Assignment #3: Classification

SANDEEP DASARI **UIN-829002252**

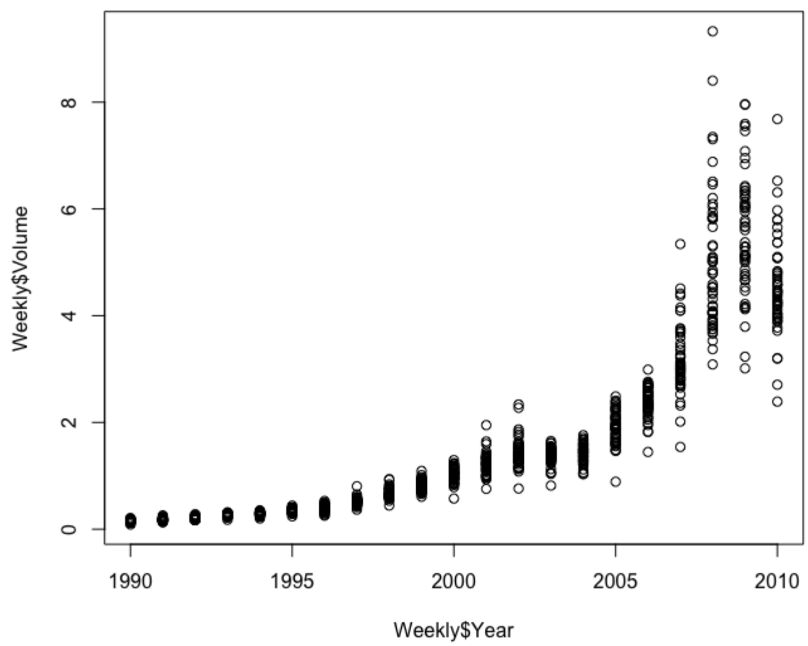
Problem 1

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

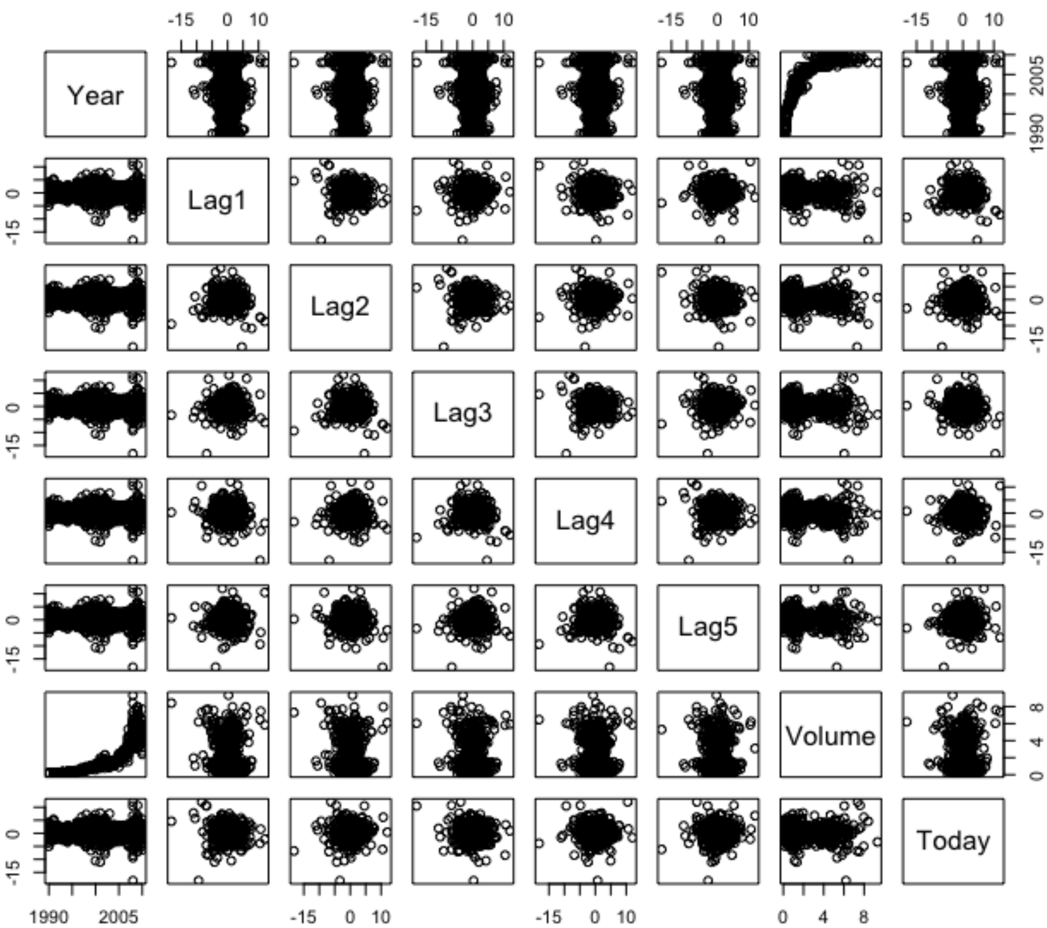
(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

```
> library("ISLR")
> attach(Weekly)
> summary(Weekly)
      Year          Lag1               Lag2               Lag3
 Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
 1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
 Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
 Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
 3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
 Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
      Lag4               Lag5              Volume            Today
 Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
 1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
 Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
 Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
 3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
 Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
 Direction
 Down:484
 Up  :605
```

> plot(Weekly$Year,Weekly$Volume



> pairs(Weekly[,-9])



```
> cor(Weekly[,-9])
             Year         Lag1        Lag2        Lag3         Lag4
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
               Lag5      Volume        Today
Year   -0.030519101  0.84194162 -0.032459894
Lag1   -0.008183096 -0.06495131 -0.075031842
Lag2   -0.072499482 -0.08551314  0.059166717
Lag3    0.060657175 -0.06928771 -0.071243639
Lag4   -0.075675027 -0.06107462 -0.007825873
Lag5    1.000000000 -0.05851741  0.011012698
Volume -0.058517414  1.00000000 -0.033077783
Today   0.011012698 -0.03307778  1.000000000
```

Result- From the above correlations we can see that Year and Volume are highly correlated with a value of 0.84. They have a positive relation.
Whereas, the other variables are not related to any other variables, like lags (1-5) are not related among themselves nor any other variables.

(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
> fit = glm(Direction~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data = Weekly, family=binomial)
> summary(fit)
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Weekly)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
```

```
Lag4       -0.02779    0.02646  -1.050   0.2937
Lag5       -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500.4
Number of Fisher Scoring iterations: 4
```

Result – Lag2 appears to be most statistically significant for this model, because the p-value is the less(<0.05).

(c) Compute the confusion matrix and performance measures (accuracy, error rate, sensitivity, specificity). Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression. Does the error rate represent the performance of logistic regression in prediction? (hint: is it training error rate or test error rate?)

```
> fit_predict = predict(fit,type= "response")
> fit_predict[1:10]
        1         2         3         4         5         6         7         8 
0.6086249 0.6010314 0.5875699 0.4816416 0.6169013 0.5684190 0.5786097 0.5151972 
        9        10 
0.5715200 0.5554287 
> pred= rep("Down",1089)
> pred[fit_predict > 0.5] = "Up"
> table(pred, Weekly$Direction)
pred   Down  Up
  Down   54  48
  Up    430 557
```

Result-

TP= 557, TN = 54, FP = 48, FN = 430

Accuracy = (TP+TN)/(N+P) = 56.10%

Error Rate = (FP+FN)/(N+P) = 43.89%

Sensitivity = (TP/P) = 92.06%

Specificity = (TN/N) = 11.15%

We can see that logistic regression accurately predicted 56.1% of the time. The training error rate is 43.9%, but this usually underestimates test error rate.

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and performance measures (accuracy, error rate, sensitivity, specificity) for the held out data (that is, the data from 2009 and 2010).

```
> train = (Weekly$Year < 2009)
> Weekly.test = Weekly[!train,]
> Direction.test = Weekly$Direction[!train]
> fit_lgreg = glm(Direction~Lag2,data=Weekly, family = binomial, subset = train)
> summary(fit_lgreg)
Call:
glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
    subset = train)
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.536  -1.264   1.021   1.091   1.368
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.20326    0.06428   3.162  0.00157 **
Lag2         0.05810    0.02870   2.024  0.04298 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1350.5  on 983  degrees of freedom
AIC: 1354.5
Number of Fisher Scoring iterations: 4
> prob_lgreg = predict(fit_lgreg,Weekly.test, type = "response")
> pred_lgreg = rep("Down", length(prob_lgreg))
> pred_lgreg[prob_lgreg > 0.5] <- "Up"
> confusion_lgreg = table(pred_lgreg,Direction.test)
> mean(pred_lgreg == Direction.test)
[1] 0.625
```

```
> confusion_lgreg
          Direction.test
pred_lgreg Down Up
      Down    9  5
      Up     34 56
```

Result – Accuracy for this prediction is (9+56)/ (14+90) = 62.5% . Error rate = 37.5%

Sensitivity =91.80%

Specificity = 20.93%

(e) Repeat (d) using LDA.

```
> library("MASS")
> fit_lda = lda(Direction ~ Lag2,data = Weekly, subset= train)
> fit_lda
Call:
lda(Direction ~ Lag2, data = Weekly, subset = train)
Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843
Group means:
            Lag2
Down -0.03568254
Up    0.26036581
Coefficients of linear discriminants:
           LD1
Lag2 0.4414162
> pred_lda = predict(fit_lda,Weekly.test)
> table(pred_lda$class, Direction.test)
      Direction.test
       Down Up
  Down    9  5
  Up     34 56
```

Result – Accuracy for this prediction is (9+56)/ (14+90) = 62.5% . Error rate = 37.5%

Sensitivity = 91.80%

Specificity = 20.93%

(f) Repeat (d) using QDA.

```
> fit_qda = qda(Direction ~ Lag2,data = Weekly, subset= train)
> fit_qda
Call:
qda(Direction ~ Lag2, data = Weekly, subset = train)
Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843
Group means:
             Lag2
Down -0.03568254
Up    0.26036581
> pred_qda = predict(fit_qda,Weekly.test)
> confusion_qda = table(pred_qda$class, Direction.test)
> confusion_qda
      Direction.test
       Down Up
  Down    0  0
  Up     43 61
```

Result – Accuracy for this prediction is (61+0)/ (0+104) = 58.65%

Error rate = (0+43)/(0+104) = 41.35%

Sensitivity = 100%

Specificity = (0/0) = %

(g) Repeat (d) using KNN with K = 1.

```
> train.data = Weekly[Weekly$Year < 2009, ]
> test.data = Weekly[Weekly$Year > 2008, ]
> set.seed(1)
> train_x = cbind(train.data$Lag2)
> train_y = cbind(train.data$Direction)
> test_x = cbind(test.data$Lag2)
> fit_knn = knn(train_x,test_x,train_y,k=1)
> table(fit_knn,test.data$Direction)
```

```
fit_knn Down Up
     1   21 30
     2   22 31
```

Result – Accuracy for this prediction is = (21 + 31) / (51 + 53) = 50%

Error Rate = (30 + 22) / (51 + 53) = 50%

Sensitivity = 50.81%

Specificity = 48.83%

(h) Which of these methods appears to provide the best results on this data?

Result – Both LDA and logistic regression have the same and highest accuracy (62.5%) among all methods.

(i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifiers.

```
> train = (Weekly$Year < 2009)
> Weekly.test = Weekly[!train,]
> Direction.test = Weekly$Direction[!train]

> fit2_logreg = glm(Direction~Lag1 + (Lag2)^2 + (Lag5)^3,data= Weekly, family= binomial,subset = train)
> mean(pred_logreg == Direction.test)
[1] 0.5576923
> confusion_logreg
          Direction.test
pred_logreg Down Up
       Down    8 11
       Up     35 50

> fit2_logreg = glm(Direction~Lag1 + (Lag2)^2 ,data= Weekly, family= binomial,subset = train)
> mean(pred_logreg == Direction.test)
[1] 0.5769231
> confusion_logreg
          Direction.test
pred_logreg Down Up
       Down    7  8
       Up     36 53

> fit2_logreg = glm(Direction~Lag1 + (Lag2)^2 + (Lag5),data= Weekly, family= binomial,subset = train)
> mean(pred_logreg == Direction.test)
[1] 0.5576923
> confusion_logreg
          Direction.test
pred_logreg Down Up
```

```
       Down    8 11
       Up     35 50

> #LDA
> fit_lda = lda(Direction~Lag1 + (Lag2)^2,data = Weekly, subset= train)
> pred_lda = predict(fit_lda,Weekly.test)
> table(pred_lda$class, Direction.test)
      Direction.test
       Down Up
  Down    7  8
  Up     36 53

> #QDA
> fit_qda = qda(Direction ~ Lag1 + (Lag2)^2 , data = Weekly, subset= train)
> pred_qda = predict(fit_qda,Weekly.test)
> table(pred_qda$class, Direction.test)
      Direction.test
       Down Up
  Down    7 10
  Up     36 51

> #KNN =2
> train.data = Weekly[Weekly$Year < 2009, ]
> test.data = Weekly[Weekly$Year > 2008, ]
> set.seed(1)
> train_x = cbind(train.data$Lag2)
> train_y = cbind(train.data$Direction)
> test_x = cbind(test.data$Lag2)
> fit_knn = knn(train_x,test_x,train_y,k=2)
> table(fit_knn,test.data$Direction)

fit_knn Down Up
      1   19 27
      2   24 34
> #KNN = 5
> fit_knn = knn(train_x,test_x,train_y,k=5)
> table(fit_knn,test.data$Direction)
fit_knn Down Up
      1   16 21
      2   27 40

#KNN = 4
> fit_knn = knn(train_x,test_x,train_y,k=4)
> table(fit_knn,test.data$Direction)
fit_knn Down Up
      1   20 17
```

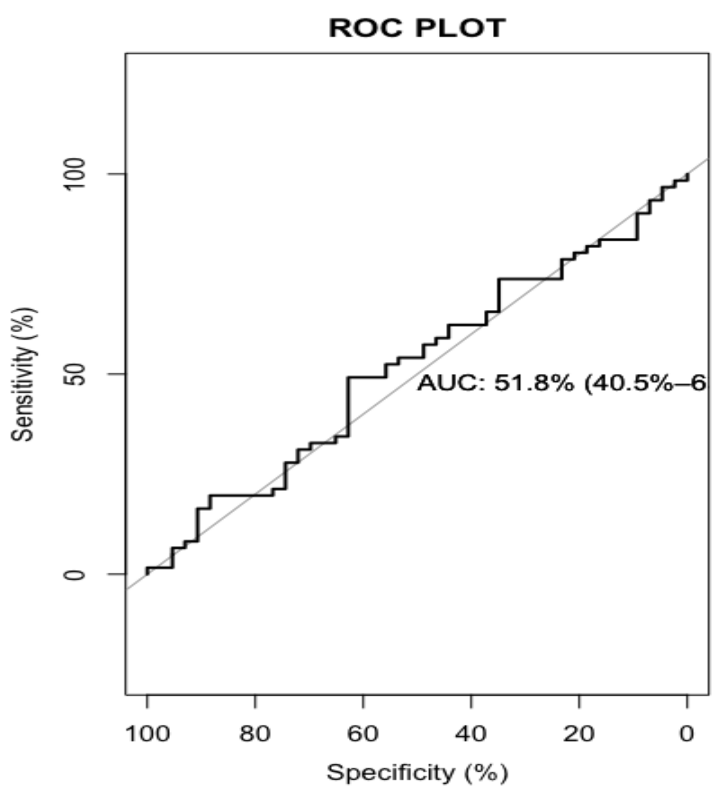
2   23 44

Result – After trying out some models with different combinations, logistic regression and LDA for the model,Direction~Lag1 + (Lag2)^2 gives the best fit in terms of accuracy (~ 57.69) after Direction ~ Lag2 (~62.5%) .
Both LDA and logistic regression give similar results for the above selected models.
For KNN, accuracy is highest for k=4. (64/104) = 61.5%.

Problem 2
Perform ROC analysis and present the results for logistic regression and LDA used for the best model chosen in Question 1(i).

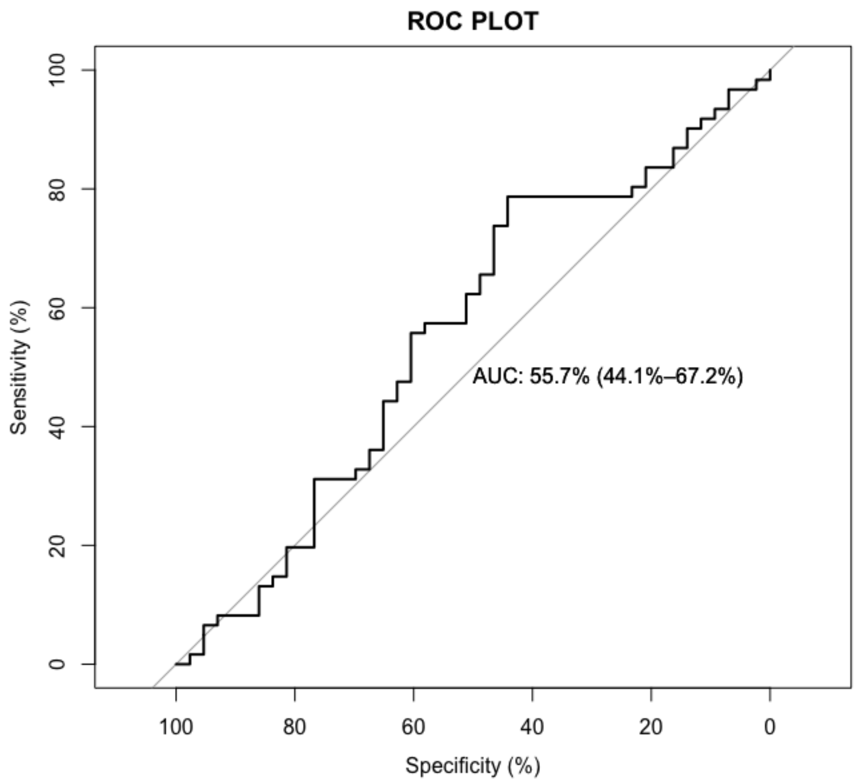
<u>*For Logistic Regression-*</u>

```
> install.packages("pROC")
> library(pROC)
> roc(Weekly.test$Direction, prob_logreg, percent=TRUE, plot=TRUE,
  ci=TRUE, print.auc = TRUE,main = "ROC PLOT")
```



For LDA-

```
> roc(Weekly.test$Direction, pred_lda$posterior[,2], percent=TRUE, plot=TRUE,
   ci=TRUE,print.auc = TRUE, main = "ROC PLOT")
```



Result- We can see the plotted ROC characteristics for *Sensitivity vs Specificity* for LDA and logistic regression. The curve mostly falls on the "no information" classifier line.

Problem 3

In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.

(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median( ) function. Note that you may find it helpful to use the data.frame( ) function to create a single data set containing both mpg01 and the other Auto variables.

```
> attach(Auto)
> mpg01 <- rep(0, length(mpg))
> mpg01[mpg > median(mpg)] <- 1
> Auto <- data.frame(Auto, mpg01)
> head(Auto)
  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 mpg01
1 chevrolet chevelle malibu     0
2         buick skylark 320     0
3        plymouth satellite     0
4             amc rebel sst     0
5               ford torino     0
6          ford galaxie 500     0
```

(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and Boxplots may be useful tools to answer this question. Describe your findings.

```
> summary(Auto)
      mpg          cylinders      displacement     horsepower        weight      acceleration
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804   Median :15.50
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean   :15.54
```

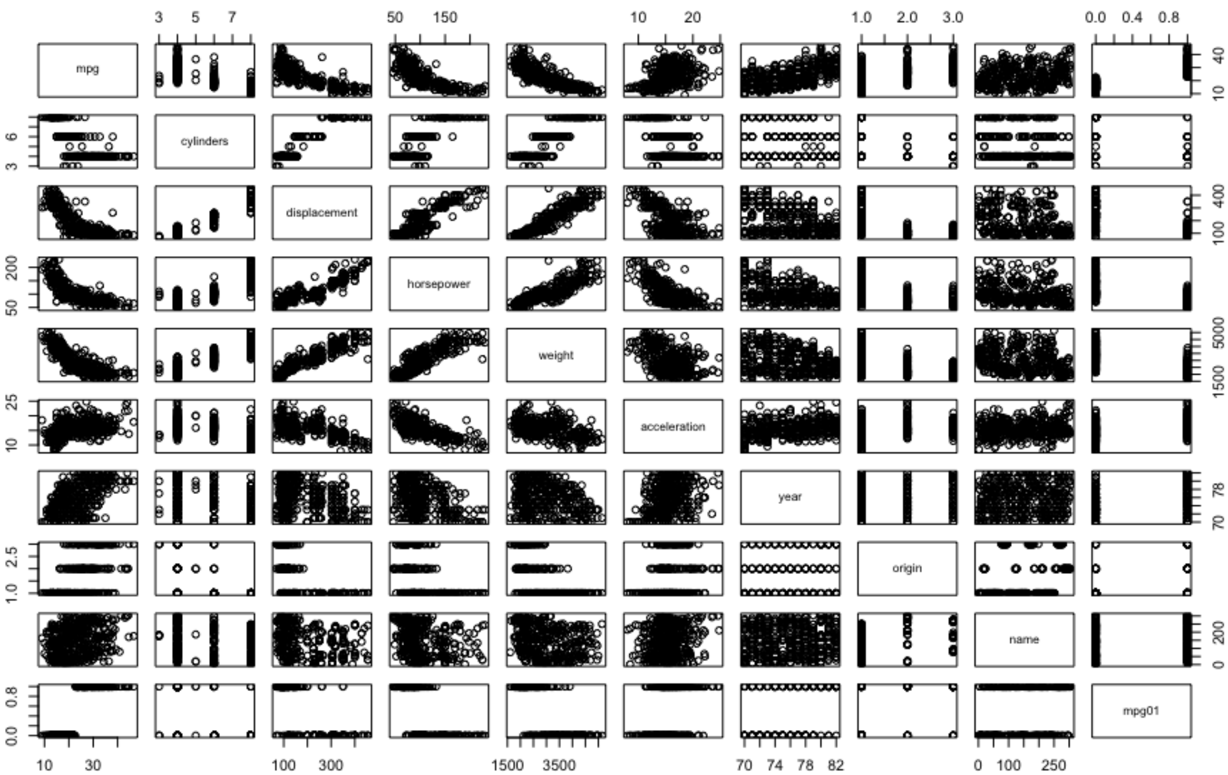
```
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80

      year           origin                      name         mpg01
 Min.   :70.00   Min.   :1.000   amc matador       :  5   Min.   :0.0
 1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5   1st Qu.:0.0
 Median :76.00   Median :1.000   toyota corolla    :  5   Median :0.5
 Mean   :75.98   Mean   :1.577   amc gremlin       :  4   Mean   :0.5
 3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4   3rd Qu.:1.0
 Max.   :82.00   Max.   :3.000   chevrolet chevette:  4   Max.   :1.0
                                 (Other)           :365
> cor(Auto[,-9])
                    mpg  cylinders displacement horsepower     weight acceleration       year     origin
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285  0.5805410  0.5652088
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834 -0.3456474 -0.5689316
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -0.3698552 -0.6145351
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -0.4163615 -0.4551715
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -0.3091199 -0.5850054
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000  0.2903161  0.2127458
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161  1.0000000  0.1815277
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458  0.1815277  1.0000000
mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215  0.4299042  0.5136984
                  mpg01
mpg           0.8369392
cylinders    -0.7591939
displacement -0.7534766
horsepower   -0.6670526
weight       -0.7577566
acceleration  0.3468215
year          0.4299042
origin        0.5136984
mpg01         1.0000000
```

```
> pairs(Auto)
```

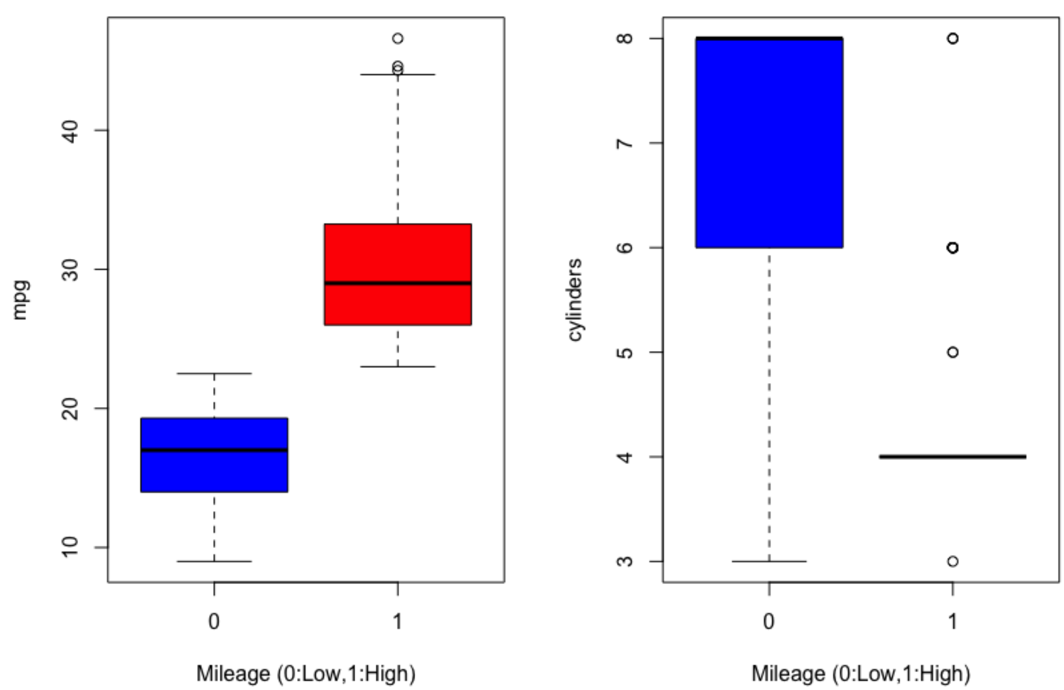
<u>To create boxplots</u>

```
> par(mfrow = c(1,2))
> for(i in 1:(ncol(Auto)-2)) {
boxplot(Auto[ ,i] ~ as.factor(Auto$mpg01), xlab = "Mileage (0:Low,1:High)",ylab = colnames(Auto)[i], col = c("blue","red"))
}
```

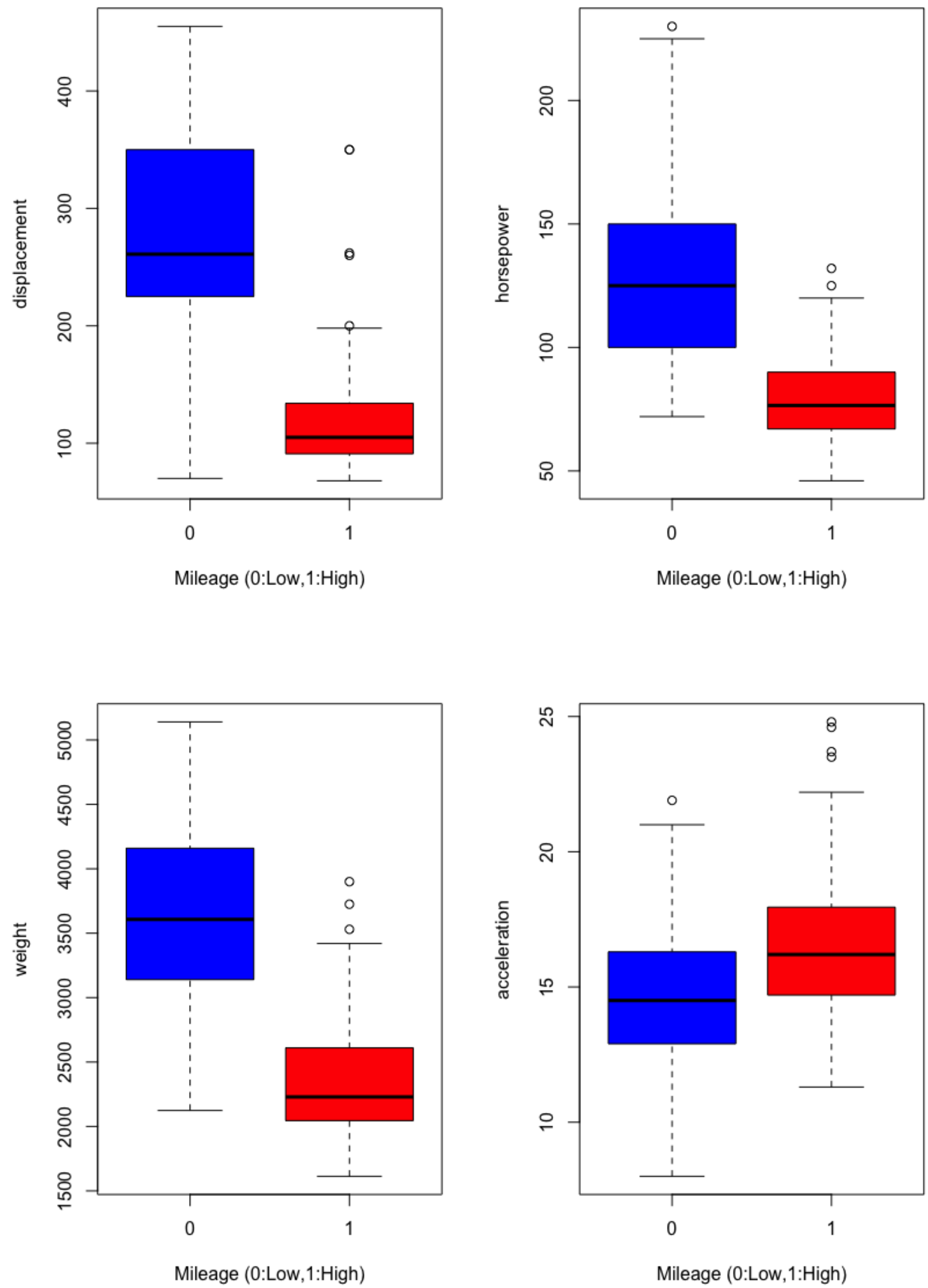
Result – From the correlation matrix we can see that mpg01 has a weak positive correlation with acceleration, origin and year.
From the correlation matrix we can also see that mpg has a strong negative correlation with weight, displacement, horsepower and cylinders.

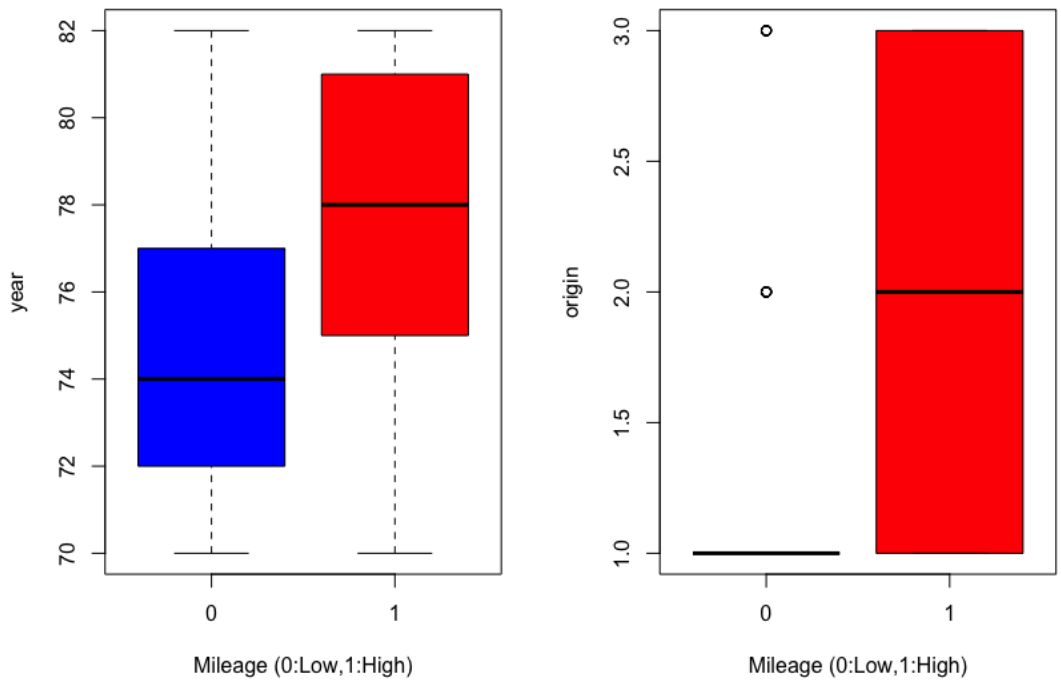


mpg
cylinders
displacement
horsepower
weight
acceleration
year
origin
name
mpg01



mpg
Mileage (0:Low,1:High)
cylinders
Mileage (0:Low,1:High)


displacement
100
200
300
400
0
1
Mileage (0:Low,1:High)
horsepower
50
100
150
200
0
1
Mileage (0:Low,1:High)
weight
1500
2000
2500
3000
3500
4000
4500
5000
0
1
Mileage (0:Low,1:High)
acceleration
10
15
20
25
0
1
Mileage (0:Low,1:High)



(c) Split the data into a training set and a test set.

```
> splitAuto <- sample(1: dim(Auto)[1], size = dim(Auto)[1]*0.75)
> Auto_train <- Auto[splitAuto, ]
> Auto_test <- Auto[- splitAuto, ]
> dim(Auto_test)
[1] 98 10
> dim(Auto_train)
[1] 294  10
```

(d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
> fit_lda =  lda(mpg01 ~ horsepower + cylinders + weight + displacement, data = Auto_train)
> summary(fit_lda)
        Length Class  Mode
prior   2      -none- numeric
counts  2      -none- numeric
means   8      -none- numeric
scaling 4      -none- numeric
lev     2      -none- character
```

```
svd     1      -none- numeric
N       1      -none- numeric
call    3      -none- call
terms   3      terms  call
xlevels 0      -none- list
> problda <- predict(fit_lda, Auto_test, type = "response")
> predlda = problda$class
> table(predlda,Auto_test$mpg01)
predlda  0  1
      0 46  2
      1  7 43
```

Result – Test Accuracy = 89/98 = 90.8%
Test Error rate = 9.18%

(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
> fit_qda =  qda(mpg01 ~ horsepower + cylinders + weight + displacement, data = Auto_train)
> summary(fit_qda)
        Length Class  Mode
prior    2     -none- numeric
counts   2     -none- numeric
means    8     -none- numeric
scaling 32     -none- numeric
ldet     2     -none- numeric
lev      2     -none- character
N        1     -none- numeric
call     3     -none- call
terms    3     terms  call
xlevels  0     -none- list
> probqda <- predict(fit_qda, Auto_test, type = "response")
> predqda = probqda$class
> table(predqda,Auto_test$mpg01)
predqda  0  1
      0  46  3
      1   7 42
```

Result – Test Accuracy = 88/98 = 89.97%
Test Error rate = 10.03%

(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
> fitlgreg <- glm(mpg01 ~ horsepower + cylinders + weight + displacement, family =
 "binomial", data = Auto_train)
> summary(fitlgreg)
Call:
glm(formula = mpg01 ~ horsepower + cylinders + weight + displacement,
    family = "binomial", data = Auto_train)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5215  -0.2504   0.1457   0.3770   3.2171
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  10.6023845  1.8554166   5.714  1.1e-08 ***
horsepower   -0.0381453  0.0160081  -2.383   0.0172 *
cylinders    -0.0194600  0.4015156  -0.048   0.9613
weight       -0.0011983  0.0007612  -1.574   0.1155
displacement -0.0204199  0.0093596  -2.182   0.0291 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
  Null deviance: 407.35  on 293  degrees of freedom
Residual deviance: 156.44  on 289  degrees of freedom
AIC: 166.44
Number of Fisher Scoring iterations: 7
> problgreg <- predict(fitlgreg, Auto_test, type = "response")
> predlgreg <- rep(0, dim(Auto_test)[1])
> predlgreg [problgreg > 0.5] = 1
> table (predlgreg, Auto$mpg01[-splitAuto])
predlgreg  0   1
        0  38  6
        1 10 44
```

Result – Test Accuracy = 82/98 = 83.67%
Test Error rate = 16.32%

(*g*) ***Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?***

For K=1

```
> set.seed(1)
> train_x = cbind(Auto_train$horsepower,Auto_train$cylinders,Auto_train$weight,Auto_train$displacement)
> train_y = cbind(Auto_train$mpg01)
> test_x = cbind(Auto_test$horsepower,Auto_test$cylinders,Auto_test$weight,Auto_test$displacement)
> fit_knn = knn(train_x,test_x,train_y,k=1)
> table(fit_knn,Auto_test$mpg01)
fit_knn  0  1
      0 40  6
      1  8 44
```

Error rate = 14.28%

For K=2

```
> fit_knn = knn(train_x,test_x,train_y,k=2)
> table(fit_knn,Auto_test$mpg01)
fit_knn  0  1
      0 40  5
      1  8 45
```

Error rate = 13.26%

For K=3

```
> fit_knn = knn(train_x,test_x,train_y,k=3)
> table(fit_knn,Auto_test$mpg01)
fit_knn  0  1
      0 39  5
      1  9 45
```

Error rate = 14.28%

For K=4

```
> fit_knn = knn(train_x,test_x,train_y,k=4)
> table(fit_knn,Auto_test$mpg01)
fit_knn  0  1
      0 38  5
```

1 10 45

Error rate = 15.30%

For K=5

```
> fit_knn = knn(train_x,test_x,train_y,k=5)
> table(fit_knn,Auto_test$mpg01)
```

```
fit_knn  0  1
      0 39  3
      1  9 47
```

Error rate = 12.24%

Result - After trying out the model for different values of k, the test error rate is minimum for k = 5 at 12.24%.