Assignment-4 Data Preparation and Analysis (CSP-571-01) Arinjay Jain(A20447307)

```
> fileUrl <- "http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.
data"
> dataFrame<-NULL
> dataFrame <- read.table(fileUrl, header=FALSE, na.strings = c('NA','?'), stringsAsFacto
rs = TRUE)
> names(dataFrame) <- c("Mpg", "Cylinders", "Displacement", "Horsepower", "Weight", "Acce
leration", "ModelYear", "Origin", "CarName")
> #checking the datatype of all columns
> sapply(dataFrame, class)
         Mpg    Cylinders Displacement   Horsepower       Weight Acceleration    ModelYea
r
   "numeric"    "integer"    "numeric"    "numeric"    "numeric"    "numeric"    "integer
"
      Origin      CarName
   "integer"     "factor"
> # Checking NA in columns
> colSums(is.na(dataFrame))
         Mpg    Cylinders Displacement   Horsepower       Weight Acceleration    ModelYea
r
           0            0            0            6            0            0
0
      Origin      CarName
           0            0
```

Found Horsepower have 6 NA values

#Here we found HorsePower have 6 missing value. Will replace it with median is more appropriate here.

```
> horsepower_med<-median(dataFrame$Horsepower, na.rm = TRUE)
> dataFrame$Horsepower[is.na(dataFrame$Horsepower)]<-horsepower_med
> head(dataFrame)
  Mpg Cylinders Displacement Horsepower Weight Acceleration ModelYear 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
                   CarName
1 chevrolet chevelle malibu
2         buick skylark 320
3        plymouth satellite
4             amc rebel sst
5               ford torino
6          ford galaxie 500
```

 # 2. Identify all of the categorical variables,

 # all of the numeric variables

 # Store it in the variables below.

 # 2 points

```
> sapply(dataFrame, class)
         Mpg    Cylinders Displacement   Horsepower       Weight Acceleration    ModelYea
r
   "numeric"    "integer"    "numeric"    "numeric"    "numeric"    "numeric"    "integer
"
      Origin      CarName
    "factor"     "factor"
```

```
> #ORIGIN, CYLINDERS, MODELYEAR are catagorical variables
> dataFrame$Origin <- as.factor(dataFrame$Origin)
> dataFrame$Cylinders <- as.factor(dataFrame$Cylinders)
> dataFrame$ModelYear <- as.factor(dataFrame$ModelYear)
```

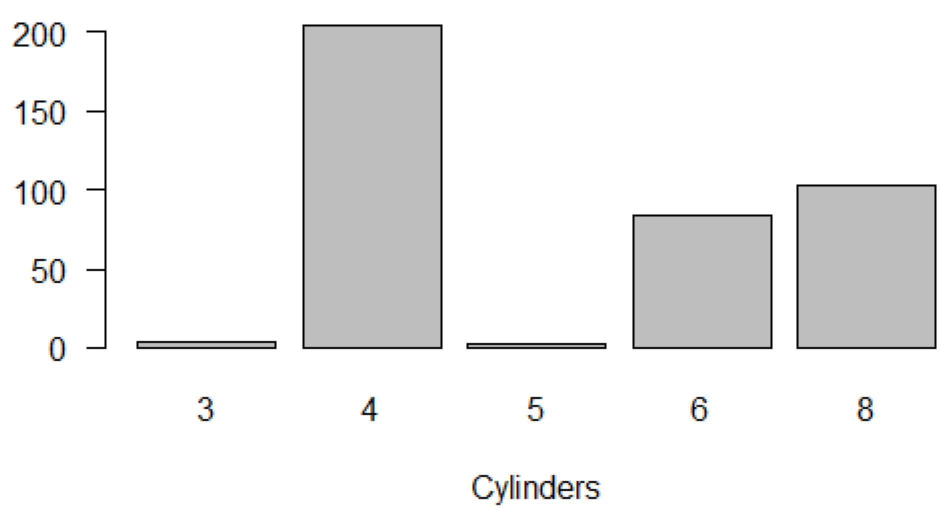
```
> sapply(dataFrame, class)
          Mpg    Cylinders Displacement   Horsepower       Weight Acceleration    ModelYear
    "numeric"     "factor"    "numeric"    "numeric"    "numeric"    "numeric"     "factor
"
       Origin      CarName
     "factor"     "factor"
> numVars<-names(dataFrame)[sapply(dataFrame, is.numeric)]
> print(numVars)
[1] "Mpg"          "Displacement" "Horsepower"   "Weight"       "Acceleration"
> catVars<-names(dataFrame)[sapply(dataFrame, is.factor)]
> print(catVars)
[1] "Cylinders" "ModelYear" "Origin"    "CarName"
```

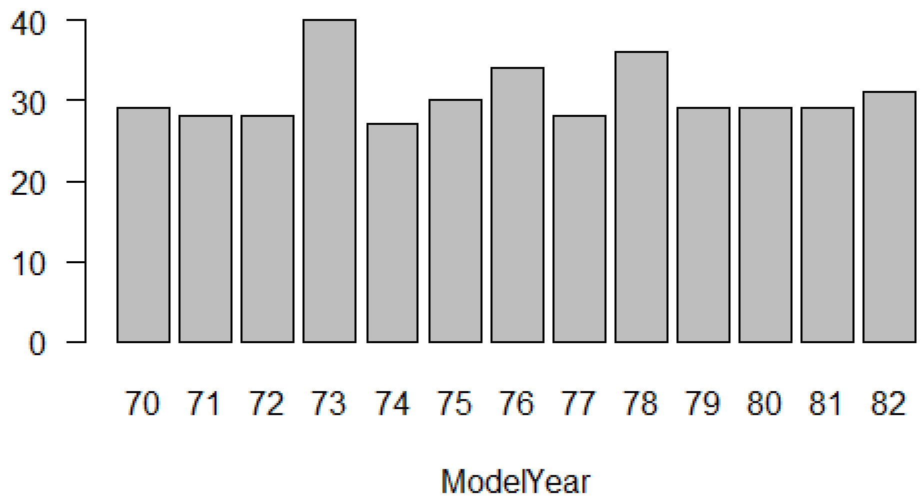
 # 3. Identify the appropriate descriptive statistics and graph for this data set.

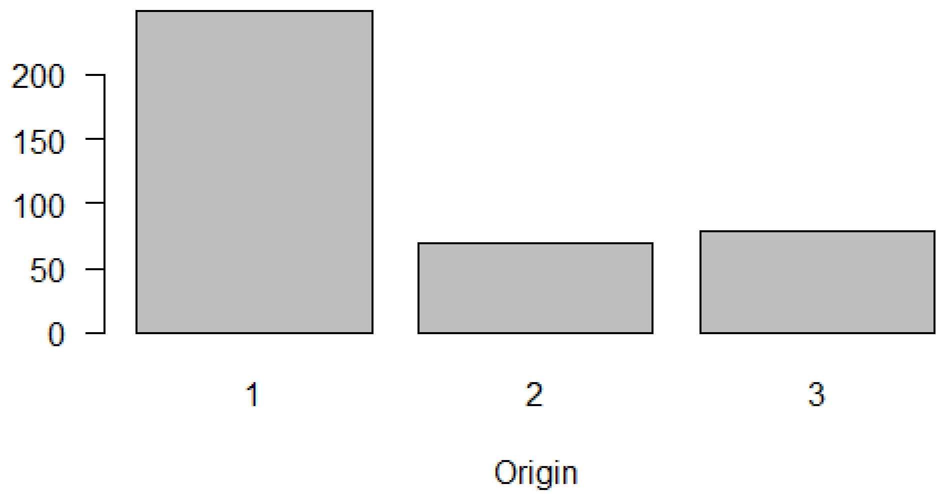
 # Execute on those and use the comments to discuss relevant relationships or insights discovered.

 # 2 points

```
> #summary of whole data set.
> summary(dataFrame)
      Mpg        Cylinders  Displacement     Horsepower        Weight      Acceleration  
 Min.   : 9.00   3:  4     Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00  
 1st Qu.:17.50   4:204     1st Qu.:104.2   1st Qu.: 76.0   1st Qu.:2224   1st Qu.:13.82  
 Median :23.00   5:  3     Median :148.5   Median : 93.5   Median :2804   Median :15.50  
 Mean   :23.51   6: 84     Mean   :193.4   Mean   :104.3   Mean   :2970   Mean   :15.57  
 3rd Qu.:29.00   8:103     3rd Qu.:262.0   3rd Qu.:125.0   3rd Qu.:3608   3rd Qu.:17.18  
 Max.   :46.60             Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80  
                                                                                         
   ModelYear    Origin            CarName   
 73     : 40   1:249   ford pinto    :  6  
 78     : 36   2: 70   amc matador   :  5  
 76     : 34   3: 79   ford maverick :  5  
 82     : 31           toyota corolla:  5  
 75     : 30           amc gremlin   :  4  
 70     : 29           amc hornet    :  4  
 (Other):198           (Other)       :369  
> for(k in catVars){
+   if(k!= colnames(dataFrame[9])){ ##not loop on Carname col
+     barplot(table(dataFrame[[k]]), xlab=k, las = 1)
+   }
+ }
```



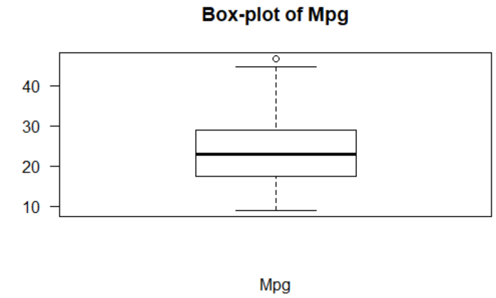


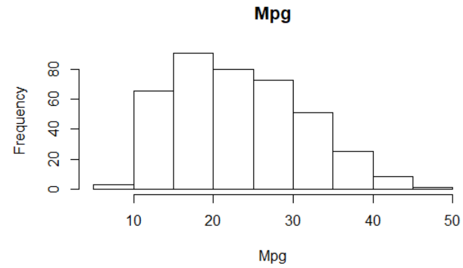


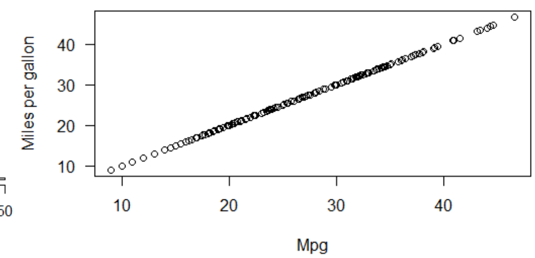
 # Results and Information from BAR Chart -
 # column cylinders has 200+ records at 4 category.
 # column Origin has 250+ records at 1 category.
 # column Model Year is almost uniformly distributed, except at 73, 76 and 78. Max records are at 73

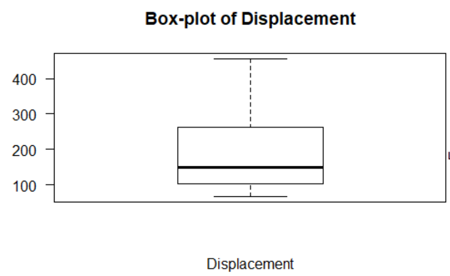
#For numeric columns I will use box-plot, histogram and plot between variables
#BOXPLOT #Histogram #Plot

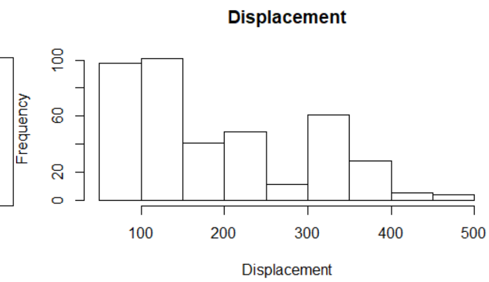
```
for (i in numVars){
+   lable <- paste("Box-plot of", i)
+   boxplot(dataFrame[[i]], main = lable, xlab = i, las = 1)
+   # histograms
+   hist(dataFrame[[i]], main = i, xlab = i)
+
+   #Plot
+   plot(y=dataFrame$Mpg, x=dataFrame[[i]], ylab = "Miles per gallon", xlab = i, las = 1)
+ }
```

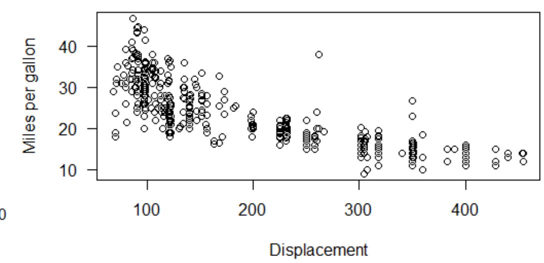

Box-plot of Mpg
Mpg

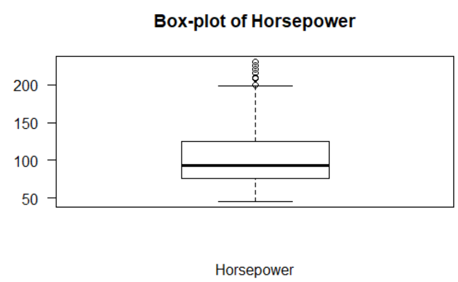

Mpg
Frequency
Mpg

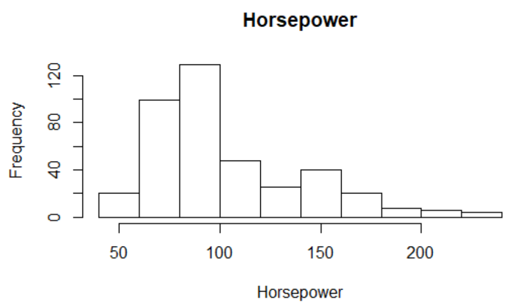

Miles per gallon
Mpg

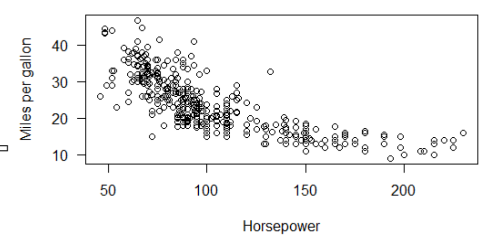

Box-plot of Displacement
Displacement

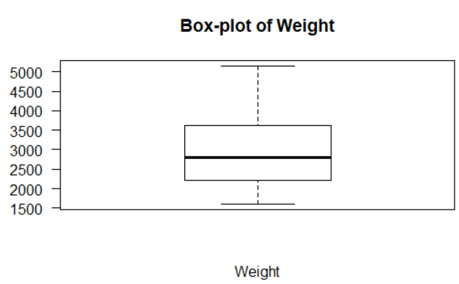

Displacement
Frequency
Displacement

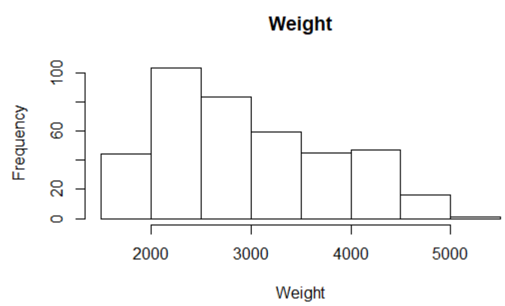

Miles per gallon
Displacement

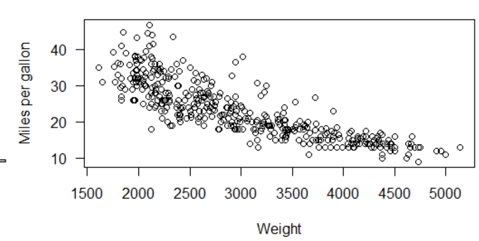

Box-plot of Horsepower
Horsepower

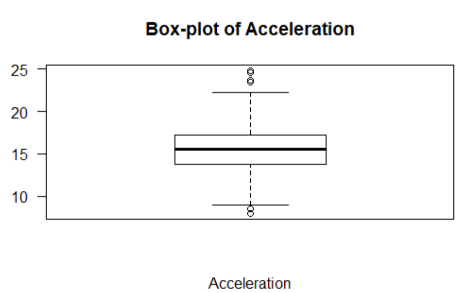

Horsepower
Frequency
Horsepower

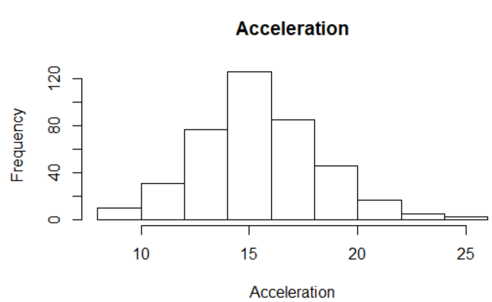

Miles per gallon
Horsepower

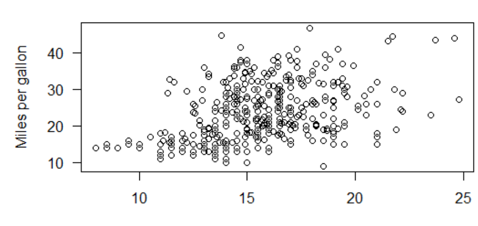

Box-plot of Weight
Weight


Weight
Frequency
Weight


Miles per gallon
Weight


Box-plot of Acceleration
Acceleration


Acceleration
Frequency
Acceleration


Miles per gallon
Acceleration

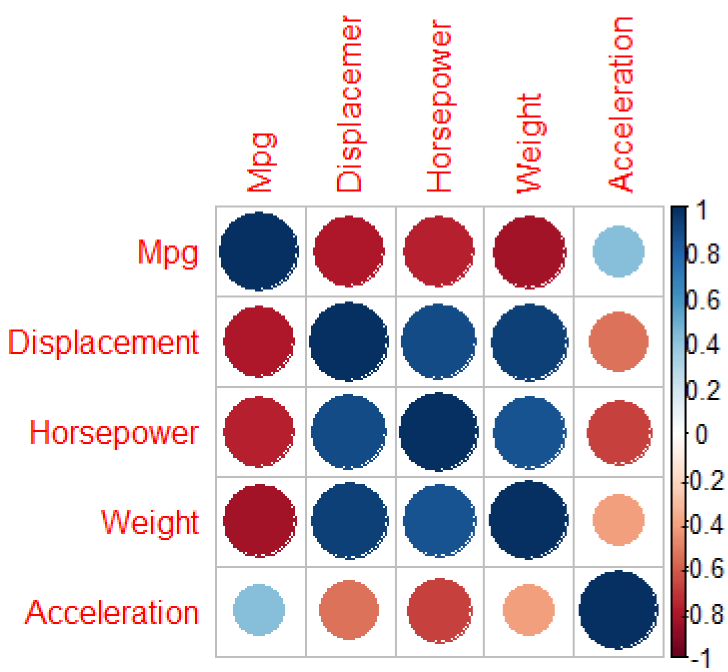
 # Results and Information from BOXPLOT -
 # 1 outlier in MPG value around 46 and a half numbers of cars have good Miles per gallon about 23.
 # many outliers cars have more than 200 horsepower.
 # weight seems perfect no outliers are present from min 1613, median 2804 to max 5140.
 # In acceleration some lower and upper outliers and mean 15.50.

 # Results and Information from histograms -
 # from all histograms we can say Acceleration is more seem like Normally Distributed.
 # and others are like left-shifted, lower values have more frequencies like positively skewed Mode<Median< Mean on the x-axis.

 # Results and Information from the plots -
 # acceleration has a positive correlation with Mpg(Miles per gallon)
 # and others (Displacement, Horsepower, and Weighthave) have negative correlation with the Mpg((Miles p er gallon) and non-linear.

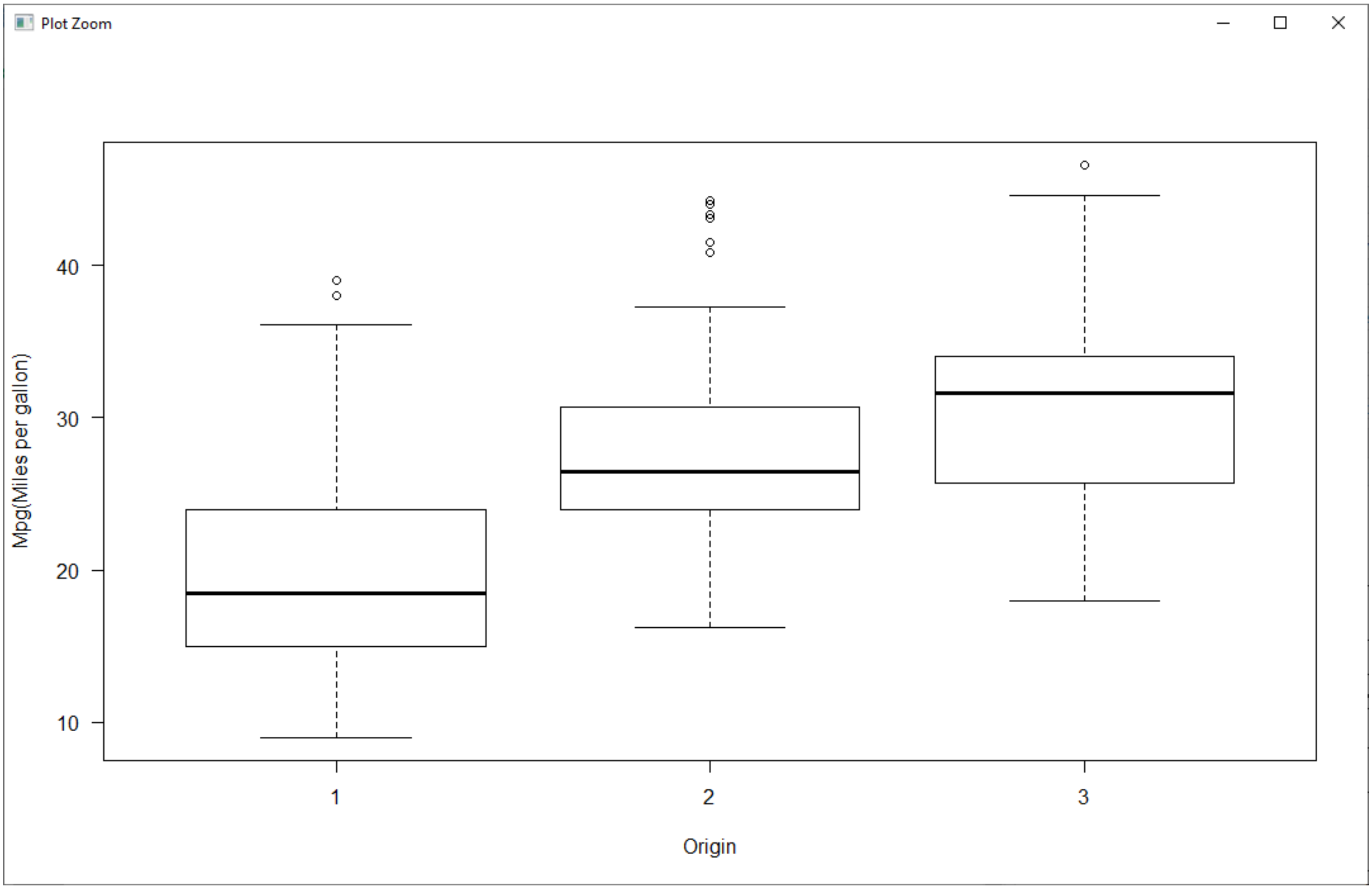
 # 4. Create a correlation matrix for all of the numeric variables.
 # 2 points

```
> corMatrix <- cor(dataFrame[numVars])
> corMatrix
                    Mpg Displacement Horsepower     Weight Acceleration
Mpg           1.0000000   -0.8042028 -0.7734532 -0.8317409    0.4202889
Displacement -0.8042028    1.0000000  0.8957782  0.9328241   -0.5436841
Horsepower   -0.7734532    0.8957782  1.0000000  0.8624424   -0.6865897
Weight       -0.8317409    0.9328241  0.8624424  1.0000000   -0.4174573
Acceleration  0.4202889   -0.5436841 -0.6865897 -0.4174573    1.0000000
> corrplot(corMatrix, method = "circle", diag = TRUE)
```



 # 5. Create a box plot of mpg versus origin
 # 2 points

```
boxplot(dataFrame$Mpg~dataFrame$Origin, xlab = 'Origin', ylab = 'Mpg(Miles per gallon)', 
las = 1)
```



 # 6. Divide the data into a train/test set (80% and 20% respectively) using stratified sampling

 # 2 points

```
> library('caret')
> set.seed(42)
> indexs <- createDataPartition(y = dataFrame$Mpg, times = 1, p = 0.8, list = FALSE)
> train_DF <- dataFrame[indexs,]
> test_DF <- dataFrame[-indexs,]
> head(train_DF)
  Mpg Cylinders Displacement Horsepower Weight Acceleration ModelYear Origin
1  18         8          307        130   3504         12.0        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
7  14         8          454        220   4354          9.0        70      1
                    CarName
1 chevrolet chevelle malibu
3        plymouth satellite
4             amc rebel sst
5               ford torino
6          ford galaxie 500
7          chevrolet impala
```

 # 7. Fit a linear model to the data using the numeric variables only. Calculate the R2 on the test set.**
 # 3 points

```
> #Liner model
> groupvars<-numVars[-1]
> # This returns the formula:
> modelFormula <- as.formula(paste('Mpg', paste(groupvars, collapse=" + "), sep=" ~ "))
> model <- lm(modelFormula, data = train_DF)# build the model
> summary(model)

Call:
lm(formula = modelFormula, data = train_DF)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.4763  -2.8329  -0.2614   2.1657  14.1051 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  45.9099180  2.6355691  17.419  < 2e-16 ***
Displacement -0.0080674  0.0073871  -1.092   0.2756    
Horsepower   -0.0430874  0.0178473  -2.414   0.0163 *  
Weight       -0.0051874  0.0008706  -5.959 6.79e-09 ***
Acceleration -0.0720721  0.1335929  -0.539   0.5899    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.143 on 316 degrees of freedom
Multiple R-squared:  0.7137,	Adjusted R-squared:   0.71 
F-statistic: 196.9 on 4 and 316 DF,  p-value: < 2.2e-16

> Mpg_pred<-predict(model, test_DF)
> #residual = predict - actual
> res<- Mpg_pred - test_DF$Mpg
> sse <- sum(res**2)
> #sst = sum((y-yhat)**2)
> sst<- sum((test_DF$Mpg-mean(test_DF$Mpg))**2)
> rSq <- 1-sse/sst
> rSq #R**2 on  test data is 0.668995
[1] 0.668995
```

 # 8. Programmatically identify and remove the non-significant variables (alpha = .05). Fit a new model with t hose variables removed.
 # Calculate the R2 on the test set with the new model. Did this improve performance?**
 # 4 points

```
> xvars1 <- rownames(summary(model)$coefficients[summary(model)$coefficients[,4]<0.05,])[
-1]
> xvars1 #significant variables P-value < (alpha = .05)
[1] "Horsepower" "Weight"    
> 
> modelFormula1 <- as.formula(paste('Mpg', paste(xvars1, collapse=" + "), sep=" ~ "))
> model1 <- lm(modelFormula1, data = train_DF)
> summary(model1)

Call:
lm(formula = modelFormula1, data = train_DF)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.0163  -2.7286  -0.2674   2.2123  13.7605 

Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.6891167  0.8591385  53.180  < 2e-16 ***
Horsepower  -0.0447472  0.0121543  -3.682 0.000272 ***
Weight      -0.0059536  0.0005483 -10.859  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.139 on 318 degrees of freedom
Multiple R-squared:  0.7125,   Adjusted R-squared:  0.7107
F-statistic:   394 on 2 and 318 DF,  p-value: < 2.2e-16

>
> Mpg_pred1<-predict(model1, test_DF)
>
> #residual = predict - actual
> res1<- Mpg_pred1 - test_DF$Mpg
> sse1 <- sum(res1**2)
> #sst = sum((y-yhat)**2)
> sst1<- sum((test_DF$Mpg-mean(test_DF$Mpg))**2)
> rSq1 <- 1-sse1/sst1
> rSq1 #R**2 on  test data is 0.6711464
[1] 0.6711464
>
> #The performance of the model does seems improve when compared to the previous model.
```

 # 9. Attempt to fit a model on all of the relevant independent variables (including carName).
 # Then calculate the R2 on a test set. You will likely encounter an error.**
 # Explain why this error occurs. Fix this error.
 # 4 points

```
> xvars2<-c(xvars1,catVars)

> xvars2
[1] "Horsepower" "Weight"     "Cylinders"  "ModelYear"  "Origin"     "CarName"

> modelFormula2 <- as.formula(paste('Mpg', paste(xvars2, collapse=" + "), sep=" ~ "))
>
> #Creating model(Name is model9 for question-9)
> model9 <- lm(modelFormula2, data = train_DF)
> summary(model9)

> Mpg_pred9<-predict(model9, test_DF)
Error in model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels
) :
  factor CarName has new levels amc concord dl, amc spirit dl, audi 100 ls, buick century
luxus (sw), buick lesabre custom, ……….
```

 # Error is due to carName variable have some value or names that are new or unseen in the training set.
 # and there is no dummy variables for the same records. Hence, when the test record
 # tries to predict the mpg for cars from test data which are not present, an error is occurred.
 # One solution: carName variable should not be considered in the model.

```
> xVars3 <- c(xvars2,catVars[which(catVars != "CarName")])
> modelFormula91 <- as.formula(paste('Mpg', paste(xVars3, collapse=" + "), sep=" ~ "))
> model91 <- lm(modelFormula91, data = train_DF)
> summary(model91)

Call:
lm(formula = modelFormula91, data = train_DF)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-6.6648 -1.6245  0.0365  1.4584 11.7346

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 31.7814353  2.0192279  15.739  < 2e-16 ***
Horsepower  -0.0282616  0.0105130  -2.688  0.00758 ** 
Weight      -0.0051487  0.0005292  -9.730  < 2e-16 ***
Cylinders4   7.0446997  1.4584647   4.830 2.18e-06 ***
Cylinders5   6.5546163  2.4970834   2.625  0.00911 ** 
Cylinders6   4.9711326  1.5391303   3.230  0.00138 ** 
Cylinders8   7.8472801  1.6695440   4.700 3.97e-06 ***
ModelYear71  0.8422658  0.8650809   0.974  0.33103    
ModelYear72 -0.7989975  0.8547878  -0.935  0.35068    
ModelYear73 -0.7830053  0.7592606  -1.031  0.30324    
ModelYear74  1.0196530  0.9137662   1.116  0.26537    
ModelYear75  0.9300840  0.9231949   1.007  0.31452    
ModelYear76  1.0612362  0.8608552   1.233  0.21863    
ModelYear77  2.5852670  0.8768896   2.948  0.00345 ** 
ModelYear78  2.6035068  0.8306218   3.134  0.00189 ** 
ModelYear79  4.8236572  0.8684570   5.554 6.15e-08 ***
ModelYear80  8.9633727  0.9261207   9.678  < 2e-16 ***
ModelYear81  6.0850953  0.9185001   6.625 1.61e-10 ***
ModelYear82  7.3244742  0.9002053   8.136 1.09e-14 ***
Origin2      1.5123700  0.5068614   2.984  0.00308 ** 
Origin3      1.4309273  0.5047034   2.835  0.00489 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.76 on 300 degrees of freedom
Multiple R-squared:  0.8794,	Adjusted R-squared:  0.8713 
F-statistic: 109.3 on 20 and 300 DF,  p-value: < 2.2e-16

> 
> 
> Mpg_pred91<-predict(model91, test_DF)
> 
> #residual = predict - actual
> res91<- Mpg_pred91 - test_DF$Mpg
> sse91 <- sum(res91**2)
> #sst = sum((y-yhat)**2)
> sst91<- sst ##always same
> rSq91 <- 1-sse91/sst91
> rSq91#R**2 on  test data is 0.8376818
[1] 0.8376818
```

Here we go we have biger R2 that means this model that the regression line perfectly fits the data**

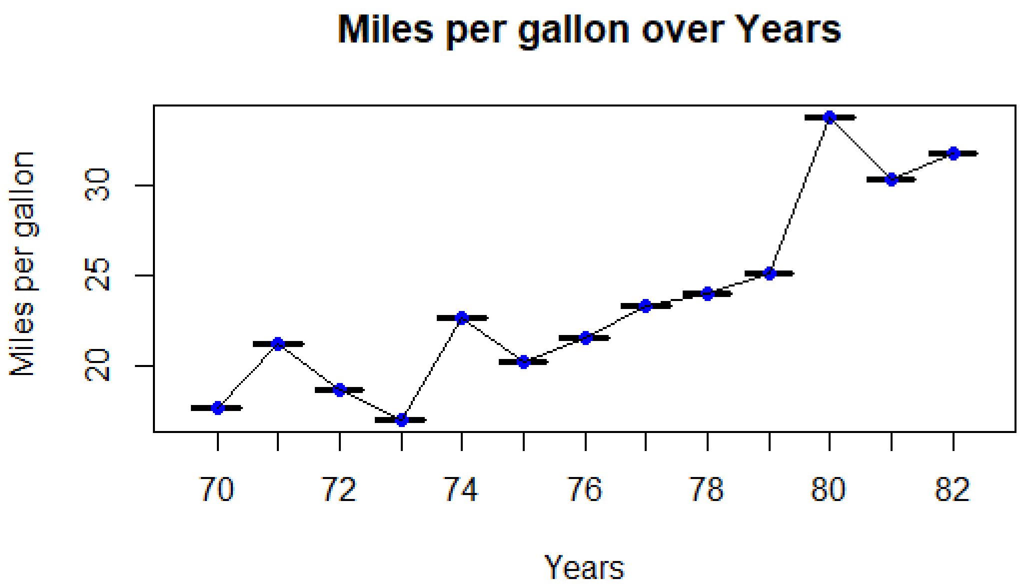
 # 10. Determine the relationship between model year and mpg.

 # Interpret this relationship.

 # Theorize why this relationship might occur.

 # 4 points

```
> average_mpg_year<- tapply(dataFrame$Mpg,dataFrame$ModelYear,mean)
> yearvalue<-unique(dataFrame$ModelYear)
> numericyear<-as.numeric(levels(yearvalue))[yearvalue]
> cor_mpg_myears<-cor(numericyear,average_mpg_year, method = "pearson")
> cor_mpg_myears #0.884
[1] 0.8839478
> 
> data= data.frame(yearvalue,average_mpg_year)
> plot(data, xlab="Years", ylab=" Miles per gallon")
> title (" Miles per gallon over Years")
> points(average_mpg_year,col="blue",pch=19)
> lines(average_mpg_year)
```



#highly positive correlation that means they have a positive increasing relationship between Mpg and Mode l years.
 # and seems logically correct because as per market demand for better Miles per gallon.(though a couple of drops are seen, the overall mpg is increasing)
#so every year companies try to give better performance in this direction that we can see in the above plot.

 # 11. Using only the variables provided, build the best linear model
 # you can (as measured by R2 on the test data)**
 # Record the value obtained in the comments below. Make sure to show all your code.
 # Record the best R2 value on the test set in the comments below.**
 # My Best R2 value: 0.8640147**
 # 4 points

```
> library(leaps)
> squ_model <- lm(Mpg ~ 1 + Cylinders + Displacement + I(Displacement^2) + Horsepower + I
(Horsepower^2) +Weight + I(Weight^2)+ Acceleration + I(Acceleration^2)+ModelYear + Origin
, data=train_DF)
> summary(squ_model)

Call:
lm(formula = Mpg ~ 1 + Cylinders + Displacement + I(Displacement^2) + 
    Horsepower + I(Horsepower^2) + Weight + I(Weight^2) + Acceleration + 
    I(Acceleration^2) + ModelYear + Origin, data = train_DF)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.6295 -1.3657 -0.0022  1.3509  9.7557 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        6.668e+01  5.743e+00  11.612  < 2e-16 ***
Cylinders4         7.885e+00  1.462e+00   5.394 1.42e-07 ***
Cylinders5         8.972e+00  2.313e+00   3.879 0.000129 ***
Cylinders6         8.148e+00  1.824e+00   4.467 1.13e-05 ***
```

```
Cylinders8           8.835e+00  2.135e+00   4.138 4.57e-05 ***
Displacement        -3.798e-02  2.282e-02  -1.664 0.097159 .
I(Displacement^2)    5.496e-05  3.943e-05   1.394 0.164443
Horsepower          -7.423e-02  4.116e-02  -1.804 0.072331 .
I(Horsepower^2)      9.255e-05  1.493e-04   0.620 0.535688
Weight              -1.413e-02  2.754e-03  -5.131 5.24e-07 ***
I(Weight^2)          1.623e-06  3.742e-07   4.337 1.99e-05 ***
Acceleration        -1.881e+00  5.334e-01  -3.528 0.000486 ***
I(Acceleration^2)    5.112e-02  1.577e-02   3.241 0.001327 **
ModelYear71          1.282e-01  8.217e-01   0.156 0.876167
ModelYear72         -2.624e-01  7.885e-01  -0.333 0.739531
ModelYear73         -7.414e-01  7.023e-01  -1.056 0.291958
ModelYear74          9.328e-01  8.514e-01   1.096 0.274119
ModelYear75          1.206e+00  8.406e-01   1.434 0.152587
ModelYear76          1.388e+00  7.887e-01   1.760 0.079431 .
ModelYear77          2.793e+00  8.080e-01   3.456 0.000628 ***
ModelYear78          3.095e+00  7.580e-01   4.083 5.73e-05 ***
ModelYear79          5.158e+00  7.972e-01   6.471 4.07e-10 ***
ModelYear80          9.126e+00  8.284e-01  11.017  < 2e-16 ***
ModelYear81          6.205e+00  8.404e-01   7.383 1.60e-12 ***
ModelYear82          7.568e+00  8.142e-01   9.295  < 2e-16 ***
Origin2              4.803e-01  5.273e-01   0.911 0.363171
Origin3              4.806e-01  5.021e-01   0.957 0.339271
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.432 on 294 degrees of freedom
Multiple R-squared:  0.9082,	Adjusted R-squared:  0.9001 
F-statistic: 111.9 on 26 and 294 DF,  p-value: < 2.2e-16

> 
> mybest_model <- step(squ_model, scope = list(lower= Mpg~1, upper= Mpg ~ 1 + Cylinders + 
Displacement + I(Displacement^2) + Horsepower + I(Horsepower^2) +Weight + I(Weight^2)+ Ac
celeration + I(Acceleration^2)+ModelYear + Origin, data=train_DF), direction = 'both')
Start:  AIC=596.29
Mpg ~ 1 + Cylinders + Displacement + I(Displacement^2) + Horsepower + 
    I(Horsepower^2) + Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear + Origin

                    Df Sum of Sq    RSS    AIC
- Origin             2      6.68 1745.3 593.53
- I(Horsepower^2)    1      2.27 1740.9 594.71
<none>                           1738.6 596.29
- I(Displacement^2)  1     11.49 1750.1 596.41
- Displacement       1     16.38 1755.0 597.30
- Horsepower         1     19.23 1757.8 597.83
- I(Acceleration^2)  1     62.12 1800.7 605.56
- Acceleration       1     73.59 1812.2 607.60
- I(Weight^2)        1    111.24 1849.8 614.20
- Cylinders          4    185.52 1924.1 620.84
- Weight             1    155.70 1894.3 621.83
- ModelYear         12   2261.76 4000.4 839.79

Step:  AIC=593.53
Mpg ~ Cylinders + Displacement + I(Displacement^2) + Horsepower + 
    I(Horsepower^2) + Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear

                    Df Sum of Sq    RSS    AIC
- I(Horsepower^2)    1      1.45 1746.7 591.79
<none>                           1745.3 593.53
- Horsepower         1     16.97 1762.2 594.63
- I(Displacement^2)  1     21.67 1766.9 595.49
+ Origin             2      6.68 1738.6 596.29
- Displacement       1     35.02 1780.3 597.90
- I(Acceleration^2)  1     69.66 1814.9 604.09
- Acceleration       1     82.34 1827.6 606.32
- I(Weight^2)        1    108.48 1853.7 610.88
- Weight             1    152.94 1898.2 618.49
- Cylinders          4    211.90 1957.2 622.31
```

```
- ModelYear        12   2388.37 4133.6 846.31

Step:  AIC=591.79
Mpg ~ Cylinders + Displacement + I(Displacement^2) + Horsepower + 
    Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear

                    Df Sum of Sq    RSS    AIC
<none>                           1746.7 591.79
+ I(Horsepower^2)    1      1.45 1745.3 593.53
+ Origin             2      5.86 1740.9 594.71
- I(Displacement^2)  1     30.80 1777.5 595.40
- Displacement       1     40.43 1787.2 597.14
- I(Acceleration^2)  1     78.52 1825.2 603.91
- Horsepower         1     85.75 1832.5 605.18
- Acceleration       1     87.56 1834.3 605.49
- I(Weight^2)        1    125.15 1871.9 612.00
- Cylinders          4    217.31 1964.0 621.43
- Weight             1    181.40 1928.1 621.51
- ModelYear         12   2438.98 4185.7 848.32
> summary(mybest_model)

Call:
lm(formula = Mpg ~ Cylinders + Displacement + I(Displacement^2) + 
    Horsepower + Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear, data = train_DF)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.6674 -1.4750  0.0407  1.3341  9.9523 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        6.743e+01  5.672e+00  11.888  < 2e-16 ***
Cylinders4         8.206e+00  1.396e+00   5.878 1.12e-08 ***
Cylinders5         9.491e+00  2.203e+00   4.307 2.25e-05 ***
Cylinders6         8.670e+00  1.742e+00   4.977 1.10e-06 ***
Cylinders8         9.392e+00  2.043e+00   4.598 6.33e-06 ***
Displacement      -5.142e-02  1.961e-02  -2.622 0.009196 ** 
I(Displacement^2)  7.726e-05  3.376e-05   2.288 0.022814 *  
Horsepower        -5.001e-02  1.310e-02  -3.818 0.000164 ***
Weight            -1.422e-02  2.560e-03  -5.554 6.21e-08 ***
I(Weight^2)        1.641e-06  3.558e-07   4.613 5.91e-06 ***
Acceleration      -1.980e+00  5.133e-01  -3.859 0.000140 ***
I(Acceleration^2)  5.460e-02  1.494e-02   3.654 0.000305 ***
ModelYear71       -2.565e-02  7.884e-01  -0.033 0.974072    
ModelYear72       -3.851e-01  7.666e-01  -0.502 0.615756    
ModelYear73       -8.135e-01  6.854e-01  -1.187 0.236167    
ModelYear74        8.292e-01  8.421e-01   0.985 0.325618    
ModelYear75        1.143e+00  8.314e-01   1.374 0.170361    
ModelYear76        1.321e+00  7.790e-01   1.696 0.090919 .  
ModelYear77        2.668e+00  7.947e-01   3.356 0.000892 ***
ModelYear78        3.013e+00  7.475e-01   4.031 7.07e-05 ***
ModelYear79        4.996e+00  7.779e-01   6.423 5.29e-10 ***
ModelYear80        9.084e+00  8.160e-01  11.132  < 2e-16 ***
ModelYear81        6.099e+00  8.273e-01   7.373 1.67e-12 ***
ModelYear82        7.402e+00  7.918e-01   9.348  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.425 on 297 degrees of freedom
Multiple R-squared:  0.9078,	Adjusted R-squared:  0.9007 
F-statistic: 127.1 on 23 and 297 DF,  p-value: < 2.2e-16

> 
> 
> predmybest <- predict(mybest_model, test_DF)
> 
> residual_mybest_Model <- predmybest - test_DF[,"Mpg"]
> SST_mybest_model <- sst   #SST never change
```

```
> SSE_mybest_model <- sum(residual_mybest_Model**2)
>
> rSq_mybest_Model <- 1-SSE_mybest_model/SST_mybest_model


> rSq_mybest_Model
[1] 0.8640147
```

 # the best model obtained using the quadratic terms has the R2 value on the test-data = 0.8640147**

```
> # the best model obtained using the quadratic terms has the R**2 value on the test-data
= 0.8640147

> # best model formula get from summary: summary(mybest_model)
> bestmodelFormula = Mpg ~ Cylinders + Displacement + I(Displacement^2) + Horsepower + We
ight + I(Weight^2) + Acceleration + I(Acceleration^2) + ModelYear
>
> # this function return the Adjusted R-squared
> adjRSquare<-function(n,k,Rsqu){
+   adjRSqu = 1- (1-Rsqu)*(n-1)/(n-k-1)
+   return(adjRSqu)
+ }
>
> #cal the Adjusted R-squared
> n = nrow(test_DF)
> k = 9 #no of parameter of above model get this from summary
> Rsqu = rSq_mybest_Model
> adj_rSq_bestmodel<-adjRSquare(n,k,Rsqu)
> adj_rSq_bestmodel
[1] 0.8457481
> #Best Adjusted R**2 without brand: 0.8457481
```

 # 12. Your boss wants to know if the

 # brand of the car will add predictive power to

 # your model. Create new variables called "brand" and "model" from the carName

 # column. Do some research to figure out how to do this.

 # Clean up the brand variable. Add the cleaned up "brand" variable to the

 # best model you built from the previous question.

 # Compare the adjusted R2 on the test data set.**

 # Best Adjusted R2 without brand variable: 0.8457481**

 # Best Adjusted R2 with brand variable: 0.9256365**

 # 4 points

```
> #Findng brand name and car model
> carName <- dataFrame$CarName
> rexp <- "^(\\w+)\\s?(.*)$"
> brand_carmodel <- data.frame(CarBrand=sub(rexp,"\\1",carName), CarModel=sub(rexp,"\\2",
carName))
> head(brand_carmodel)
   CarBrand        CarModel
1 chevrolet chevelle malibu
2     buick     skylark 320
3  plymouth       satellite
4       amc       rebel sst
5      ford          torino
6      ford     galaxie 500
> dataFrame[,'BRAND'] <- as.factor(brand_carmodel$CarBrand)
> dataFrame[,'MODEL'] <- brand_carmodel$CarModel
> tail(dataFrame)
    Mpg Cylinders Displacement Horsepower Weight Acceleration ModelYear Origin
393  27         4          151         90   2950         17.3        82      1
394  27         4          140         86   2790         15.6        82      1
```

```
395  44         4           97          52   2130          24.6          82       2
396  32         4          135          84   2295          11.6          82       1
397  28         4          120          79   2625          18.6          82       1
398  31         4          119          82   2720          19.4          82       1
             CarName     BRAND       MODEL
393 chevrolet camaro chevrolet      camaro
394  ford mustang gl      ford mustang gl
395        vw pickup        vw      pickup
396    dodge rampage     dodge     rampage
397      ford ranger      ford      ranger
398       chevy s-10     chevy        s-10


> set.seed(42)
> #library('caret')
> inTrain_new <- createDataPartition(y = dataFrame$BRAND, p = 0.8, list = FALSE)
> train_DF_new <- dataFrame[inTrain_new,]
> test_DF_new <- dataFrame[-inTrain_new,]
> stopifnot(nrow(train_DF_new) + nrow(test_DF_new) == nrow(dataFrame))
> head(train_DF_new)
  Mpg Cylinders Displacement Horsepower Weight Acceleration ModelYear Origin
1  18         8          307        130   3504         12.0        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
7  14         8          454        220   4354          9.0        70      1
                    CarName     BRAND          MODEL
1 chevrolet chevelle malibu chevrolet chevelle malibu
3        plymouth satellite  plymouth      satellite
4             amc rebel sst       amc      rebel sst
5               ford torino      ford         torino
6          ford galaxie 500      ford    galaxie 500
7          chevrolet impala chevrolet         impala
> head(test_DF_new)
   Mpg Cylinders Displacement Horsepower Weight Acceleration ModelYear Origin
2   15         8          350        165   3693         11.5        70      1
22  24         4          107         90   2430         14.5        70      2
30  27         4           97         88   2130         14.5        71      3
37  19         6          250         88   3302         15.5        71      1
38  18         6          232        100   3288         15.5        71      1
40  14         8          400        175   4464         11.5        71      1
                    CarName   BRAND            MODEL
2          buick skylark 320   buick      skylark 320
22              audi 100 ls    audi           100 ls
30             datsun pl510  datsun            pl510
37           ford torino 500    ford       torino 500
38               amc matador     amc          matador
40 pontiac catalina brougham pontiac catalina brougham


> library(leaps)
> squ_model12 <- lm(Mpg ~ 1 + Cylinders + Displacement + I(Displacement^2) + Horsepower +
I(Horsepower^2) +Weight + I(Weight^2)+ Acceleration + I(Acceleration^2)+ModelYear + Origi
n + BRAND, data=train_DF_new)
> summary(squ_model12)

Call:
lm(formula = Mpg ~ 1 + Cylinders + Displacement + I(Displacement^2) +
    Horsepower + I(Horsepower^2) + Weight + I(Weight^2) + Acceleration +
    I(Acceleration^2) + ModelYear + Origin + BRAND, data = train_DF_new)

Residuals:
   Min     1Q Median     3Q    Max
-7.245 -1.330  0.000  1.284 10.966

Coefficients: (2 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        6.556e+01  6.989e+00   9.382  < 2e-16 ***
Cylinders4         6.993e+00  1.860e+00   3.759 0.000209 ***
Cylinders5         8.878e+00  2.683e+00   3.309 0.001064 **
```

```
Cylinders6          7.718e+00  2.233e+00   3.457 0.000635 ***
Cylinders8          8.659e+00  2.545e+00   3.402 0.000771 ***
Displacement       -2.369e-02  2.641e-02  -0.897 0.370477
I(Displacement^2)   4.174e-05  4.387e-05   0.951 0.342283
```

....

....

...

```
> mybest_model12 <- step(squ_model12, scope = list(lower= Mpg~1, upper= Mpg ~ 1 + Cylinde
rs + Displacement + I(Displacement^2) + Horsepower + I(Horsepower^2) +Weight + I(Weight^2
)+ Acceleration + I(Acceleration^2)+ModelYear + Origin + BRAND, data=train_DF_new), direc
tion = 'both')
Start:  AIC=652.24
Mpg ~ 1 + Cylinders + Displacement + I(Displacement^2) + Horsepower + 
    I(Horsepower^2) + Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear + Origin + BRAND


Step:  AIC=652.24
Mpg ~ Cylinders + Displacement + I(Displacement^2) + Horsepower + 
    I(Horsepower^2) + Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear + BRAND

                    Df Sum of Sq    RSS    AIC
- Displacement       1      4.91 1657.5 651.22
- I(Displacement^2)  1      5.52 1658.1 651.34
- I(Acceleration^2)  1      9.72 1662.3 652.18
<none>                           1652.6 652.24
- Acceleration       1     17.63 1670.2 653.75
- I(Horsepower^2)    1     34.68 1687.2 657.11
- Cylinders          4     95.69 1748.3 662.87
- Horsepower         1     82.54 1735.1 666.37
- BRAND             35    487.90 2140.5 667.86
- I(Weight^2)        1     93.55 1746.1 668.46
- Weight             1    123.14 1775.7 674.02
- ModelYear         12   1884.32 3536.9 880.10

Step:  AIC=651.22
Mpg ~ Cylinders + I(Displacement^2) + Horsepower + I(Horsepower^2) + 
    Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear + BRAND

                    Df Sum of Sq    RSS    AIC
- I(Displacement^2)  1      0.62 1658.1 649.34
<none>                           1657.5 651.22
- I(Acceleration^2)  1     12.58 1670.0 651.72
+ Displacement       1      4.91 1652.6 652.24
- Acceleration       1     21.00 1678.5 653.39
- I(Horsepower^2)    1     41.81 1699.3 657.46
- Cylinders          4     96.24 1753.7 661.90
- Horsepower         1     88.26 1745.7 666.39
- BRAND             35    549.00 2206.5 675.92
- I(Weight^2)        1    211.56 1869.0 688.98
- Weight             1    310.89 1968.4 706.12
- ModelYear         12   1952.50 3610.0 884.87

Step:  AIC=649.34
Mpg ~ Cylinders + Horsepower + I(Horsepower^2) + Weight + I(Weight^2) + 
    Acceleration + I(Acceleration^2) + ModelYear + BRAND

                    Df Sum of Sq    RSS    AIC
<none>                           1658.1 649.34
- I(Acceleration^2)  1     15.02 1673.1 650.33
+ I(Displacement^2)  1      0.62 1657.5 651.22
+ Displacement       1      0.01 1658.1 651.34
- Acceleration       1     25.42 1683.5 652.38
- I(Horsepower^2)    1     54.27 1712.4 658.00
- Cylinders          4    108.05 1766.1 662.24
- Horsepower         1    101.83 1759.9 667.07
- BRAND             35    549.79 2207.9 674.13
```

```
- I(Weight^2)        1    214.82 1872.9 687.67
- Weight             1    310.36 1968.5 704.13
- ModelYear         12   1965.97 3624.1 884.16
> summary(mybest_model12)

Call:
lm(formula = Mpg ~ Cylinders + Horsepower + I(Horsepower^2) + 
    Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
    ModelYear + BRAND, data = train_DF_new)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.2376 -1.3234  0.0171  1.2002 10.9135 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        6.858e+01  6.144e+00  11.161  < 2e-16 ***
Cylinders4         6.376e+00  1.691e+00   3.772 0.000199 ***
Cylinders5         8.115e+00  2.499e+00   3.247 0.001313 ** 
Cylinders6         6.663e+00  1.760e+00   3.786 0.000188 ***
Cylinders8         7.465e+00  1.840e+00   4.057 6.48e-05 ***
Horsepower        -1.729e-01  4.223e-02  -4.095 5.57e-05 ***
I(Horsepower^2)    3.978e-04  1.331e-04   2.989 0.003053 ** 
Weight            -1.686e-02  2.359e-03  -7.148 8.03e-12 ***
I(Weight^2)        1.986e-06  3.339e-07   5.947 8.32e-09 ***
Acceleration      -1.128e+00  5.512e-01  -2.046 0.041743 *  
I(Acceleration^2)  2.585e-02  1.644e-02   1.572 0.117019    
ModelYear71        1.131e-01  8.605e-01   0.131 0.895567    
ModelYear72        1.860e-01  8.097e-01   0.230 0.818467    
ModelYear73       -8.002e-01  7.181e-01  -1.114 0.266121    
ModelYear74        9.652e-01  8.458e-01   1.141 0.254794    
ModelYear75        1.166e+00  7.958e-01   1.465 0.144025    
ModelYear76        1.311e+00  7.791e-01   1.682 0.093685 .  
ModelYear77        3.079e+00  8.225e-01   3.743 0.000222 ***
ModelYear78        3.216e+00  7.731e-01   4.159 4.28e-05 ***
ModelYear79        4.429e+00  8.429e-01   5.254 3.00e-07 ***
ModelYear80        8.915e+00  8.503e-01  10.484  < 2e-16 ***
ModelYear81        6.220e+00  8.457e-01   7.354 2.24e-12 ***
ModelYear82        7.971e+00  8.562e-01   9.310  < 2e-16 ***
BRANDaudi          1.290e+00  1.346e+00   0.959 0.338488    
BRANDbmw           8.072e-01  1.945e+00   0.415 0.678409    
BRANDbuick         8.953e-01  8.878e-01   1.008 0.314147    
BRANDcadillac      4.143e+00  1.890e+00   2.192 0.029206 *  
BRANDcapri         2.574e+00  2.595e+00   0.992 0.322191    
BRANDchevroelt     9.346e-01  2.584e+00   0.362 0.717828    
BRANDchevrolet     3.336e-01  6.867e-01   0.486 0.627522    
BRANDchevy         9.401e-02  1.570e+00   0.060 0.952294    
BRANDchrysler     -1.176e+00  1.283e+00  -0.917 0.360168    
BRANDdatsun        2.645e+00  8.774e-01   3.015 0.002814 ** 
BRANDdodge         8.829e-01  7.714e-01   1.145 0.253375    
BRANDfiat          1.585e+00  1.203e+00   1.318 0.188604    
BRANDford         -8.585e-01  6.730e-01  -1.276 0.203156    
BRANDhi           -1.145e+00  2.738e+00  -0.418 0.676130    
BRANDhonda         4.169e-01  1.071e+00   0.389 0.697420    
BRANDmaxda         4.667e-01  2.120e+00   0.220 0.825905    
BRANDmazda         2.082e+00  1.291e+00   1.613 0.107959    
BRANDmercedes      2.108e+00  1.700e+00   1.240 0.216065    
BRANDmercury      -1.267e+00  1.012e+00  -1.252 0.211580    
BRANDnissan        3.285e+00  2.608e+00   1.260 0.208863    
BRANDoldsmobile    2.799e+00  1.062e+00   2.635 0.008885 ** 
```

....

....

```
> #best model formula with brand: formula = Mpg ~ Cylinders + Horsepower + I(Horsepower^2
) + 
> # Weight + I(Weight^2) + Acceleration + I(Acceleration^2) + 
> #  ModelYear + BRAND
> 
> predmybest12 <- predict(mybest_model12, test_DF_new)
```

```
>
> residual_mybest_Model12 <- predmybest12 - test_DF_new[,"Mpg"]
> SST_mybest_model12 <- sum((test_DF_new[,"Mpg"] - mean(test_DF_new[,"Mpg"]))^2)
> SSE_mybest_model12 <- sum(residual_mybest_Model12**2)
>
> rSq_mybest_Model12 <- 1-SSE_mybest_model12/SST_mybest_model12
> rSq_mybest_Model12
[1] 0.935777
>
> #Best model R-Square with Brand: 0.935777
```

#Best model R-Square with Brand: 0.935777

```
> brandbestformula<- Mpg ~ Cylinders + Horsepower + I(Horsepower^2) +
+   Weight + I(Weight^2) + Acceleration + I(Acceleration^2) +
+   ModelYear + BRAND
>
> #cal the Adjusted R-squared
> n12 = nrow(test_DF_new)
> k12 = 9 #no of parameter of above model get this from summary
> Rsqu12 = rSq_mybest_Model12
> adj_rSq_bestmodel12<-adjRSquare(n12,k12,Rsqu12)
> adj_rSq_bestmodel12
[1] 0.9256365
> #Best Adjusted R**2 with brand: 0.9256365

# With the 'Brand' variable added the adjusted R-Squared value: 0.9256365.
# Without the 'Brand' variable added the adjusted R-Squared value from the best model: 0.
8457481.
# The adjusted R-Squared values is more with the 'BRAND' variable in the datset.
```