Fuel Efficiency Prediction: Linear Regression

By Vishwa Pardeshi

In this notebook, we will predict the fuel consumption rate of cars using the autompg dataset which is available on the popular UCI machine learning repository. To learn more about linear regression refer here.

Learning Outcome: By following the notebook you will be able to

1. Perform context inspired EDA to understand relationship between predictor variables and mpg (fuel consumption variable)
2. Implement & infer Simple Linear Regression
3. Plot OLS regression line & diagnostic plot
4. Implement & infer multiple linear regression model
5. Integrate interaction effects in the model
6. Implement non-linear transformation to the model

Setup

Here, we will load the libraries and packages.

```
library(MASS)
library(ISLR)
library(tidyverse)
library(ggplot2)
library(corrplot)
```

The *Auto* dataset is present in the ISLR package and hence can be called directly

Exploratory Data Analysis

Glimpse of the data

Let us first look at the head of the dataset.

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```r
cat("The auto dataset shape is ", dim(Auto)[1], "x", dim(Auto)[2])
```

```
## The auto dataset shape is  392 x 9
```

We notice tht there are 9 columns. Out of which a few look categorical such as model year and origin. Additionally, name is a string. Let's explore the summary of the data to undertand the data type of variables.

```r
summary(Auto)
```

```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##
##      weight      acceleration        year           origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
##
##                 name
##  amc matador       :  5
##  ford pinto        :  5
##  toyota corolla    :  5
##  amc gremlin       :  4
##  amc hornet        :  4
##  chevrolet chevette:  4
##  (Other)           :365
```

From summary we notice that origin, and cylinder and year could be categorical variables as they seem to be whole numbers with a limited range. Looking at the data variable this is not evident as they are all numeric value.

```r
str(Auto)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  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 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241
```

We notice that all the variables have the correct data type except name which can be converted to string.

Number of unique values in each column

Let's check out the number of unique values in each column.

```
no_unique_values <- function(column){
  unique_values_list <- unique(column)
  return(length(unique_values_list))
}
apply(Auto ,2,no_unique_values)
```

```
##          mpg    cylinders displacement   horsepower       weight
##          127            5           81           93          346
## acceleration         year       origin         name
##           95           13            3          301
```

Thus, We have established that: * mpg: continuous

- cylinders: multi-valued discrete
- displacement: continuous
- horsepower: continuous
- weight: continuous
- acceleration: continuous
- model year: multi-valued discrete
- origin: multi-valued discrete
- car name: string (unique for each instance)

Number of missing values in the dataset

```
no_missing_values <- function(column){
  return(sum(is.na(column)))
}

cat("The number of missing values in each column\n\n\n")
```

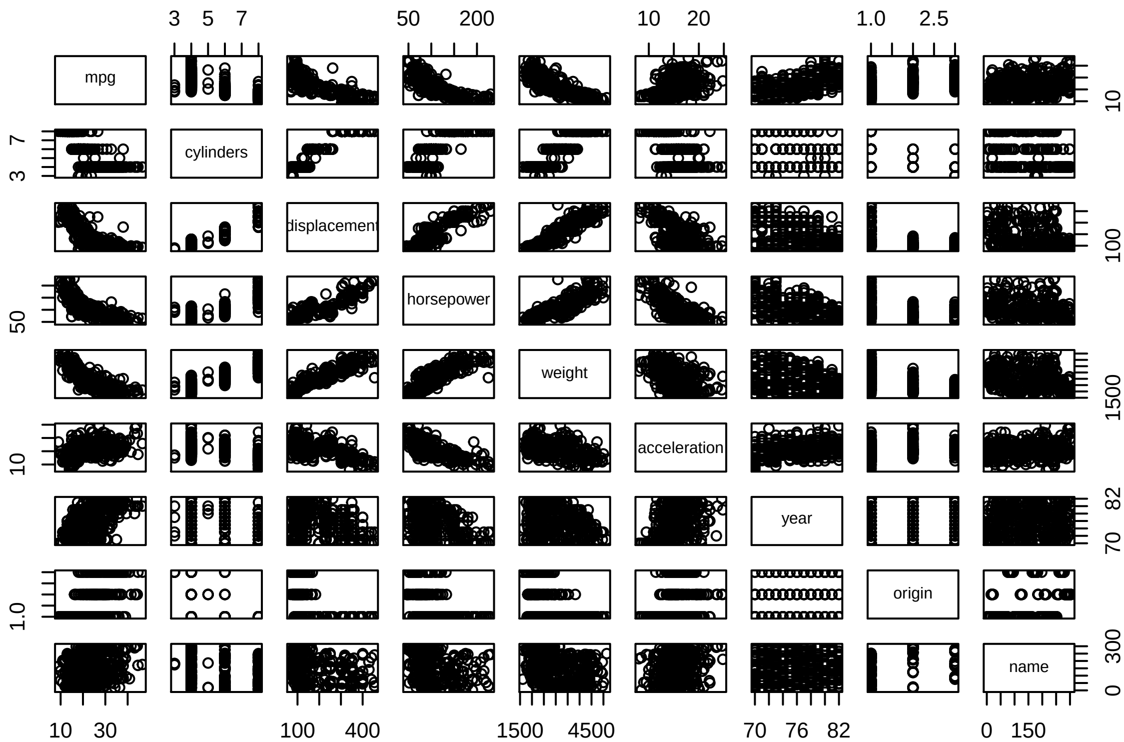
```
## The number of missing values in each column
```

```
apply(Auto ,2,no_missing_values)
```

```
##          mpg    cylinders displacement   horsepower       weight
##            0            0            0            0            0
## acceleration         year       origin         name
##            0            0            0            0
```

Scatterplot Matrix

```r
plot(Auto)
```



Viewing the scatterplot alongwith the summary makes our observation about the nature of the variable more clear. From the scatterplot we can also notice the relationship trend between our response variable mpg and the other potential predictor variable.

We notive that horsepower, weight, displacement have a clear relationship with mpg though not necessarily linear

Correlation Matrix

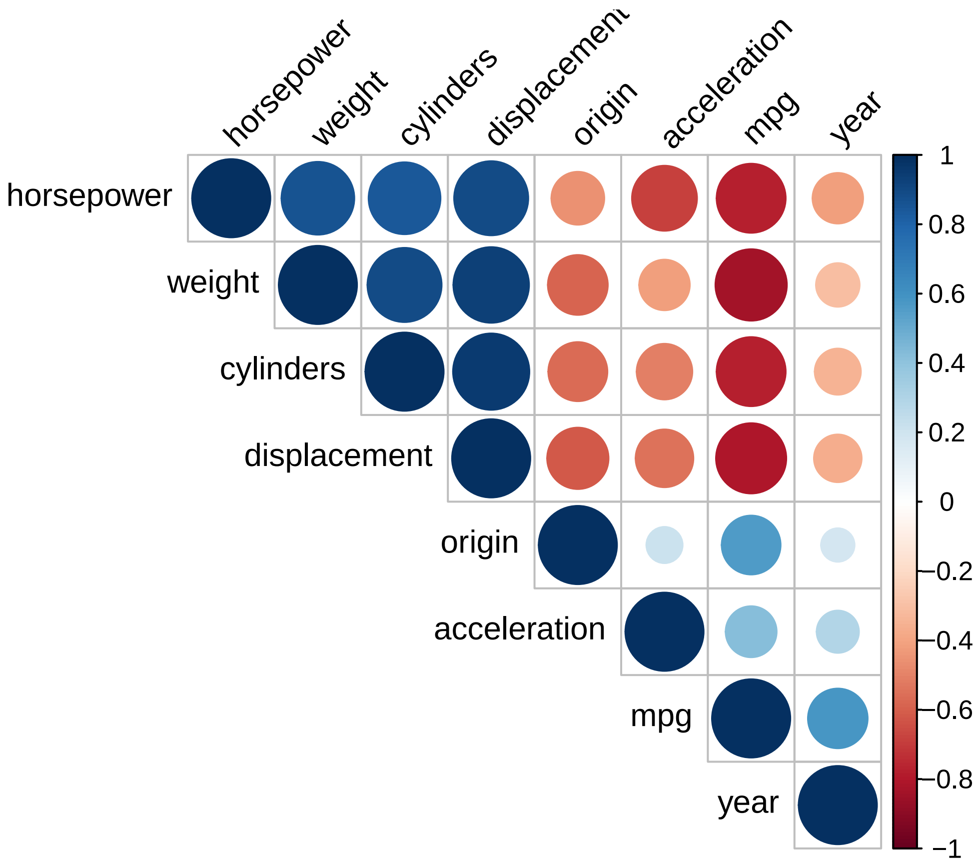
```r
corr_matrix <- cor(Auto[, -which(names(Auto)=="name")])
corr_matrix
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
```

```
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

Heatmap of Correlation

```r
corrplot(corr_matrix, type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)
```



We notice here that mpg has significant correlation with 1. horsepower

2. weight
3. cylinders
4. displacement
5. year

Clean Data

Convert to categorical form i.e. factor

```
discrete_col <- c("origin", 'year', 'cylinders')
Auto[, discrete_col] <- lapply(Auto[, discrete_col], factor)
str(Auto[, discrete_col])
```

```
## 'data.frame':    392 obs. of  3 variables:
##  $ origin   : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ year     : Factor w/ 13 levels "70","71","72",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ cylinders: Factor w/ 5 levels "3","4","5","6",..: 5 5 5 5 5 5 5 5 5 5 ...
```

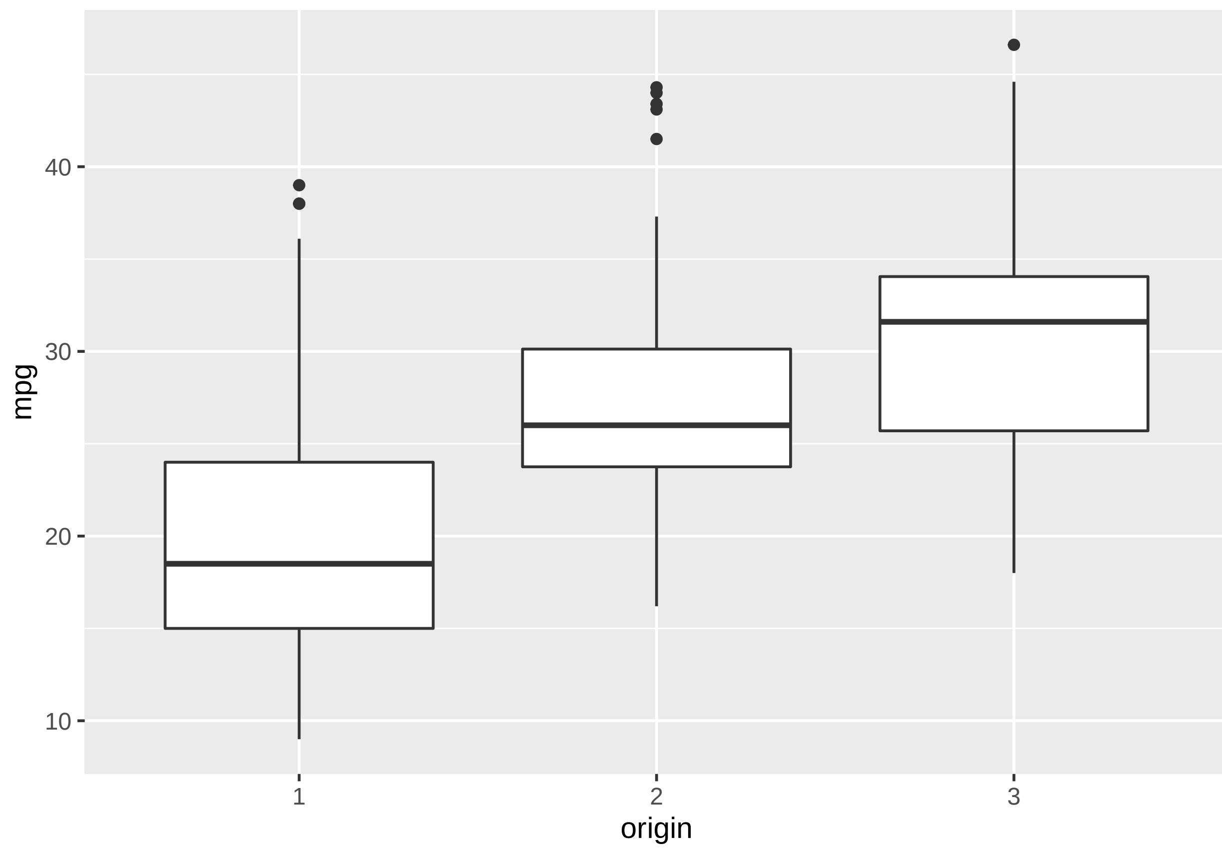
Now, we can visualize the relationship between the response and other variables

Explore relationship of mpg with discrete variable

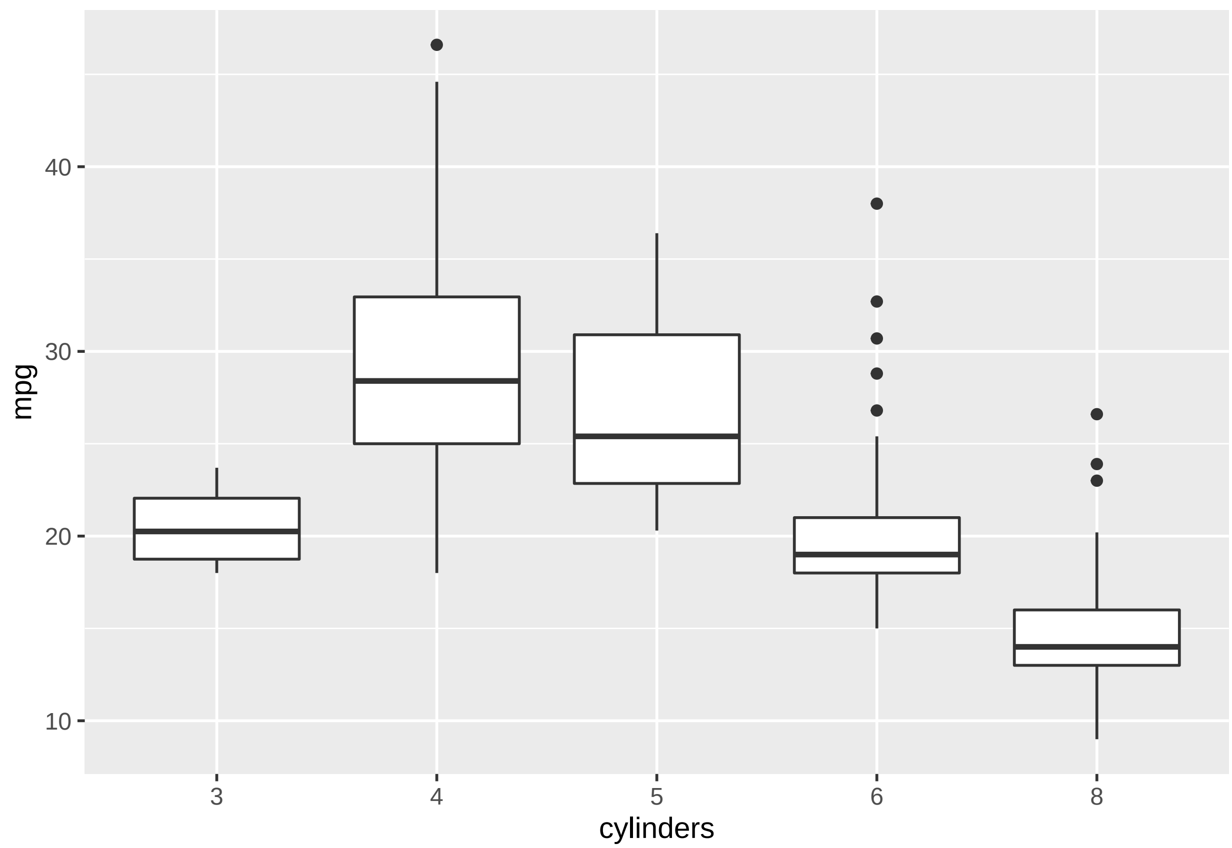
```
attach(Auto)
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

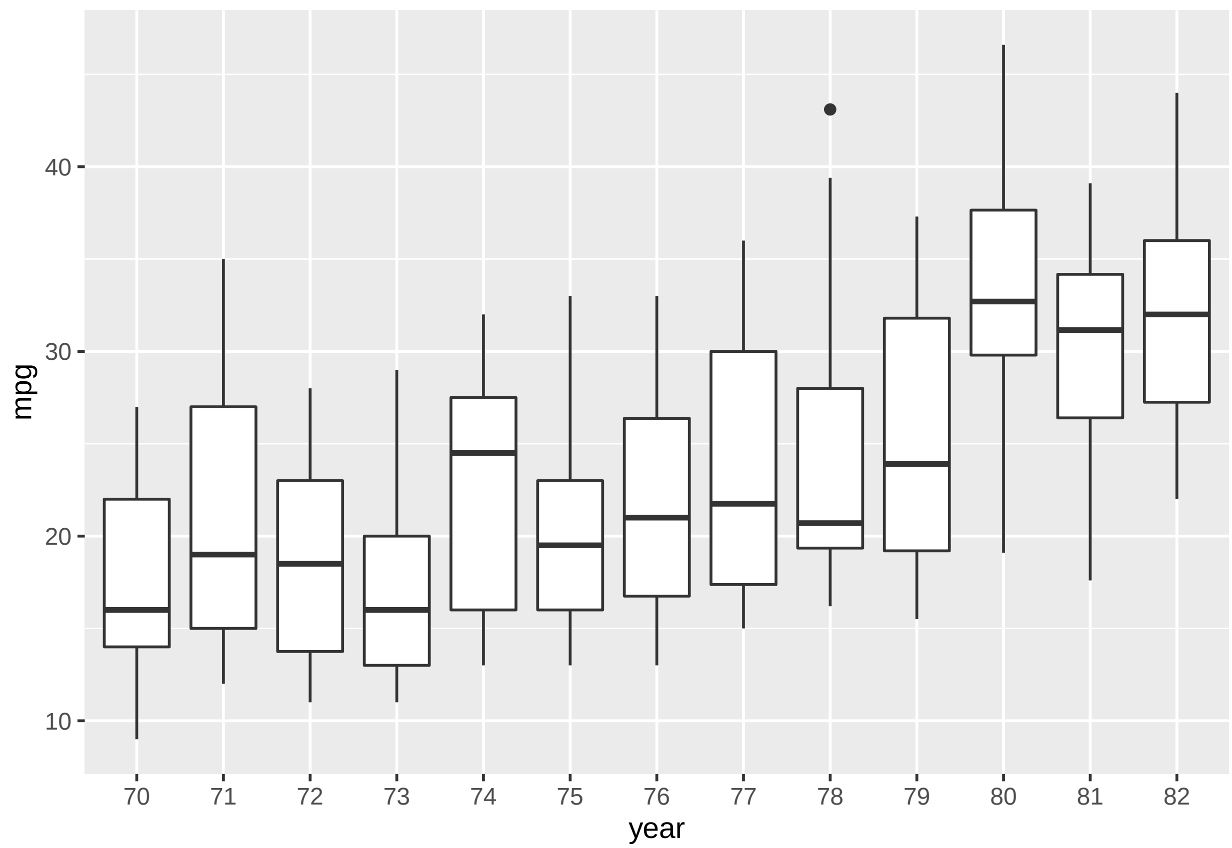
```
par(mfrow=c(1,3))
ggplot(data = Auto, mapping = aes(x = origin, y = mpg)) +
         geom_boxplot()
```




```r
ggplot(data = Auto, mapping = aes(x = cylinders, y = mpg)) +
         geom_boxplot()
```




```r
ggplot(data = Auto, mapping = aes(x = year, y = mpg)) +
        geom_boxplot()
```



Simple Linear Regression

For building our simple linear regression model, using our findings from scatterplot and correlation matrix, we are using the following variables.

Predictor Variable : horsepower **Response Variable** : mpg

```
simple.lm <- lm(mpg ~ horsepower)
summary(simple.lm)
```

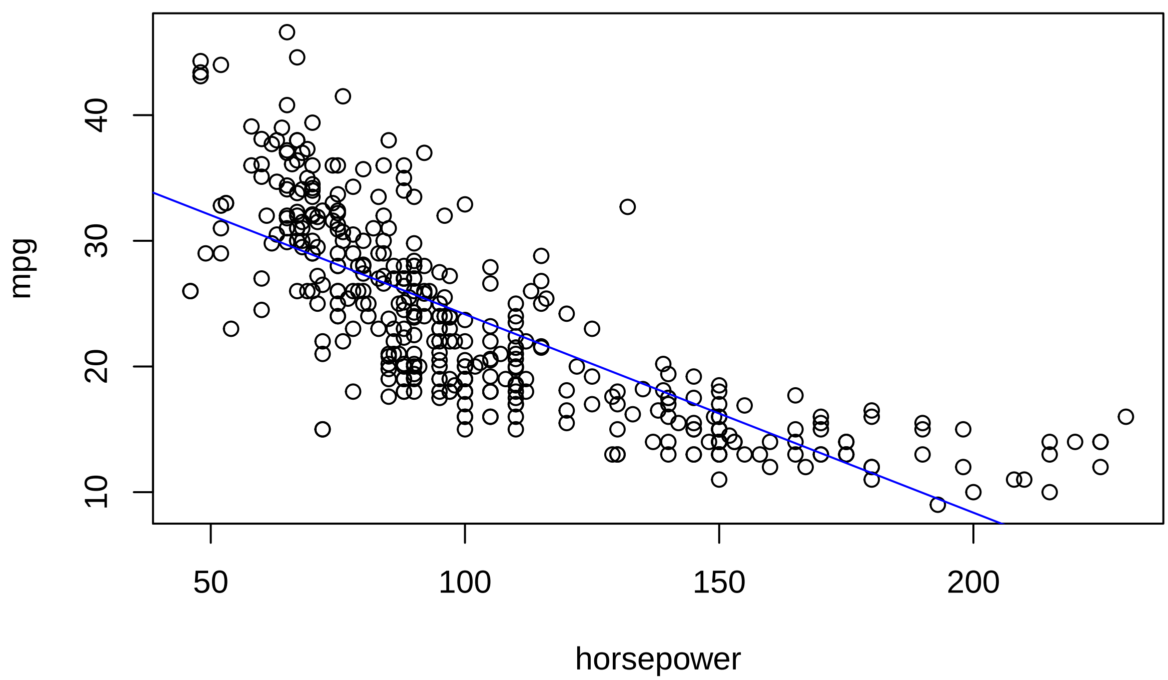
```
##
## Call:
## lm(formula = mpg ~ horsepower)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

Inference from Simple Linear Regression model : mpg ~ horsepower

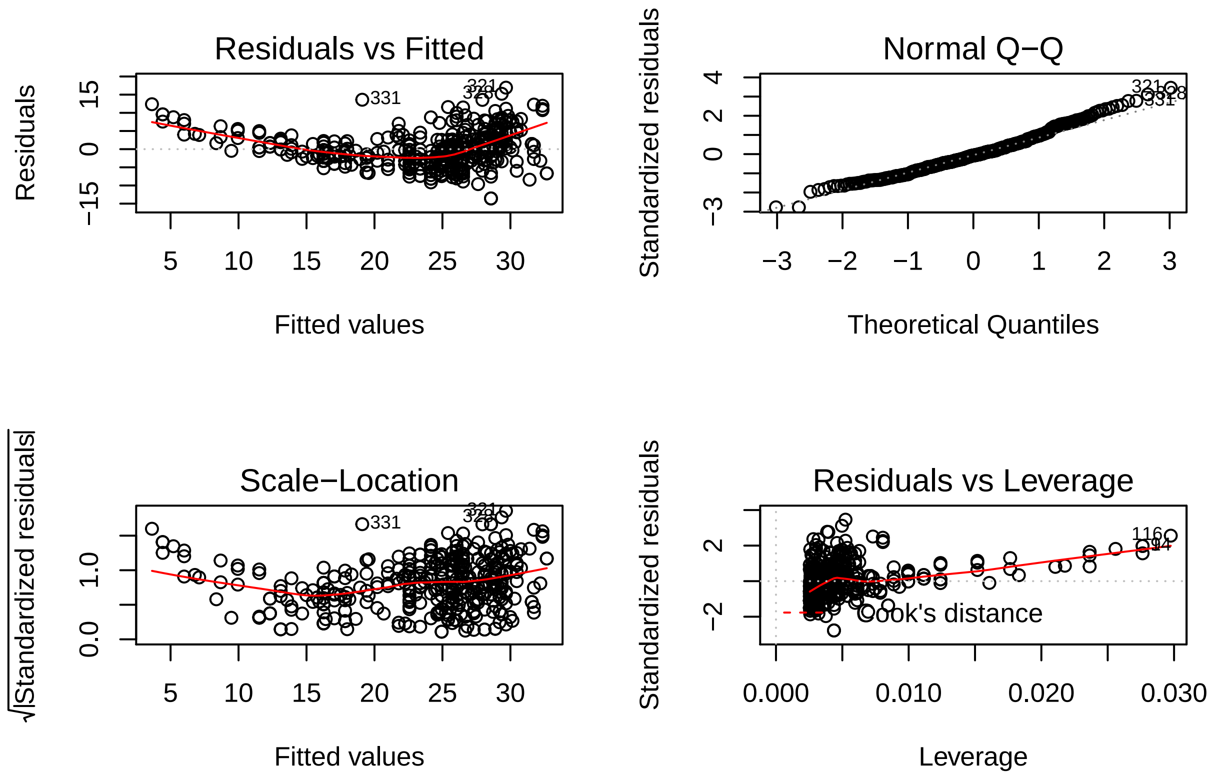
1. From the p-value associated with the F-statistic, we know that there is a strong association between horsepower & mpg.
2. As the coefficient estimate is negative, the relationship between mpg and horsepower is negative.
3. The fit can be observed below which shows a curved pattern which is missed by the straight line assumption of linear model

```r
plot(horsepower, mpg)
abline(simple.lm, col = 'blue')
```



This non-linear relationship is highlighted in the residual vs fitted diagnostic plot.

```r
diagnostic_plot <- function(model){
  par(mfrow=c(2,2))
  plot(model)
}
diagnostic_plot(simple.lm)
```




```r
new_data <- data.frame(horsepower = 98)

cat("The predicted mpg for 98 horsepower is ", predict(simple.lm, new_data))
```

```
## The predicted mpg for 98 horsepower is  24.46708
```

```r
#predict the confidence and prediction interval
conf_interval <- predict(simple.lm, new_data, interval = "confidence")
cat("\n\nThe confidence interval is [", conf_interval[, 2],",",  conf_interval[, 3], "]")
```

```
##
##
## The confidence interval is [ 23.97308 , 24.96108 ]
```

```r
pred_interval <- predict(simple.lm, new_data, interval = "prediction")
cat("\n\nThe prediction interval is [", pred_interval[, 2],",",  pred_interval[, 3], "]")
```

```
##
##
## The prediction interval is [ 14.8094 , 34.12476 ]
```

Multiple Linear Regression

For building our simple linear regression model, using our findings from scatterplot and correlation matrix, we are using all the predictor vairables expect name.

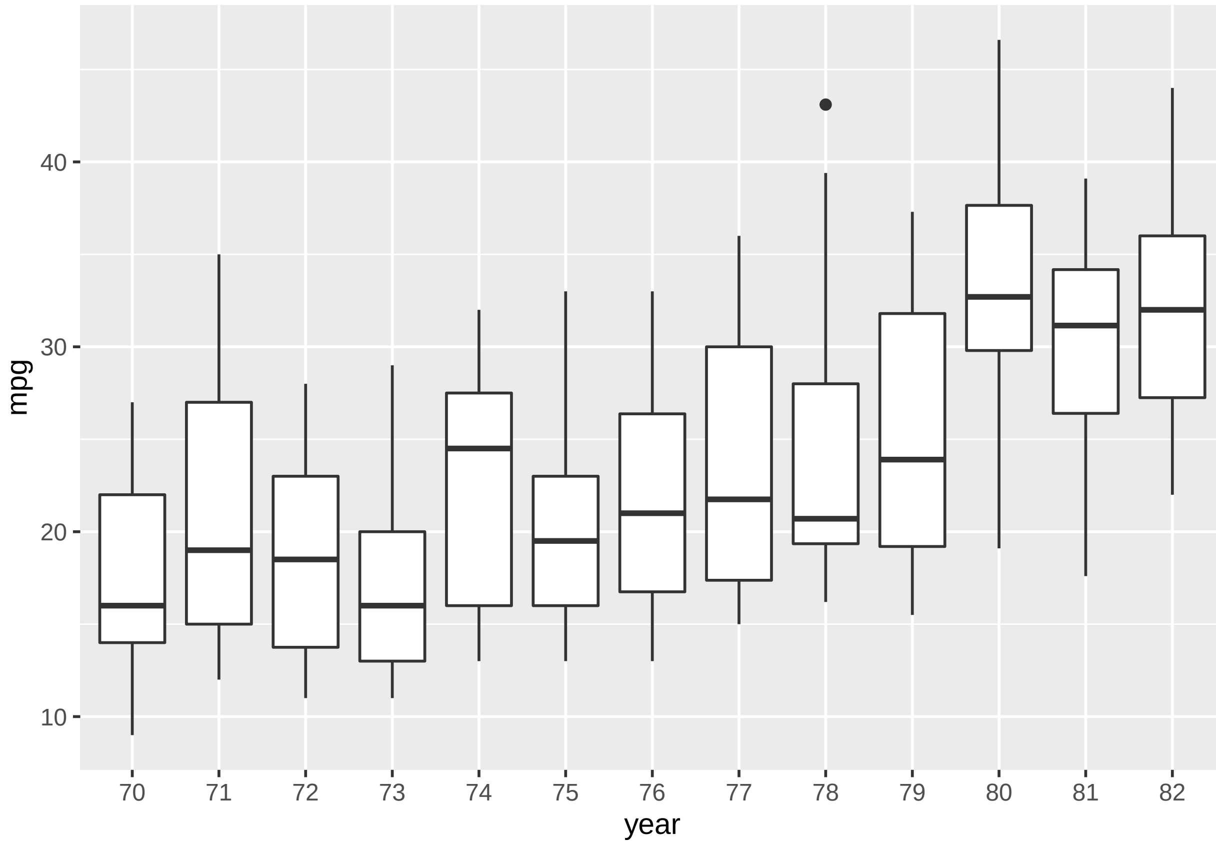
```r
multiple.lm <- lm(mpg ~ . -name, data = Auto)
summary(multiple.lm)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9267 -1.6678 -0.0506  1.4493 11.6002
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.9168415  2.3608985  13.095  < 2e-16 ***
## cylinders4    6.9399216  1.5365961   4.516 8.48e-06 ***
## cylinders5    6.6377310  2.3372687   2.840 0.004762 **
## cylinders6    4.2973139  1.7057848   2.519 0.012182 *
## cylinders8    6.3668129  1.9687277   3.234 0.001331 **
## displacement  0.0118246  0.0067755   1.745 0.081785 .
## horsepower   -0.0392323  0.0130356  -3.010 0.002795 **
## weight       -0.0051802  0.0006241  -8.300 1.99e-15 ***
## acceleration  0.0036080  0.0868925   0.042 0.966902
## year71        0.9104285  0.8155744   1.116 0.265019
## year72       -0.4903062  0.8038193  -0.610 0.542257
## year73       -0.5528934  0.7214463  -0.766 0.443947
## year74        1.2419976  0.8547434   1.453 0.147056
## year75        0.8704016  0.8374036   1.039 0.299297
## year76        1.4966598  0.8019080   1.866 0.062782 .
## year77        2.9986967  0.8198949   3.657 0.000292 ***
## year78        2.9737783  0.7792185   3.816 0.000159 ***
## year79        4.8961763  0.8248124   5.936 6.74e-09 ***
## year80        9.0589316  0.8751948  10.351  < 2e-16 ***
## year81        6.4581580  0.8637018   7.477 5.58e-13 ***
## year82        7.8375850  0.8493560   9.228  < 2e-16 ***
## origin2       1.6932853  0.5162117   3.280 0.001136 **
## origin3       2.2929268  0.4967645   4.616 5.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.848 on 369 degrees of freedom
## Multiple R-squared:  0.8744, Adjusted R-squared:  0.8669
## F-statistic: 116.8 on 22 and 369 DF,  p-value: < 2.2e-16
```

Inference from Multiple Linear Regression model : mpg ~ . - name

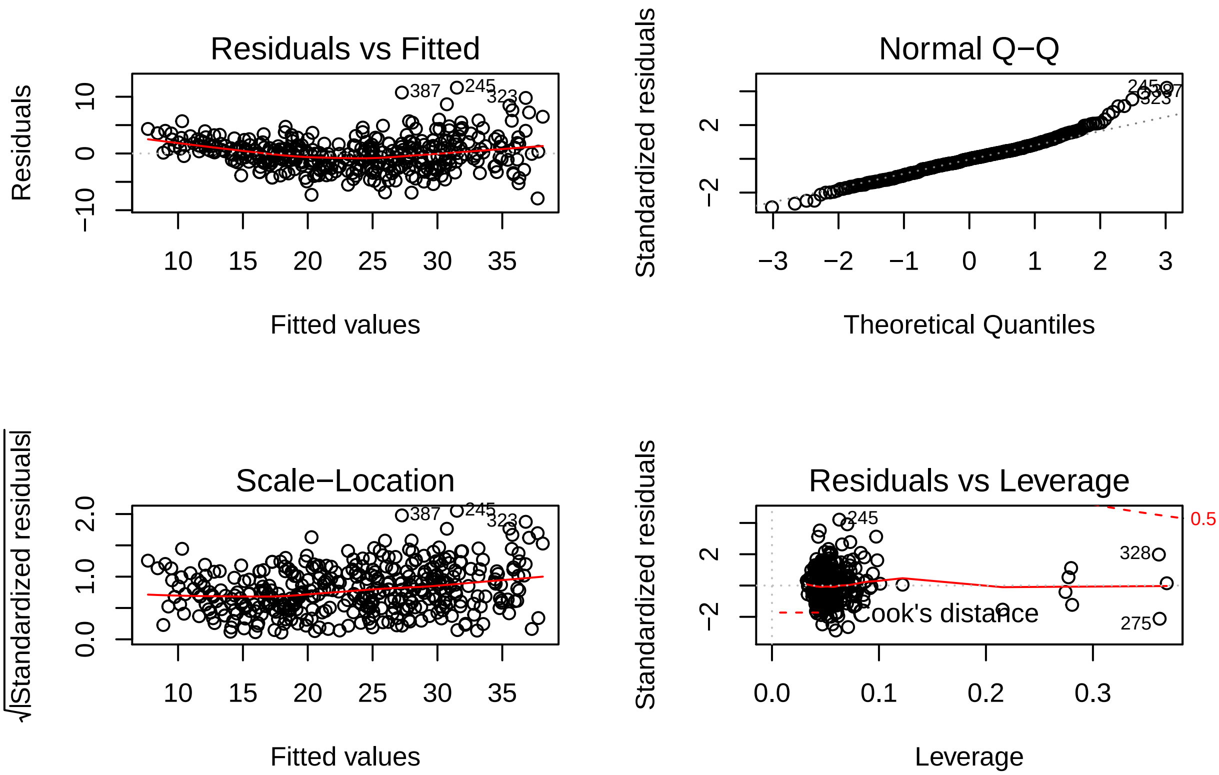
1. From the p-value associated with the F-statistic, we know that there is a relationship between mpg and the predictor variables.
2. Of all the predictors, displacement, weight, year and origin have statistically significant relationship to the response.
3. The coefficient of the year 0.75 suggest that later model have better mpg as shown in the figure below

```r
ggplot(data = Auto, mapping = aes(x = year, y = mpg)) +
         geom_boxplot()
```



4. The residual vs fitted plot show evidence of non-linearity. There is a funnel shape hinting at heteroscedasticity.
5. Additionally, high leverage is noticed for point 14.

```r
diagnostic_plot(multiple.lm)
```




```r
library(rms)
```

```
## Warning: package 'rms' was built under R version 3.6.2
```

```
## Loading required package: Hmisc
```

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

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 3.6.2
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve
```

```
rms::vif(multiple.lm)
```

```
##   cylinders4   cylinders5   cylinders6   cylinders8 displacement
##    28.529864     2.005663    23.477667    36.297458    24.240924
##   horsepower       weight acceleration       year71       year72
##    12.139922    13.551565     2.771138     2.062309     2.071792
##       year73       year74       year75       year76       year77
##     2.305582     2.187237     2.395904     2.462526     2.155488
##       year78       year79       year80       year81       year82
##     2.448166     2.253123     2.374849     2.391977     2.464786
##      origin2      origin3
##     1.847054     1.919740
```

6. Multi collinearity: When VIF values lies in 5 - 10 range, a problematic amount of collinearity is presented. This was evident from the correlation plot too. The problem of collinearity is that it reduces the accuracy of the estimates of the regression coefficients and it causes the standard error to increase.

Interaction Term

```
inter.lm1 <- lm(mpg~displacement+origin+year*weight, data=Auto)
inter.lm2 <- lm(mpg~year+origin+displacement*weight, data=Auto)
summary(inter.lm1)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + origin + year * weight, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7913  -1.6923   0.0385   1.6285  11.4805
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   34.7946118  2.3059551  15.089  < 2e-16 ***
## displacement   0.0028271  0.0047578   0.594 0.552747
## origin2        2.2134814  0.5032141   4.399 1.43e-05 ***
## origin3        1.4619548  0.4872595   3.000 0.002883 **
## year71         2.0378249  2.7778352   0.734 0.463665
## year72        -1.0551438  2.9298669  -0.360 0.718957
## year73        -5.2819682  2.7631468  -1.912 0.056718 .
```

```
## year74         3.6795664  2.8416099   1.295 0.196182
## year75         0.1052171  3.1582296   0.033 0.973442
## year76         5.1245063  2.9259137   1.751 0.080718 .
## year77         7.3137550  2.9046368   2.518 0.012233 *
## year78        14.5480420  3.1747764   4.582 6.33e-06 ***
## year79        14.6067832  3.1845253   4.587 6.21e-06 ***
## year80        21.9191113  3.9920734   5.491 7.54e-08 ***
## year81        15.7498499  3.4674095   4.542 7.59e-06 ***
## year82        18.3543645  4.4689400   4.107 4.95e-05 ***
## weight        -0.0054504  0.0008283  -6.580 1.64e-10 ***
## year71:weight -0.0001164  0.0008262  -0.141 0.888073
## year72:weight  0.0004163  0.0008496   0.490 0.624412
## year73:weight  0.0014515  0.0007880   1.842 0.066282 .
## year74:weight -0.0004688  0.0008752  -0.536 0.592531
## year75:weight  0.0004605  0.0009436   0.488 0.625816
## year76:weight -0.0009151  0.0008786  -1.042 0.298312
## year77:weight -0.0011901  0.0008751  -1.360 0.174663
## year78:weight -0.0038018  0.0010054  -3.782 0.000182 ***
## year79:weight -0.0028298  0.0009628  -2.939 0.003503 **
## year80:weight -0.0046242  0.0014941  -3.095 0.002122 **
## year81:weight -0.0030900  0.0012098  -2.554 0.011053 *
## year82:weight -0.0036276  0.0017160  -2.114 0.035193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.855 on 363 degrees of freedom
## Multiple R-squared:  0.8758, Adjusted R-squared:  0.8662
## F-statistic: 91.39 on 28 and 363 DF,  p-value: < 2.2e-16
```

```r
summary(inter.lm2)
```

```
##
## Call:
## lm(formula = mpg ~ year + origin + displacement * weight, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8567  -1.5177   0.1011   1.4845  11.8388
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.023e+01  1.658e+00  30.301  < 2e-16 ***
## year71               7.912e-01  7.516e-01   1.053  0.29317
## year72              -3.950e-01  7.567e-01  -0.522  0.60202
## year73              -9.209e-01  6.818e-01  -1.351  0.17760
## year74               1.595e+00  7.836e-01   2.035  0.04252 *
## year75               1.878e+00  7.496e-01   2.505  0.01266 *
## year76               2.298e+00  7.262e-01   3.164  0.00168 **
## year77               3.163e+00  7.617e-01   4.152 4.08e-05 ***
## year78               3.838e+00  7.214e-01   5.321 1.79e-07 ***
## year79               5.928e+00  7.467e-01   7.939 2.40e-14 ***
## year80               1.001e+01  7.957e-01  12.577  < 2e-16 ***
## year81               7.349e+00  7.821e-01   9.397  < 2e-16 ***
## year82               8.734e+00  7.809e-01  11.184  < 2e-16 ***
```

```
## origin2              1.016e+00  4.852e-01   2.094  0.03692 *
## origin3              3.832e-01  4.774e-01   0.803  0.42262
## displacement        -7.040e-02  8.897e-03  -7.913 2.88e-14 ***
## weight              -1.018e-02  6.152e-04 -16.552  < 2e-16 ***
## displacement:weight  2.069e-05  2.072e-06   9.984  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.732 on 374 degrees of freedom
## Multiple R-squared:  0.8828, Adjusted R-squared:  0.8775
## F-statistic: 165.7 on 17 and 374 DF,  p-value: < 2.2e-16
```

displacement & weight have statistically significant interaction.

Non-linear transformations

Here we are exploring various non-linear transformations such as log, polynomial.

```
nonlinear.lm1 <- lm(mpg~displacement+I(log(weight))+year+origin, data=Auto)
nonlinear.lm2 <- lm(mpg~poly(displacement,3)+weight+year+origin, data=Auto)
nonlinear.lm3 <- lm(mpg~displacement+I(weight^2)+year+origin, data=Auto)
summary(nonlinear.lm1)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + I(log(weight)) + year + origin,
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0601  -1.6555   0.1519   1.5370  11.3948
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    185.394162  10.257706  18.074  < 2e-16 ***
## displacement     0.009309   0.004031   2.310 0.021448 *
## I(log(weight)) -21.103367   1.385625 -15.230  < 2e-16 ***
## year71           1.198882   0.767482   1.562 0.119108
## year72           0.346844   0.770952   0.450 0.653049
## year73          -0.300029   0.696213  -0.431 0.666756
## year74           2.052366   0.797550   2.573 0.010456 *
## year75           1.983609   0.771061   2.573 0.010479 *
## year76           2.469789   0.745208   3.314 0.001008 **
## year77           3.639869   0.778581   4.675 4.11e-06 ***
## year78           3.853289   0.743565   5.182 3.59e-07 ***
## year79           6.122424   0.769115   7.960 2.06e-14 ***
## year80          10.474025   0.818927  12.790  < 2e-16 ***
## year81           7.677553   0.804303   9.546  < 2e-16 ***
## year82           9.246233   0.802180  11.526  < 2e-16 ***
## origin2          1.865680   0.480531   3.883 0.000122 ***
## origin3          1.293261   0.465406   2.779 0.005731 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.823 on 375 degrees of freedom
## Multiple R-squared:  0.8746, Adjusted R-squared:  0.8692
## F-statistic: 163.4 on 16 and 375 DF,  p-value: < 2.2e-16
```

```r
summary(nonlinear.lm2)
```

```
##
## Call:
## lm(formula = mpg ~ poly(displacement, 3) + weight + year + origin,
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1739  -1.5085   0.1376   1.6098  11.4039
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            34.766016   1.536021  22.634  < 2e-16 ***
## poly(displacement, 3)1 -16.625947  10.186541  -1.632  0.10349
## poly(displacement, 3)2  25.946455   3.482753   7.450 6.56e-13 ***
## poly(displacement, 3)3  -9.549773   2.992799  -3.191  0.00154 **
## weight                 -0.005054   0.000535  -9.448  < 2e-16 ***
## year71                  1.464949   0.786550   1.862  0.06332 .
## year72                 -0.160327   0.793815  -0.202  0.84005
## year73                 -0.431942   0.710701  -0.608  0.54371
## year74                  1.943962   0.826763   2.351  0.01923 *
## year75                  2.112021   0.788541   2.678  0.00772 **
## year76                  2.479230   0.766511   3.234  0.00133 **
## year77                  3.422878   0.799123   4.283 2.35e-05 ***
## year78                  4.062263   0.758887   5.353 1.52e-07 ***
## year79                  6.109192   0.785820   7.774 7.47e-14 ***
## year80                 10.170661   0.831490  12.232  < 2e-16 ***
## year81                  7.659280   0.816816   9.377  < 2e-16 ***
## year82                  9.080391   0.813725  11.159  < 2e-16 ***
## origin2                 0.657935   0.531224   1.239  0.21630
## origin3                 0.260748   0.519380   0.502  0.61594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.85 on 373 degrees of freedom
## Multiple R-squared:  0.8728, Adjusted R-squared:  0.8667
## F-statistic: 142.2 on 18 and 373 DF,  p-value: < 2.2e-16
```

```r
summary(nonlinear.lm3)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + I(weight^2) + year + origin,
##     data = Auto)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -11.3824  -2.0885  -0.0761   1.7075  13.9065
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.767e+01  1.014e+00  27.286  < 2e-16 ***
## displacement -8.967e-03  5.293e-03  -1.694 0.091075 .
## I(weight^2)  -6.635e-07  8.965e-08  -7.401 8.94e-13 ***
## year71        1.471e+00  9.229e-01   1.593 0.111918
## year72       -2.879e-01  9.271e-01  -0.311 0.756302
## year73       -5.384e-01  8.357e-01  -0.644 0.519808
## year74        1.620e+00  9.640e-01   1.680 0.093791 .
## year75        7.272e-01  9.156e-01   0.794 0.427592
## year76        1.579e+00  8.901e-01   1.774 0.076914 .
## year77        3.049e+00  9.349e-01   3.261 0.001211 **
## year78        2.733e+00  8.805e-01   3.104 0.002058 **
## year79        5.348e+00  9.152e-01   5.844 1.11e-08 ***
## year80        9.348e+00  9.741e-01   9.597  < 2e-16 ***
## year81        6.835e+00  9.590e-01   7.128 5.28e-12 ***
## year82        8.567e+00  9.580e-01   8.943  < 2e-16 ***
## origin2       2.215e+00  5.816e-01   3.808 0.000164 ***
## origin3       2.580e+00  5.491e-01   4.699 3.68e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.354 on 375 degrees of freedom
## Multiple R-squared:  0.8228, Adjusted R-squared:  0.8153
## F-statistic: 108.9 on 16 and 375 DF,  p-value: < 2.2e-16
```

From noticing the p-value associated with the different variables which have been transformed, we know that

1. displacement has a lower p-value for square as compared to cubic
2. weight^2 is statistically significant.