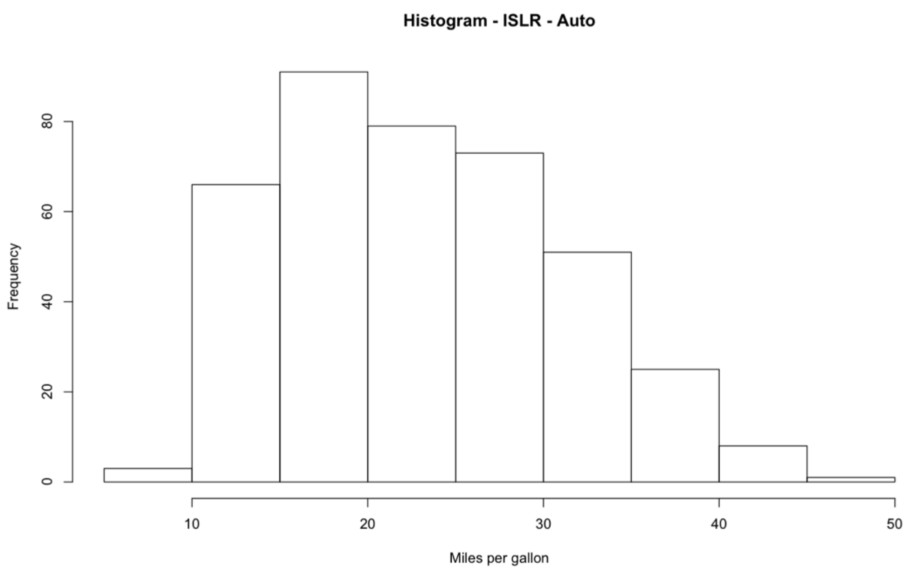
STA HOMEWORK 1

Name: Priya Murthy
UB Person No.: 50248887

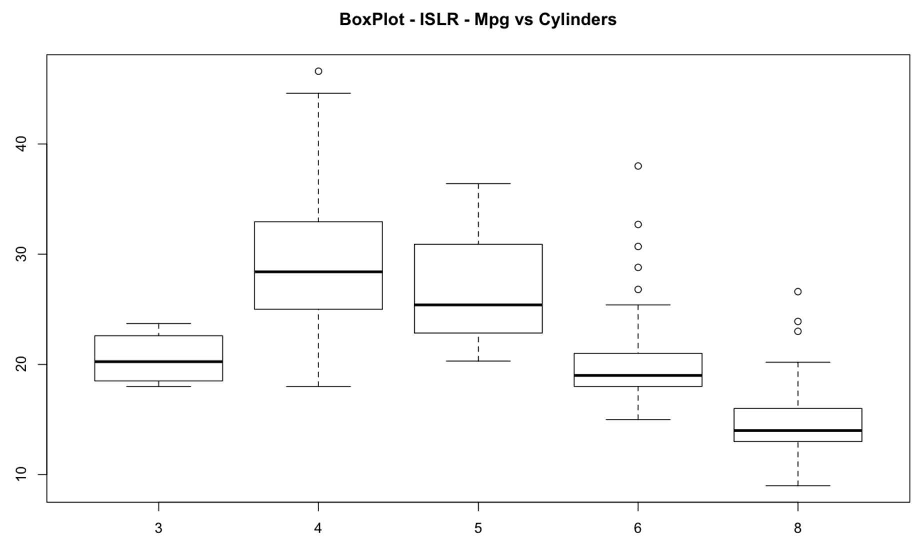
Q1. To build a predictive model for mpg (miles per gallon) using exploratory data analysis.

Histogram – ISLR - Auto



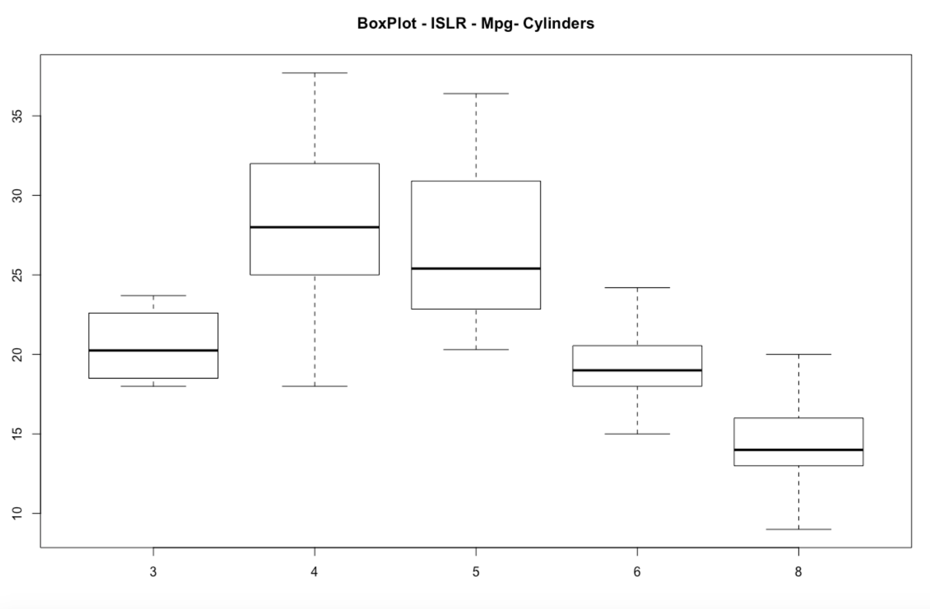
Boxplot of Mpg vs Cylinders

We can see that the frequency is maximum when MPG is between 15 to 30.

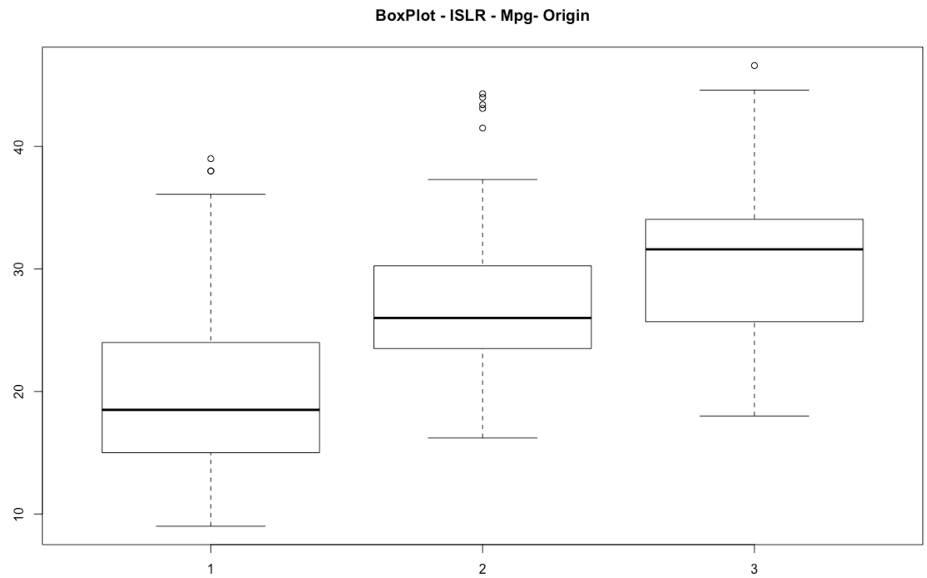


We can see that there are outliers in this data which need to be removed to clean the data set.

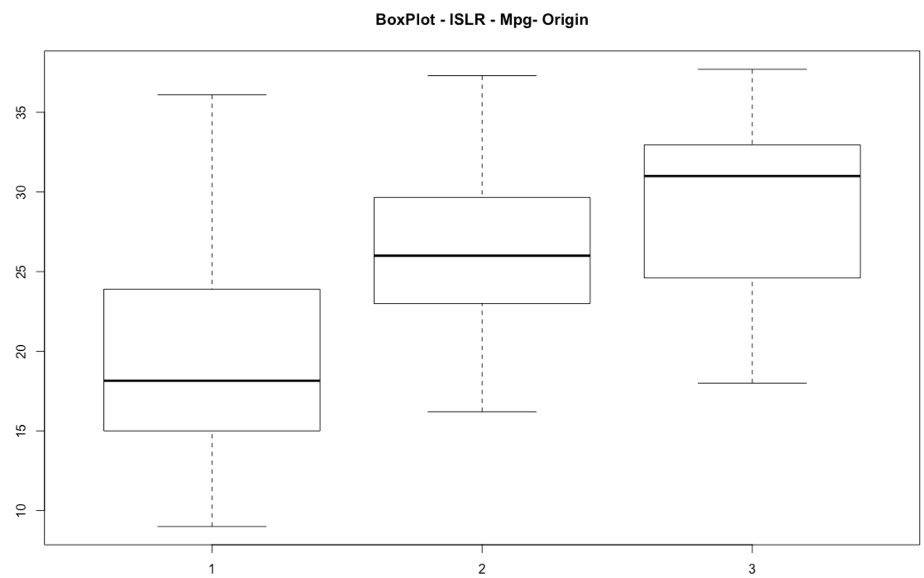
After Cleaning the Data:



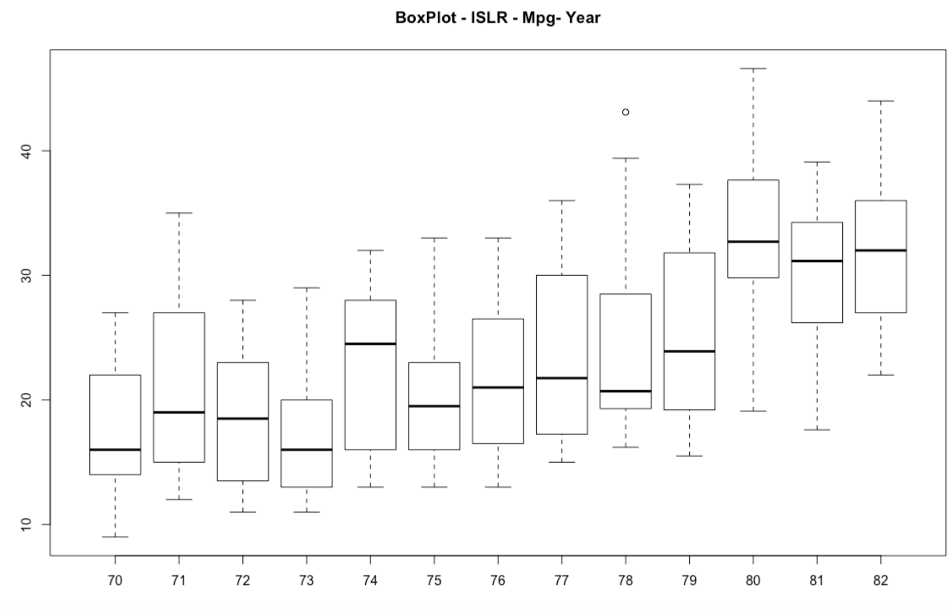
Boxplot – Mpg vs Origin



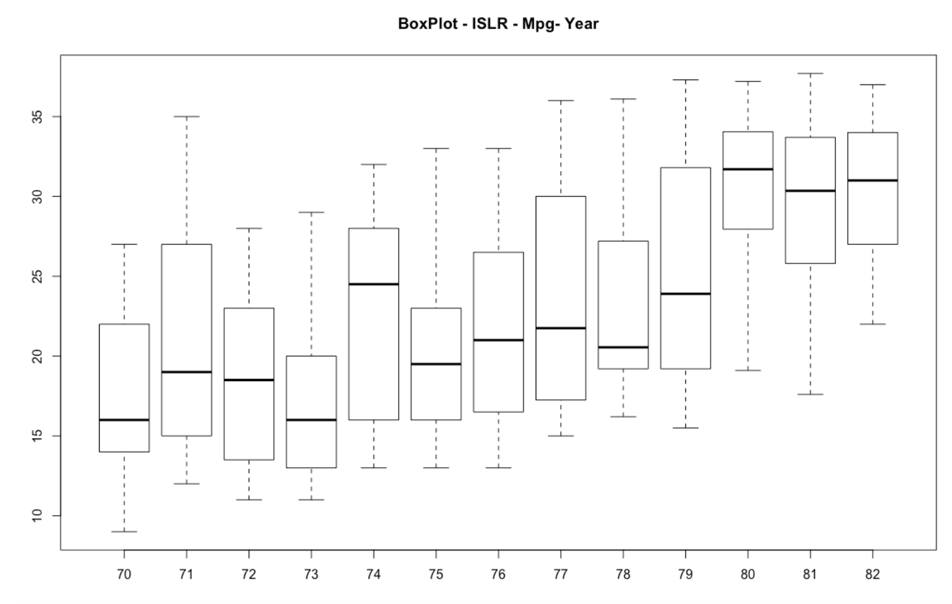
After Cleaning the Data



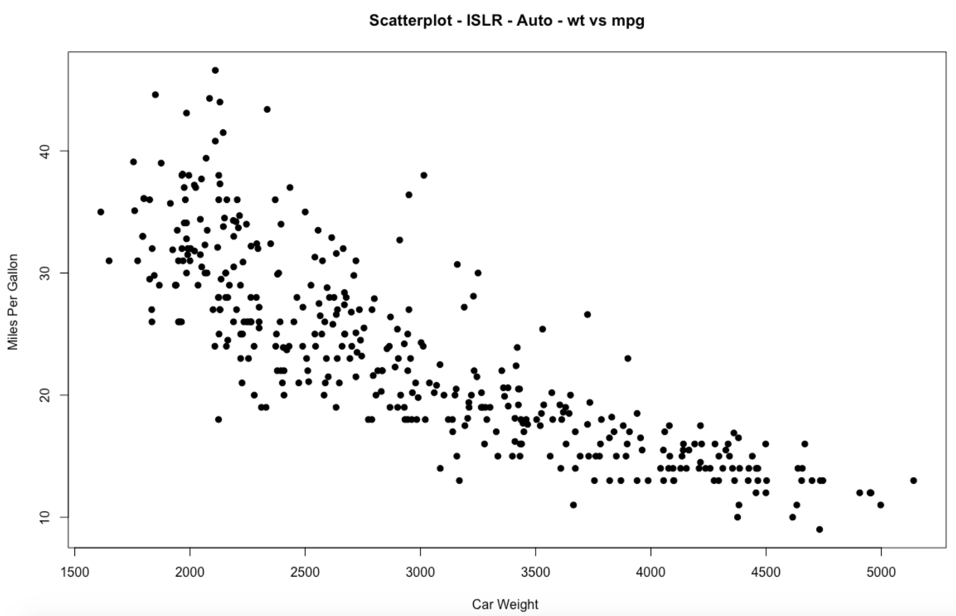
Boxplot: Mpg vs Year:
We can see that there is just one outlier

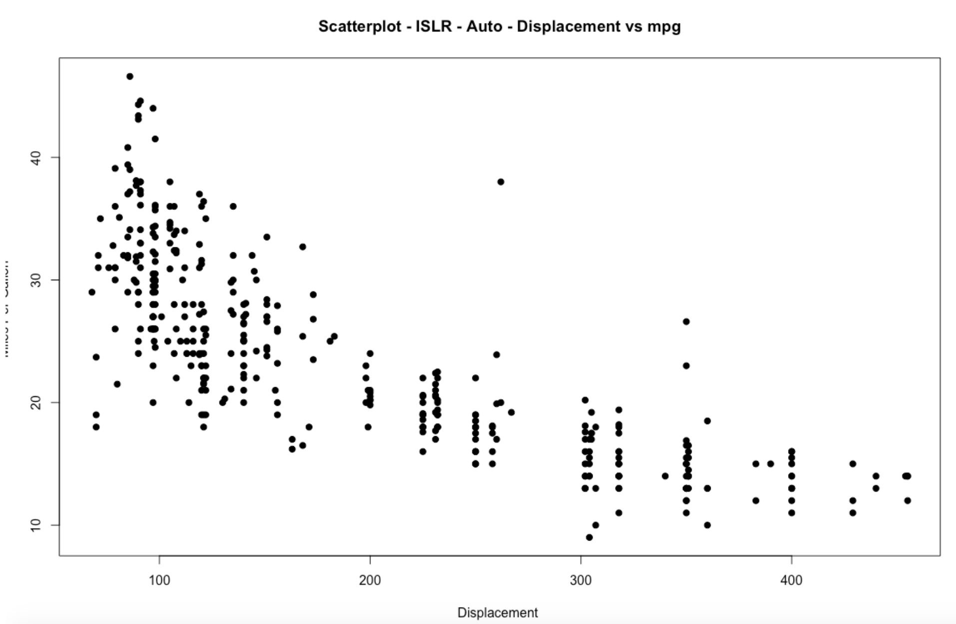


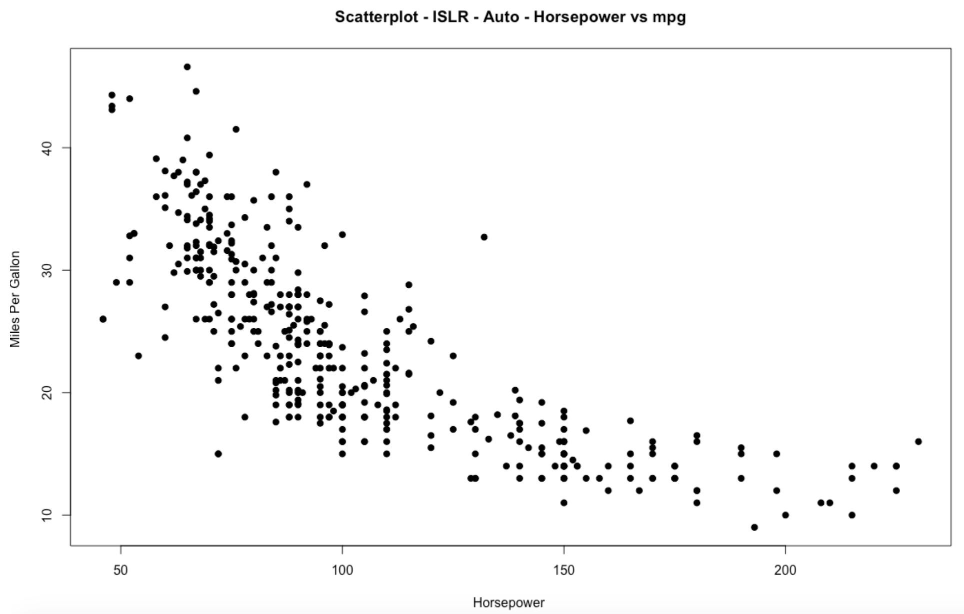
Clean Data Set:

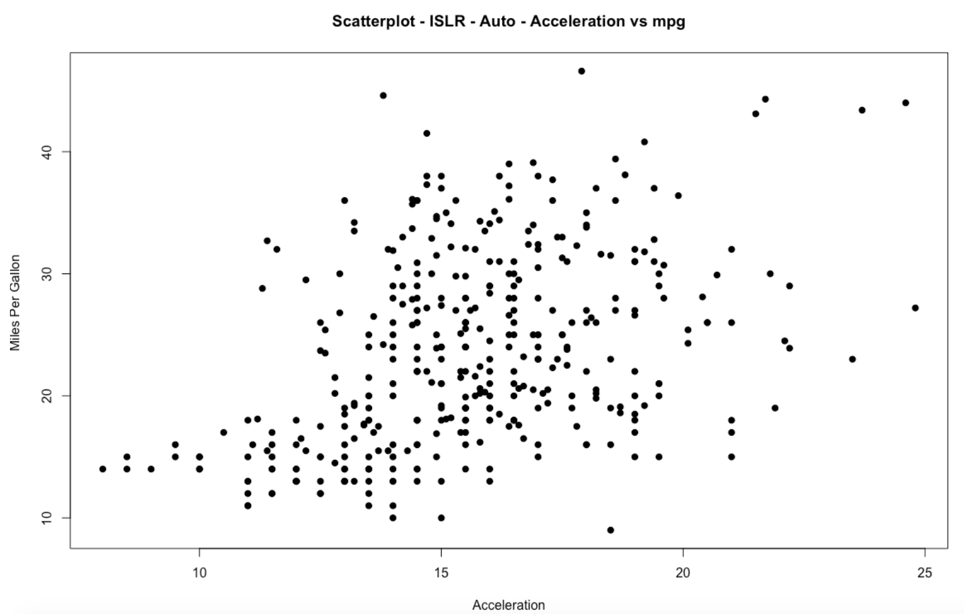


Scatter Plots between Predictors



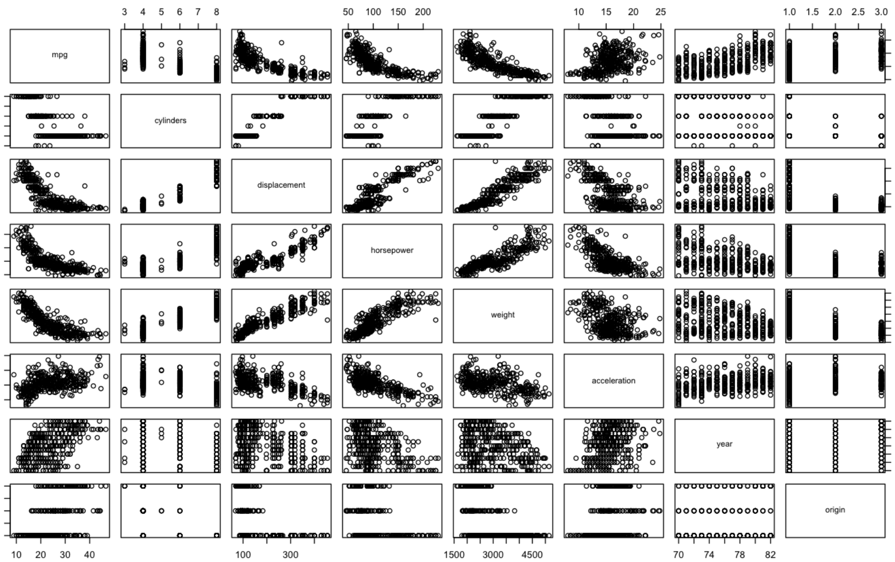






Since the Column “Name” will not have any significant relationship with the other variables. We have removed the predictor Name from the data set.

Creating a Scatterplot Matrix:



This shows the linear correlation between multiple variables of the dataset.

##lm function - creates relationship between the predictor and response variable
mpg of Auto data set is the response variable

```
> head(CleanData)
  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
> result<-lm(CleanData$mpg~.,data=CleanData)
> print(summary(result))

Call:
lm(formula = CleanData$mpg ~ ., data = CleanData)

Residuals:
    Min      1Q  Median      3Q     Max 
-10.194  -1.674   0.175   1.739  10.727 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.4150513  4.1679763  -0.819  0.41313    
cylinders    -0.8188162  0.2842599  -2.881  0.00421 ** 
displacement  0.0112256  0.0066411   1.690  0.09184 .  
horsepower   -0.0208396  0.0117810  -1.769  0.07776 .  
weight       -0.0050456  0.0005677  -8.887  < 2e-16 ***
acceleration -0.1625404  0.0899806  -1.806  0.07170 .  
year          0.6157766  0.0451731  13.631  < 2e-16 ***
origin        1.0383457  0.2450556   4.237 2.88e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.784 on 358 degrees of freedom
Multiple R-squared:  0.8483,	Adjusted R-squared:  0.8454 
F-statistic: 286.1 on 7 and 358 DF,  p-value: < 2.2e-16
```

2.a

To Determine the predictors which seem to have a significant relationship to response -> We look at the t value

Thus, Displacement, Weight, Year and Origin have a significant relationship to response as their t-value is either <-2 or greater than 2.

2.b

When every other predictor held constant, the mpg value increases with each year that passes, mpg increase by some value each year.

2.c

Creating interaction models using: and *.

```
> output = lm(mpg ~ displacement:weight, data =CleanData)
> summary(output)

Call:
lm(formula = mpg ~ displacement:weight, data = CleanData)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.4577  -2.8692  -0.6279   2.6265  10.2454 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          3.010e+01  3.389e-01   88.83   <2e-16 ***
displacement:weight -1.106e-05  3.940e-07  -28.07   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.986 on 364 degrees of freedom
Multiple R-squared:  0.6839,	Adjusted R-squared:  0.6831 
F-statistic: 787.7 on 1 and 364 DF,  p-value: < 2.2e-16
```

```
> output1 = lm(mpg ~ displacement:cylinders+displacement:weight+acceleration:horsepower, data=CleanData)
> summary(output1)

Call:
lm(formula = mpg ~ displacement:cylinders + displacement:weight + 
    acceleration:horsepower, data = CleanData)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.9214  -2.7061  -0.1983   2.3605  10.2872 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              3.940e+01  1.024e+00  38.491  < 2e-16 ***
displacement:cylinders  -4.388e-03  1.050e-03  -4.179 3.67e-05 ***
displacement:weight      2.119e-06  2.172e-06   0.975     0.33    
acceleration:horsepower -8.122e-03  8.519e-04  -9.534  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.556 on 362 degrees of freedom
Multiple R-squared:  0.7498,	Adjusted R-squared:  0.7477 
F-statistic: 361.6 on 3 and 362 DF,  p-value: < 2.2e-16


> output2 = lm(mpg ~. -cylinders-acceleration+year:origin+displacement:weight+
+              displacement:weight+acceleration:horsepower+acceleration:weight, data=CleanData)
> summary(output2)

Call:
lm(formula = mpg ~ . - cylinders - acceleration + year:origin + 
    displacement:weight + displacement:weight + acceleration:horsepower + 
    acceleration:weight, data = CleanData)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.6634 -1.3568  0.2929  1.3933  9.3265 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              2.253e+01  6.821e+00   3.303  0.00105 ** 
displacement            -7.771e-02  7.852e-03  -9.897  < 2e-16 ***
horsepower               4.789e-02  2.782e-02   1.721  0.08608 .  
weight                  -9.886e-03  1.154e-03  -8.567 3.27e-16 ***
year                     4.024e-01  8.623e-02   4.666 4.35e-06 ***
origin                  -1.151e+01  3.611e+00  -3.188  0.00156 ** 
year:origin              1.512e-01  4.654e-02   3.250  0.00127 ** 
displacement:weight      1.949e-05  1.909e-06  10.213  < 2e-16 ***
horsepower:acceleration -6.007e-03  1.960e-03  -3.064  0.00235 ** 
weight:acceleration      1.293e-04  6.325e-05   2.045  0.04161 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.39 on 356 degrees of freedom
Multiple R-squared:  0.8889,	Adjusted R-squared:  0.8861 
F-statistic: 316.4 on 9 and 356 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = mpg ~ (.) * (.), data = CleanData)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.8645 -1.2535  0.0391  1.1929  7.8053 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)   
(Intercept)                2.944e+01  4.547e+01   0.647  0.51781   
cylinders                  1.226e+01  7.074e+00   1.733  0.08410 . 
displacement              -1.609e-01  1.702e-01  -0.945  0.34511   
horsepower                 7.163e-02  2.972e-01   0.241  0.80970   
weight                    -2.255e-02  1.478e-02  -1.525  0.12809   
acceleration              -3.389e+00  1.974e+00  -1.717  0.08683 . 
year                       7.087e-01  5.211e-01   1.360  0.17477   
origin                    -9.526e+00  6.157e+00  -1.547  0.12274   
cylinders:displacement    -1.226e-02  6.328e-03  -1.937  0.05352 . 
cylinders:horsepower       2.050e-02  2.030e-02   1.010  0.31318   
cylinders:weight           4.482e-04  7.622e-04   0.588  0.55691   
cylinders:acceleration     2.507e-01  1.562e-01   1.605  0.10943   
cylinders:year            -2.172e-01  8.339e-02  -2.605  0.00960 **
cylinders:origin          -6.246e-01  4.428e-01  -1.411  0.15931   
displacement:horsepower    1.376e-04  2.427e-04   0.567  0.57123   
displacement:weight        3.158e-05  1.243e-05   2.541  0.01151 * 
displacement:acceleration -6.116e-03  2.865e-03  -2.135  0.03352 * 
displacement:year          1.904e-03  2.147e-03   0.887  0.37593   
displacement:origin        4.074e-02  1.670e-02   2.440  0.01519 * 
horsepower:weight         -4.800e-05  2.464e-05  -1.948  0.05222 . 
horsepower:acceleration    2.922e-03  3.328e-03   0.878  0.38065   
horsepower:year           -1.887e-03  3.372e-03  -0.560  0.57609   
horsepower:origin         -1.515e-02  2.516e-02  -0.602  0.54757   
weight:acceleration        2.345e-04  1.895e-04   1.237  0.21681   
weight:year                1.706e-04  1.784e-04   0.956  0.33954   
weight:origin             -9.470e-04  1.363e-03  -0.695  0.48764   
acceleration:year          1.778e-02  2.331e-02   0.763  0.44616   
acceleration:origin        3.910e-01  1.329e-01   2.943  0.00347 **
year:origin                6.542e-02  6.337e-02   1.032  0.30265   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We create the above models using mpg(miles per gallon) as the dependent variable and find the relationship of other variables (predictors) and interactions with mpg.
From all the models the third one is the one with all the variables having significant value.

3.

KNN and Linear Regression

For KNN:
Error Test is:
print(error_test)
[1] 0.02472527 0.03021978 0.03021978 0.03021978 0.03571429 0.03571429 0.03296703 0.03846154

Train Data error is:
print(error_train)
[1] 0.000000000 0.004319654 0.005759539 0.005759539 0.007919366 0.007919366 0.007919366 0.009359251

Test Accuracy KNN for different values of K
[1] 97.52747
[1] 96.97802
[1] 96.97802
[1] 96.97802
[1] 96.42857
[1] 96.42857
[1] 96.7033
[1] 96.15385

Train Accuracy KNN for different values of K
[1] 100
[1] 99.56803
[1] 99.42405
[1] 99.42405
[1] 99.20806
[1] 99.20806
[1] 99.20806
[1] 99.06407

The above values of accuracy are for k = 1,3,5,7,9,11,13,15 in order respectively.

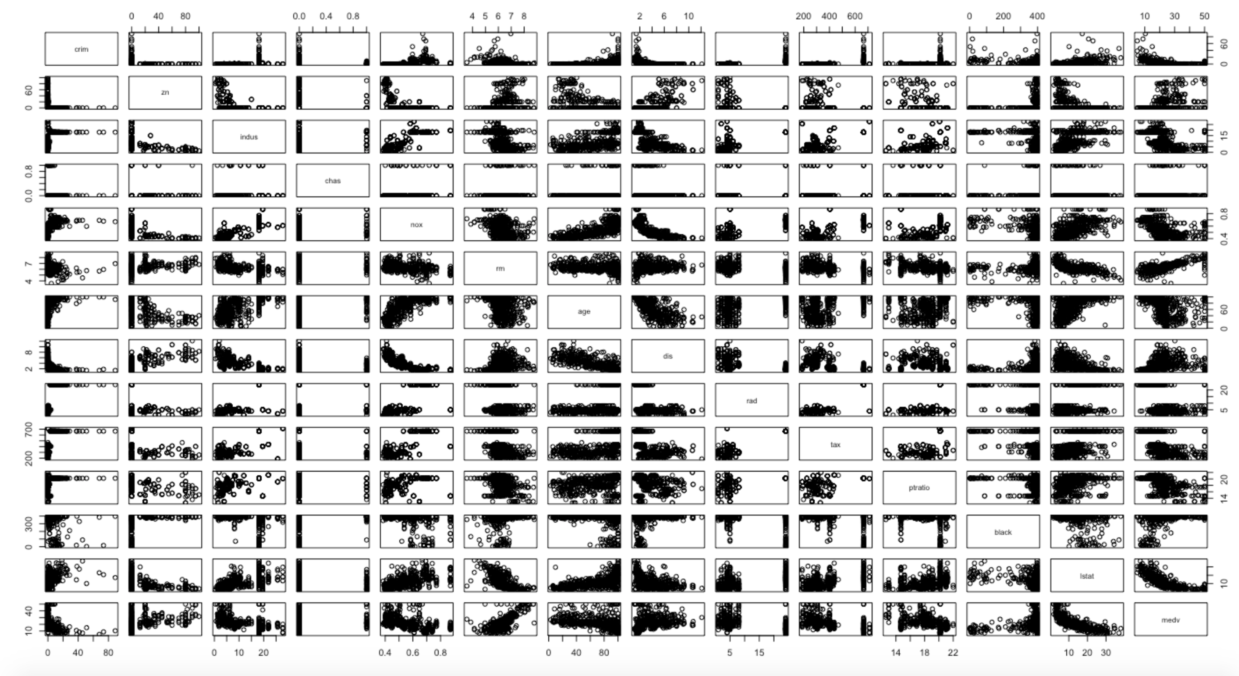
The accuracy for Linear regression for Training is 97.51883%
The accuracy for Linear Regression for Test is about 75%

Therefore -> KNN has better accuracy and it is best when K=1

4.

Boston housing data in the MASS library

a. Scatterplot matrix between all the predictors of Boston dataset



Looking at the above scatterplot we can say that, all the predictors have some relationship with the others. Also, there are some predictor pairs which are highly correlated and others that are not highly correlated. To understand further, let’s look at their correlation values.

The figure below shows the correlation between all the predictors

```
> cor(Boston, Boston)
               crim          zn       indus         chas         nox          rm         age
crim     1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171 -0.21924670  0.35273425
zn      -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371  0.31199059 -0.56953734
indus    0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145 -0.39167585  0.64477851
chas    -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281  0.09125123  0.08651777
nox      0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000 -0.30218819  0.73147010
rm      -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819  1.00000000 -0.24026493
age      0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010 -0.24026493  1.00000000
dis     -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011  0.20524621 -0.74788054
rad      0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056 -0.20984667  0.45602245
tax      0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320 -0.29204783  0.50645559
ptratio  0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268 -0.35550149  0.26151501
black   -0.38506394  0.17552032 -0.35697654  0.048788485 -0.38005064  0.12806864 -0.27353398
lstat    0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892 -0.61380827  0.60233853
medv    -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077  0.69535995 -0.37695457
                dis          rad         tax    ptratio       black      lstat       medv
crim    -0.37967009  0.625505145  0.58276431  0.2899456 -0.38506394  0.4556215 -0.3883046
zn       0.66440822 -0.311947826 -0.31456332 -0.3916785  0.17552032 -0.4129946  0.3604453
indus   -0.70802699  0.595129275  0.72076018  0.3832476 -0.35697654  0.6037997 -0.4837252
chas    -0.09917578 -0.007368241 -0.03558652 -0.1215152  0.04878848 -0.0539293  0.1752602
nox     -0.76923011  0.611440563  0.66802320  0.1889327 -0.38005064  0.5908789 -0.4273208
rm       0.20524621 -0.209846668 -0.29204783 -0.3555015  0.12806864 -0.6138083  0.6953599
age     -0.74788054  0.456022452  0.50645559  0.2615150 -0.27353398  0.6023385 -0.3769546
dis      1.00000000 -0.494587930 -0.53443158 -0.2324705  0.29151167 -0.4969958  0.2499287
rad     -0.49458793  1.000000000  0.91022819  0.4647412 -0.44441282  0.4886763 -0.3816262
tax     -0.53443158  0.910228189  1.00000000  0.4608530 -0.44180801  0.5439934 -0.4685359
ptratio -0.23247054  0.464741179  0.46085304  1.0000000 -0.17738330  0.3740443 -0.5077867
black    0.29151167 -0.444412816 -0.44180801 -0.1773833  1.00000000 -0.3660869  0.3334608
lstat   -0.49699583  0.488676335  0.54399341  0.3740443 -0.36608690  1.0000000 -0.7376627
medv     0.24992873 -0.381626231 -0.46853593 -0.5077867  0.33346082 -0.7376627  1.0000000
```

- Looking at the above correlation, we can conclude the following
  i) Crime rate has high correlation with the predictor rad(index of accessibility to radial highways) , i.e rad highly affects the value of crime rate in a particular suburb.
  ii) If we look at the next predictor, zn (proportion of residential land zoned for lots over 25,000 sq.ft.), it also has high correlation with crime rate.
  iii) Predictors ‘rad’ and ‘tax rate’ have correlation = 0.91, which is the highest correlation of all the other predictors.
  iv) Similarly, we can find the relationship between all the predictors using the correlation function.

b. Are any of the predictors associated with per capita crime rate?

```
> cor(Boston$crim, Boston)
     crim         zn     indus        chas       nox         rm       age        dis       rad
[1,]    1 -0.2004692 0.4065834 -0.05589158 0.4209717 -0.2192467 0.3527343 -0.3796701 0.6255051
          tax   ptratio      black     lstat       medv
[1,] 0.5827643 0.2899456 -0.3850639 0.4556215 -0.3883046
```

- Looking at the above output, we can say that there is association between per capita income and other crime rates.
- Rad (Index of accessibility to radial highways) has highest correlation with crime rate.

c. 4.c Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each predictor.

```
> #Calculating range of each predictor
> #Calculating range of Crime rate using Histogram
> crim1 <- subset(Boston, Boston$crim >=0 & Boston$crim < 10 )
> percentage = nrow(crim1)/nrow(Boston)
> print(percentage)
[1] 0.8932806
>
> crim2 <- subset(Boston, Boston$crim >=10  & Boston$crim < 20 )
> percentage = nrow(crim2)/nrow(Boston)
> print(percentage)
[1] 0.07114625
>
> crim3 <- subset(Boston, Boston$crim >=20  & Boston$crim < 30 )
> percentage = nrow(crim3)/nrow(Boston)
> print(percentage)
[1] 0.01976285
>
> crim4 <- subset(Boston, Boston$crim >10 )
> percentage = nrow(crim4)/nrow(Boston)
> print(percentage)
[1] 0.1067194
>
> crim5 <- subset(Boston, Boston$crim >20 )
> percentage = nrow(crim5)/nrow(Boston)
> print(percentage)
[1] 0.03557312
```

Looking at the above percentages: We can see that crime rate is > 10 for 11% Suburbs and is <10 for 8.9% of suburbs.

Yes, there are suburbs with higher crime rate

Now, to find out suburbs near to the river and away from the river (Chas predictor)

```
> Near_River <- length(Boston$chas[Boston$crim>10 & Boston$chas == 1])
> Away_River <- length(Boston$chas[Boston$crim>10 & Boston$chas == 0])
> print(Near_River)
[1] 0
> print(Away_River)
[1] 54
```

Thus, there are 54 houses which are away the river when crime rate is greater than 10.

```
> #Calculating range of Tax rate using Histogram
> tax1 <- subset(Boston, Boston$tax<500)
> percentage = nrow(tax1)/nrow(Boston)
> print(percentage)
[1] 0.729249
>
> tax2 <- subset(Boston, Boston$tax >=500 )
> percentage = nrow(tax2)/nrow(Boston)
> print(percentage)
[1] 0.270751
>
> ##Calculating how many suburbs are away from the river and near the river
> Near_River <- length(Boston$chas[Boston$tax<500 & Boston$chas == 1])
> Away_River <- length(Boston$chas[Boston$tax<500 & Boston$chas == 0])
> print(Near_River)
[1] 27
>
count(x, ..., wt = NULL, sort = FALSE)
[1] 342
> count(Away_River)
```

This, there are 342 suburbs away from the river when tax rate <500

```
> ##Calculating how many suburbs are away from the river and near the river
> Near_River <- length(Boston$chas[Boston$ptratio>=19 & Boston$chas == 1])
> Away_River <- length(Boston$chas[Boston$ptratio<19 & Boston$chas == 0])
> print(Near_River)
[1] 8
> print(Away_River)
[1] 222
> count(Away_River)
```

19 is the median of pupil teacher ratio.

Therefore, on counting the suburbs away from the river , we can see that there are 222 houses away from the river with Pupil –teacher ratio <19.

4d. In this data set, how many of the suburbs average more than seen rooms per dwelling? More than eight rooms per dwelling? Comment on the suburbs that average more than eight rooms per dwelling.

Number of suburbs which average more than seven rooms per dwelling = 64
Number of suburbs which average more than eight rooms per dwelling = 13.

```
> rooms_mt_7 <- subset(Boston , Boston$rm > 7)
> print(nrow(rooms_mt_7))
[1] 64
> rooms_mt_8 <- subset(Boston , Boston$rm > 8)
> print(nrow(rooms_mt_8))
[1] 13
> print(summary(rooms_mt_8))
      crim               zn             indus             chas             nox        
 Min.   :0.02009   Min.   : 0.00   Min.   : 2.680   Min.   :0.0000   Min.   :0.4161  
 1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970   1st Qu.:0.0000   1st Qu.:0.5040  
 Median :0.52014   Median : 0.00   Median : 6.200   Median :0.0000   Median :0.5070  
 Mean   :0.71879   Mean   :13.62   Mean   : 7.078   Mean   :0.1538   Mean   :0.5392  
 3rd Qu.:0.57834   3rd Qu.:20.00   3rd Qu.: 6.200   3rd Qu.:0.0000   3rd Qu.:0.6050  
 Max.   :3.47428   Max.   :95.00   Max.   :19.580   Max.   :1.0000   Max.   :0.7180  
       rm             age             dis             rad              tax           ptratio     
 Min.   :8.034   Min.   : 8.40   Min.   :1.801   Min.   : 2.000   Min.   :224.0   Min.   :13.00  
 1st Qu.:8.247   1st Qu.:70.40   1st Qu.:2.288   1st Qu.: 5.000   1st Qu.:264.0   1st Qu.:14.70  
 Median :8.297   Median :78.30   Median :2.894   Median : 7.000   Median :307.0   Median :17.40  
 Mean   :8.349   Mean   :71.54   Mean   :3.430   Mean   : 7.462   Mean   :325.1   Mean   :16.36  
 3rd Qu.:8.398   3rd Qu.:86.50   3rd Qu.:3.652   3rd Qu.: 8.000   3rd Qu.:307.0   3rd Qu.:17.40  
 Max.   :8.780   Max.   :93.90   Max.   :8.907   Max.   :24.000   Max.   :666.0   Max.   :20.20  
     black           lstat           medv     
 Min.   :354.6   Min.   :2.47   Min.   :21.9  
 1st Qu.:384.5   1st Qu.:3.32   1st Qu.:41.7  
 Median :386.9   Median :4.14   Median :48.3  
 Mean   :385.2   Mean   :4.31   Mean   :44.2  
 3rd Qu.:389.7   3rd Qu.:5.12   3rd Qu.:50.0  
 Max.   :396.9   Max.   :7.44   Max.   :50.0  
> 
```