for different values of horsepower, acceleration affects the log mpg values differently.

1. The interaction between acceleration and country of origin implies that based on different countries of origin, the acceleration affects the log mpg values differently.
2. The interaction between displacement and number of cylinders implies that based on the number of cylinders, different number of cylinders affects how displacement values affect log mpg values.
3. Based on the interaction between number of cylinders and acceleration, we can claim that different number of cylinders causes acceleration values to affect log mpg values differently.

***Akaike Information Criteria***

Short Note on Step AIC algorithm and why it selects the best possible Model

AIC is a method that combines a measure of model fit with a measure of model Complexity. It can be written as :

AIC = -2logL + 2p

L = Maximum Likelihood of the data using the Model p = Number of parameters in the Model

Here, -2logL is a function of the prediction Error. It measures how the Model fits the data. The smaller the value, the better the fit.

2p term penalizes more complex models so again, the smaller the value of this term, the better.

Stepwise Selection with AIC

The aim is to look for a model with the smallest AIC:

1. start with some model, simple or complex
2. do a forward step as well as a backward step based on AIC until no predictor should be added, and no predictor should be removed

The stepAIC() function in R can be used to perform this. All arguments in the stepAIC() function are set to default. If you want to set direction of stepwise regression (e.g., backward, forward, both), the direction argument should be assigned. The stepAIC() function also allows specification of the range of variables to be included in the model by using the scope argument. The lower model is the model with smallest number of variables and the upper model is the largest possible model. Both upper and lower components of scope can be explicitly specified. If scope is a single formula, it specifies the upper component, and the lower model is empty. If scope is missing, the initial model is the upper model. Recall that “^” symbol denotes interactions up to a specified degree. In our case, we specified two-degree interactions among all possible combinations of variables. Elements within I() are interpreted arithmetically.