Multiple Regression US Automobile MPG Prediction:

Dataset: This file contains information on 383 automobiles marketed in the United States between 1970 and 1982. The variables included miles per gallon (MPG), cylinders in the engine, engine displacement in cubic inches, horsepower, weight (in pounds), model year, country of origin, and auto make, and auto model.

Problem Statement: Using a random number seed and the random selection method, take a random sample of 80 automobiles from the full data set. It will have the following characteristics.

1. The "year" and "cylinders" variables will have been converted to factors.
2. Only autos from 1976 and earlier should be included.

Build an Advanced Multiple Linear Regression Model in R and answer the following questions.

Analysis

1. Conduct a multiple regression analysis using the data in the reduced data set. Use MPG as the dependent variable and cubic inches, horsepower, and weight as independent variables.
2. Show the model output with appropriate discussion of the p value for each beta coefficient (including β_0). Give proper written interpretations of the beta coefficients which explain the variable's estimated impact on the y.
3. Report and interpret confidence interval for each beta coefficient in the model.
4. Determine and state whether the model appears to be in conformity with the LINE assumptions of regression. Show appropriate graphics and give written interpretations where needed to justify the conclusions.
5. Determine whether any of the data points in the reduced data set have a high leverage in influencing the plot of the regression. Show appropriate graphics. If there are high-leverage data points report ONLY the year, make, and model of these autos.
6. Introduce squared terms into the model which are based on the horsepower and weight variables. Does using either or both these squared terms improve the fit of the model? Explain reasoning on this point.
7. Using the original multiple regression model above, introduce the "cylinders" factor variable to the model. Do any of the factor levels appear to contribute to the fit of the original model?

I. Preprocessing

```
#Author: Suryateja Chalapati

#Importing required libraries
rm(list=ls())
library(rio)
library(moments)
library(dplyr)

#Setting the working directory and importing the dataset
setwd("C:/Users/surya/Downloads")

df = import("MPG Data.xlsx", sheet = "6304 Old Auto MPG")
colnames(df)=tolower(make.names(colnames(df)))
attach(df)

#Assigning factor variable & subsetting
df$cylinders = as.factor(df$cylinders)
str(df)
```

```
## 'data.frame':    383 obs. of  9 variables:
##  $ mpg         : num  39.4 25 15 22 36 29 44.3 26 15 11 ...
##  $ cylinders   : Factor w/ 3 levels "4","6","8": 1 2 3 2 1 1 1 1 3 3 ...
##  $ cubic.inches: num  85 181 429 232 120 97 90 97 350 318 ...
##  $ horsepower  : num  70 110 198 112 88 75 48 78 145 210 ...
##  $ weight      : num  2070 2945 4341 2835 2160 ...
##  $ year        : num  78 82 70 82 82 75 80 74 75 70 ...
```

```
##  $ origin      : chr  "Japan" "USA" "USA" "USA" ...
##  $ make        : chr  "datsun" "buick" "ford" "ford" ...
##  $ model       : chr  "b210 gx" "century limited" "galaxie 500" "granada l" ...
```

```
new_sample = subset(df, year <= 76)
new_sample$year = as.factor(new_sample$year)

str(new_sample)
```

```
## 'data.frame':    211 obs. of  9 variables:
##  $ mpg         : num  15 29 26 15 11 11 18.5 18 32 21 ...
##  $ cylinders   : Factor w/ 3 levels "4","6","8": 3 1 1 3 3 3 2 3 1 2 ...
##  $ cubic.inches: num  429 97 97 350 318 429 250 307 71 155 ...
##  $ horsepower  : num  198 75 78 145 210 208 110 130 65 107 ...
##  $ weight      : num  4341 2171 2300 4440 4382 ...
##  $ year        : Factor w/ 7 levels "70","71","72",..: 1 6 5 6 1 3 7 1 5 4 ...
##  $ origin      : chr  "USA" "Japan" "Europe" "USA" ...
##  $ make        : chr  "ford" "toyota" "opel" "chevrolet" ...
##  $ model       : chr  "galaxie 500" "corolla" "manta" "bel air" ...
```

```
#Setting seed and data sampling
set.seed(36991670)
df_sample = data.frame(new_sample[sample(1:nrow(new_sample), 80, replace = FALSE),])
attach(df_sample)
```

II. Analysis

```
#Analysis_[1,2]
#Multiple Regression
lin_reg=lm(mpg ~ cubic.inches + horsepower + weight, data=df_sample)
summary(df_sample$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   14.00   18.75   19.43   24.00   33.00
```

```
summary(lin_reg)
```

```
##
## Call:
## lm(formula = mpg ~ cubic.inches + horsepower + weight, data = df_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2680 -1.5547 -0.0751  1.0850  5.6347
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.2332830  1.3513234  26.813  < 2e-16 ***
## cubic.inches -0.0137824  0.0064900  -2.124    0.037 *
## horsepower   -0.0101901  0.0132549  -0.769    0.444
## weight       -0.0039408  0.0006991  -5.637  2.8e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 76 degrees of freedom
## Multiple R-squared:  0.8274, Adjusted R-squared:  0.8206
## F-statistic: 121.4 on 3 and 76 DF,  p-value: < 2.2e-16
```

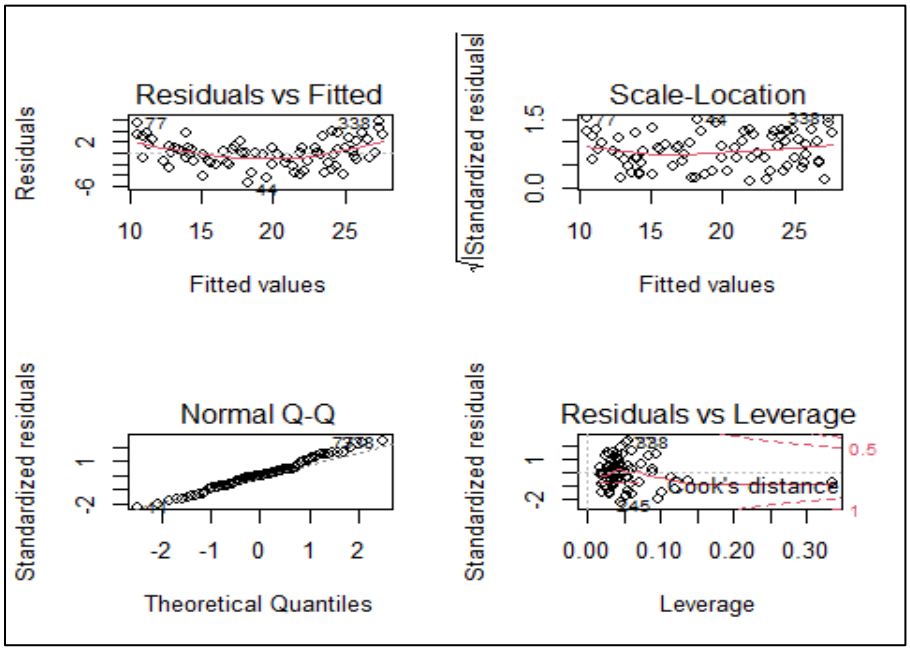
- From the data in the "Cylinders" column, it is a factor variable and R treats it as such.

- R treats the "Year" column as numeric. So, I have converted the column into factor.
- Considering only the models from 1976 and earlier, we procced with doing a multiple regression.

From the Multiple Regression analysis:

- The estimated B – Coefficients for the values are small.
- Except for horsepower (p-value = 0.44, Failed to reject the null, B – Coefficient could be zero), the p-value is < 0.05 for displacement and weight. This states we can reject the null hypothesis and accept the alternate, so we can say the B – Coefficients are not zero and are significant.
- The Regression Equation is **y [MPG] = 36.233 – 0.013*Cubic_Inches – 0.010*Year – 0.003*Cylinders**.
- The R-sq is 0.8274, that means all the X variables explains about 82.74% of variation in Y. Based on this, we can say the model is a good measure of Miles Per Gallon based on the variables we have taken for our analysis.
- We can say there is an inverse proportionality between [Weight, MPG] & [Displacement, MPG].
- All other variables being constant, a 1000 pounds increase in weight could lead to a decrease of mileage by about 3MPG.
- All others being constant, a 100 cubic inches increase in displacement could lead to a decrease of mileage by 1.37MPG.
- All others being constant, a 100 horsepower increase could lead to a decrease of mileage by 1.01MPG. But the value is not statistically significant because of the p-value.
- If displacement, horsepower and weight is zero then MPG increases by 36.23MPG, which is something we cannot interpret.

```
layout(matrix(c(1,2,3,4),2,2)) # optional 4 graphs/page
plot(lin_reg)
```



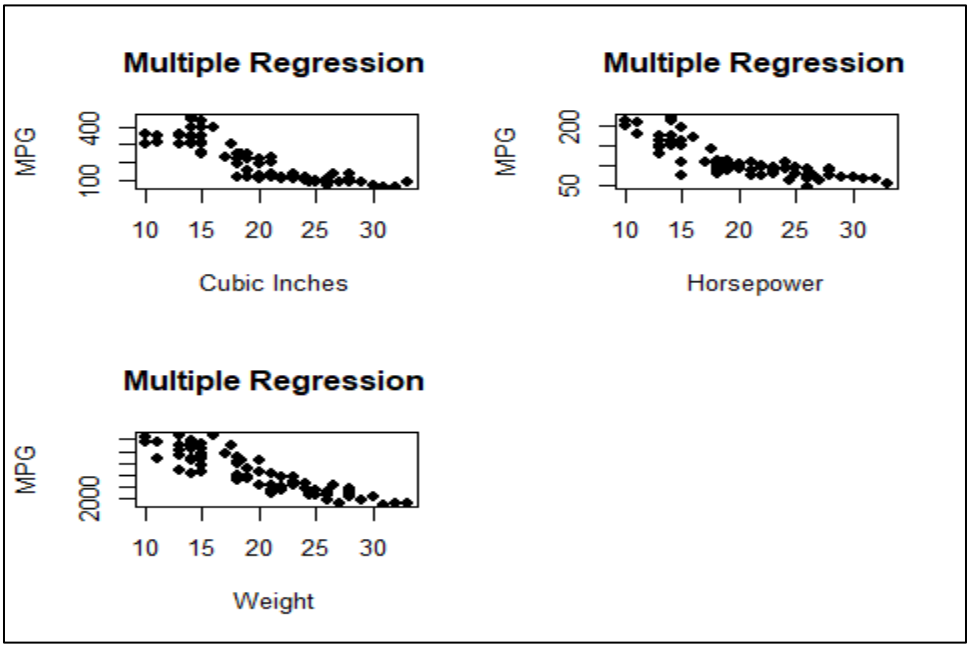

```
par(mfrow=c(1,1))

#Analysis_3
confint(lin_reg)
```

```
##                     2.5 %        97.5 %
## (Intercept)  33.541889175 38.9246768211
## cubic.inches -0.026708330 -0.0008565268
## horsepower   -0.036589543  0.0162093489
## weight       -0.005333164 -0.0025484511
```

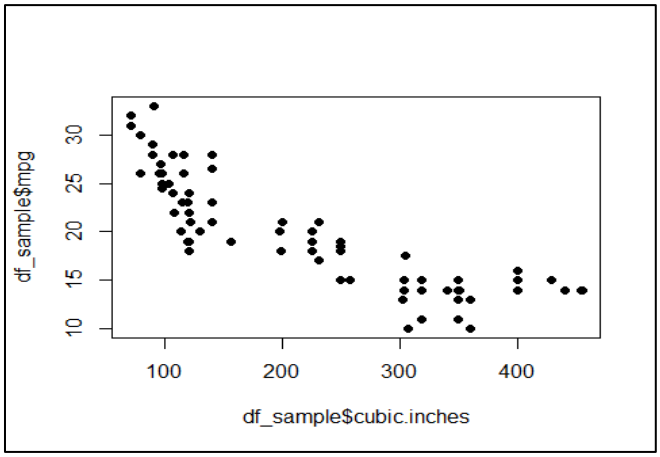
We are 95% confident the B – Coefficients can lie anywhere in the following ranges:

- The B – Coefficient of intercept lies between [33.54, 38.92], a fairly tight CI.
- The B – Coefficient of displacement lies between [-2.6, -0.08], scaled for 100 cubic inches of displacement. Still a tight CI.
- The B – Coefficient of horsepower lies between [-3.6, 1.6], scaled for change in 100 horsepower. Still a tight CI.
- The B – Coefficient of weight lies between [-5, -2.5], scaled for 1000 pounds change of weight. Again a tight CI.

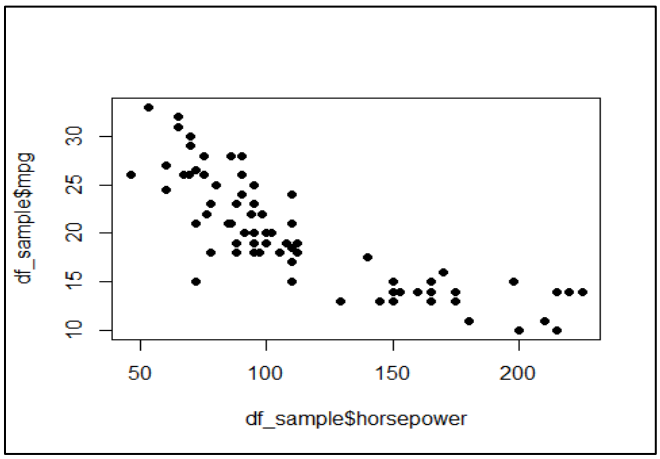
```
par(mfrow=c(2,2))
plot(df_sample$mpg, df_sample$cubic.inches,
     pch=19,main="Multiple Regression",xlab="Cubic Inches",ylab="MPG")
plot(df_sample$mpg, df_sample$horsepower,
     pch=19,main="Multiple Regression",xlab="Horsepower",ylab="MPG")
plot(df_sample$mpg, df_sample$weight,
     pch=19,main="Multiple Regression",xlab="Weight",ylab="MPG")
par(mfrow=c(1,1))
```



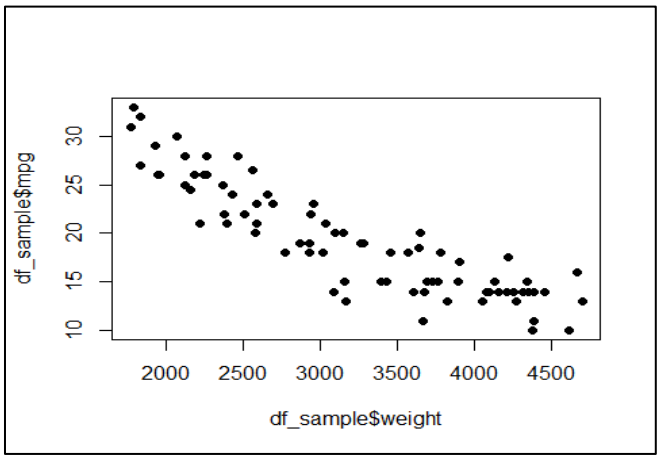

```
#Analysis_4
#Linearity
plot(df_sample$cubic.inches, df_sample$mpg, pch=19)
```



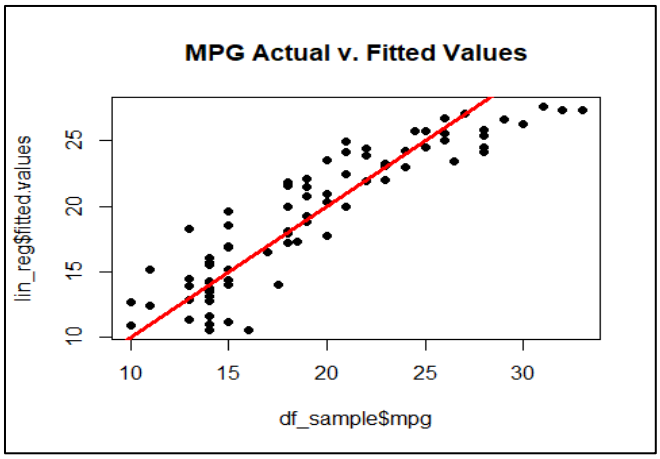

```
plot(df_sample$horsepower, df_sample$mpg, pch=19)
```




```
plot(df_sample$weight, df_sample$mpg, pch=19)
```

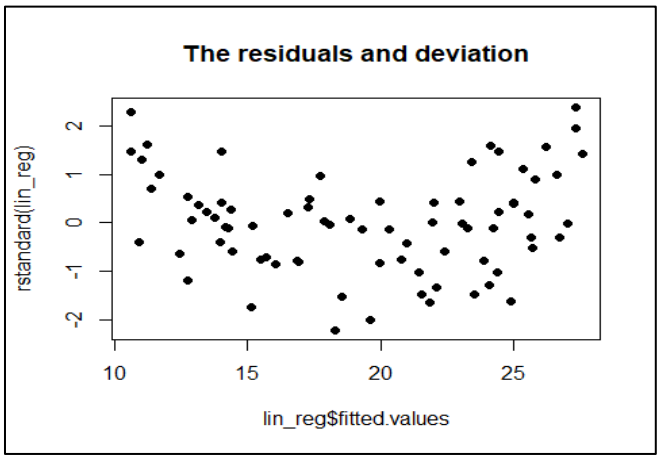



```
plot(df_sample$mpg, lin_reg$fitted.values, pch=19, main="MPG Actual v. Fitted Values")
abline(0, 1, col="red", lwd=3)
```



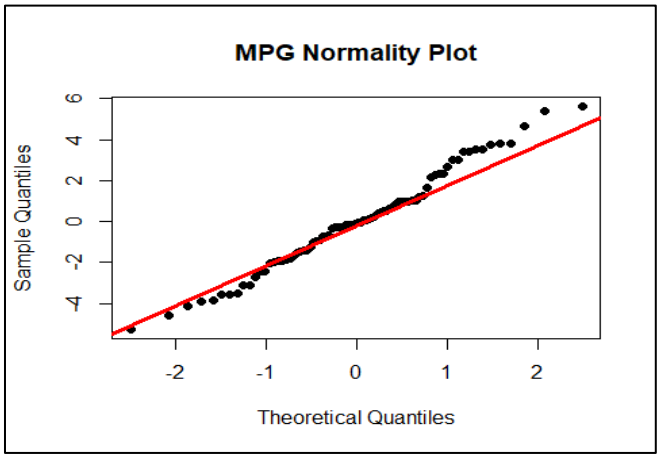
- Based on the scatter plots, displacement and horsepower seems to follow a curvilinear relationship. But weight does have a linear relationship with MPG.

```
#Independence
plot(lin_reg$fitted.values, rstandard(lin_reg), pch=19, main="The residuals and deviation")
```



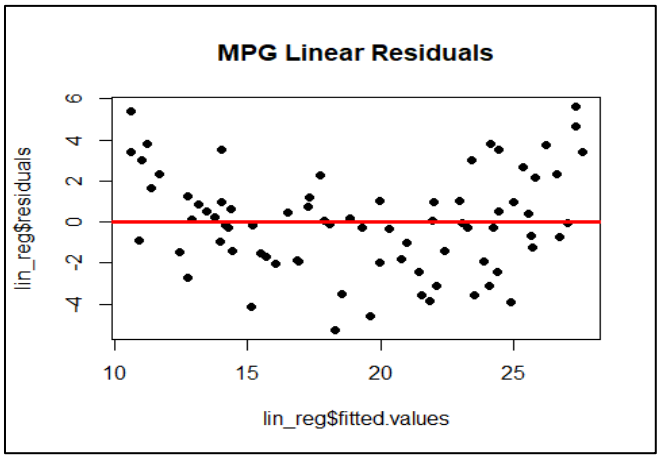
- The fitted vales with their residuals show a U – Shaped pattern. So, it fails the independence test.

```
#Normality
qqnorm(lin_reg$residuals, pch=19, main="MPG Normality Plot")
qqline(lin_reg$residuals, col="red", lwd=3)
```



- From the plot the data almost follows a normal distribution but there are few values out in the tails.

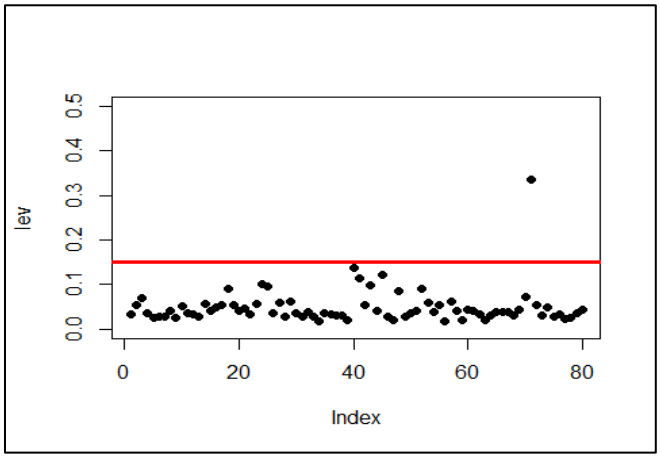
```
#Equality of Variances
plot(lin_reg$fitted.values, lin_reg$residuals, pch=19, main="MPG Linear Residuals")
abline(0, 0, col="red", lwd=3)
```



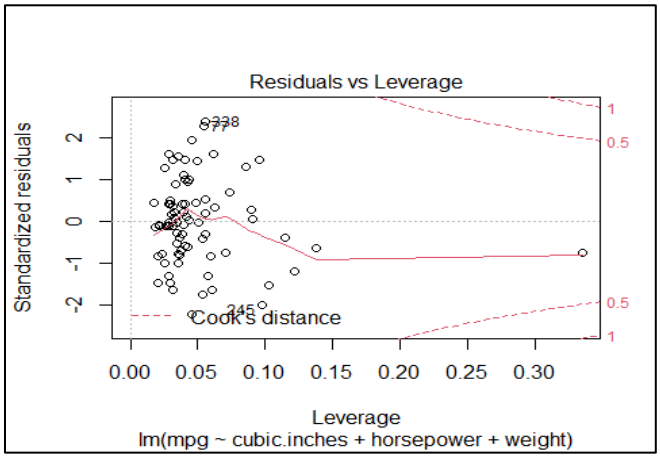
- The fitted vales with their residuals show a U – Shaped pattern. We can say the data is biased and follows homoscedasticity. So, it fails the equality of variance test.

```
#Analysis_5
#Identifying high leverage points.
```

```
lev=hat(model.matrix(lin_reg))
plot(lev, pch=19, ylim=c(0,0.5))
abline(3*mean(lev), 0, col="red", lwd=3)
```




```
plot(lin_reg, which=5)
```




```
lev_points = lev[lev > 3*mean(lev)]
loop = match(c(lev_points), lev)
for (i in loop){
  print(paste0("Make is ", df_sample$make[i], ", Model is ", df_sample$model[i], " & year is "
, df_sample$year[i]))
}
```

```
## [1] "Make is buick, Model is estate wagon (sw) & year is 70"
```

- There is one leverage point that which is as shown above. Getting rid of it really does not add much to the overall result.

```
#Getting rid of the high leverage points
no_lev=df_sample
new_lev=df_sample[lev>(3*mean(lev)),1]
no_lev=no_lev[-new_lev,]

lin_reg=lm(mpg~cubic.inches+horsepower+weight, data=df_sample)
summary(lin_reg)
```

```
##
## Call:
## lm(formula = mpg ~ cubic.inches + horsepower + weight, data = df_sample)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -5.2680 -1.5547 -0.0751  1.0850  5.6347
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.2332830  1.3513234  26.813  < 2e-16 ***
## cubic.inches -0.0137824  0.0064900  -2.124    0.037 *
## horsepower   -0.0101901  0.0132549  -0.769    0.444
## weight       -0.0039408  0.0006991  -5.637  2.8e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 76 degrees of freedom
## Multiple R-squared:  0.8274, Adjusted R-squared:  0.8206
## F-statistic: 121.4 on 3 and 76 DF,  p-value: < 2.2e-16
```

```
lin_reg2=lm(mpg~cubic.inches+horsepower+weight, data=no_lev)
summary(lin_reg2)
```

```
##
## Call:
## lm(formula = mpg ~ cubic.inches + horsepower + weight, data = no_lev)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2010 -1.5482 -0.1175  1.1180  5.6337
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.1206336  1.3471153  26.813  < 2e-16 ***
## cubic.inches -0.0155874  0.0065974  -2.363   0.0207 *
## horsepower   -0.0073699  0.0133562  -0.552   0.5827
## weight       -0.0038692  0.0006976  -5.546 4.18e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.412 on 75 degrees of freedom
## Multiple R-squared:  0.8314, Adjusted R-squared:  0.8246
## F-statistic: 123.3 on 3 and 75 DF,  p-value: < 2.2e-16
```

```
#Analysis_6
Feat_Lev = lm(mpg ~ cubic.inches + horsepower + poly(weight,2), data = df_sample)
summary(Feat_Lev)
```

```
##
## Call:
## lm(formula = mpg ~ cubic.inches + horsepower + poly(weight, 2),
##     data = df_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2858 -1.4458  0.2901  1.2139  4.3345
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       24.448762   1.154505  21.177  < 2e-16 ***
## cubic.inches      -0.009831   0.005506  -1.785   0.0782 .
## horsepower        -0.024366   0.011431  -2.132   0.0363 *
## poly(weight, 2)1 -28.729620   4.465592  -6.434 1.05e-08 ***
```

```
## poly(weight, 2)2  11.882070   2.090730   5.683 2.39e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.04 on 75 degrees of freedom
## Multiple R-squared:  0.8793, Adjusted R-squared:  0.8729
## F-statistic: 136.7 on 4 and 75 DF,  p-value: < 2.2e-16
```

```
Feat_Lev = lm(mpg ~ cubic.inches + weight + poly(horsepower,2), data = df_sample)
summary(Feat_Lev)
```

```
##
## Call:
## lm(formula = mpg ~ cubic.inches + weight + poly(horsepower, 2),
##     data = df_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1845 -1.9290  0.1171  1.4241  5.3181
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          33.3470736  2.0407991  16.340  < 2e-16 ***
## cubic.inches         -0.0155846  0.0064911  -2.401   0.0188 *
## weight               -0.0032808  0.0007887  -4.160  8.4e-05 ***
## poly(horsepower, 2)1 -6.5946639  5.4541506  -1.209   0.2304
## poly(horsepower, 2)2  4.8053244  2.7803770   1.728   0.0880 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.393 on 75 degrees of freedom
## Multiple R-squared:  0.834,  Adjusted R-squared:  0.8251
## F-statistic:  94.2 on 4 and 75 DF,  p-value: < 2.2e-16
```

```
Feat_Lev = lm(mpg ~ cubic.inches + poly(horsepower,2) + poly(weight,2), data = df_sample)
summary(Feat_Lev)
```

```
##
## Call:
## lm(formula = mpg ~ cubic.inches + poly(horsepower, 2) + poly(weight,
##     2), data = df_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4276 -1.4111  0.2063  1.2856  4.3255
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           21.395921   1.288708  16.603  < 2e-16 ***
## cubic.inches          -0.008848   0.005694  -1.554   0.1244
## poly(horsepower, 2)1  -9.174871   4.690158  -1.956   0.0542 .
## poly(horsepower, 2)2  -1.917647   2.690915  -0.713   0.4783
## poly(weight, 2)1     -30.650062   5.228325  -5.862 1.19e-07 ***
## poly(weight, 2)2      12.673471   2.373454   5.340 9.84e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.047 on 74 degrees of freedom
```

```
## Multiple R-squared:  0.8802, Adjusted R-squared:  0.8721
## F-statistic: 108.7 on 5 and 74 DF,  p-value: < 2.2e-16
```

- From the little detective work we did above, weight (Squared term) seems to improve the overall fit of the model. Having said that, both horsepower and weight (Squared terms) seem to improve the fit even more at R-sq = 88% but overall weight seems to be more linearly related to MPG.

```
#Analysis_7
lin_reg3 = lm(mpg ~ cubic.inches + horsepower + weight + cylinders, data = df_sample)
summary(lin_reg3)

##
## Call:
## lm(formula = mpg ~ cubic.inches + horsepower + weight + cylinders,
##     data = df_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7950 -1.4452  0.0484  1.1944  5.3962
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  34.9198179  1.7213025  20.287  < 2e-16 ***
## cubic.inches  0.0017037  0.0094981   0.179   0.8581
## horsepower   -0.0304704  0.0160839  -1.894   0.0621 .
## weight       -0.0032625  0.0007274  -4.485 2.62e-05 ***
## cylinders6   -3.3707386  1.3387373  -2.518   0.0140 *
## cylinders8   -3.3698280  2.1153445  -1.593   0.1154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.351 on 74 degrees of freedom
## Multiple R-squared:  0.8419, Adjusted R-squared:  0.8312
## F-statistic: 78.82 on 5 and 74 DF,  p-value: < 2.2e-16

levels(df_sample$cylinders)

## [1] "4" "6" "8"
```

- Cylinders have three levels. Adding this factor to the equation the R-sq has increased from 82% to 84% (Significant enough).
- Out of the three levels, 6-Cylinder model seems to contribute to the original model.