Linga Sai Yuvesh Venketa, Kotiala

16242113

ISL LAB-3

2. 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 chapters 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?

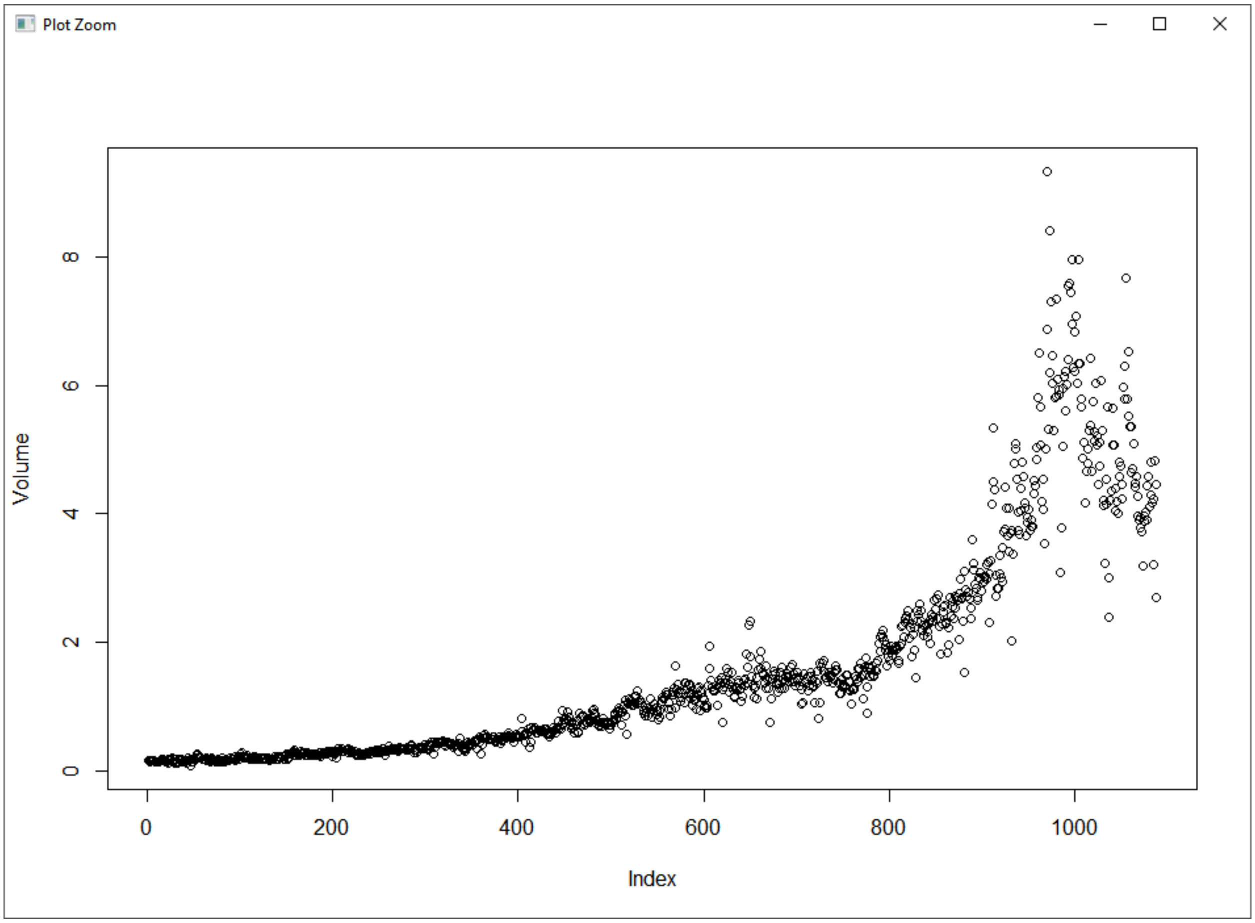
Ans.

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

Weekly percentage returns for the S&P 500 stock index between 1990 and 2010. A data frame in ISLR.

```
> Weekly
     Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
1    1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
2    1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
3    1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
4    1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
5    1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
6    1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
7    1990 -1.372  1.178  0.712  3.514 -2.576 0.1517220  0.807        Up
8    1990  0.807 -1.372  1.178  0.712  3.514 0.1323100  0.041        Up
9    1990  0.041  0.807 -1.372  1.178  0.712 0.1439720  1.253        Up
10   1990  1.253  0.041  0.807 -1.372  1.178 0.1336350 -2.678      Down
11   1990 -2.678  1.253  0.041  0.807 -1.372 0.1490240 -1.793      Down
12   1990 -1.793 -2.678  1.253  0.041  0.807 0.1357900  2.820        Up
13   1990  2.820 -1.793 -2.678  1.253  0.041 0.1398980  4.022        Up
14   1990  4.022  2.820 -1.793 -2.678  1.253 0.1643420  0.750        Up
15   1990  0.750  4.022  2.820 -1.793 -2.678 0.1756480 -0.017      Down
16   1990 -0.017  0.750  4.022  2.820 -1.793 0.1634700  2.420        Up
17   1990  2.420 -0.017  0.750  4.022  2.820 0.1726250 -1.225      Down
18   1990 -1.225  2.420 -0.017  0.750  4.022 0.1684460  1.171        Up
19   1990  1.171 -1.225  2.420 -0.017  0.750 0.1552920 -2.061      Down
20   1990 -2.061  1.171 -1.225  2.420 -0.017 0.1433920  0.729        Up
21   1990  0.729 -2.061  1.171 -1.225  2.420 0.1405540  0.112        Up
22   1990  0.112  0.729 -2.061  1.171 -1.225 0.1250750  2.480        Up
23   1990  2.480  0.112  0.729 -2.061  1.171 0.1716040 -1.552      Down
24   1990 -1.552  2.480  0.112  0.729 -2.061 0.1669560 -2.259      Down
25   1990 -2.259 -1.552  2.480  0.112  0.729 0.1717180 -2.428      Down
```

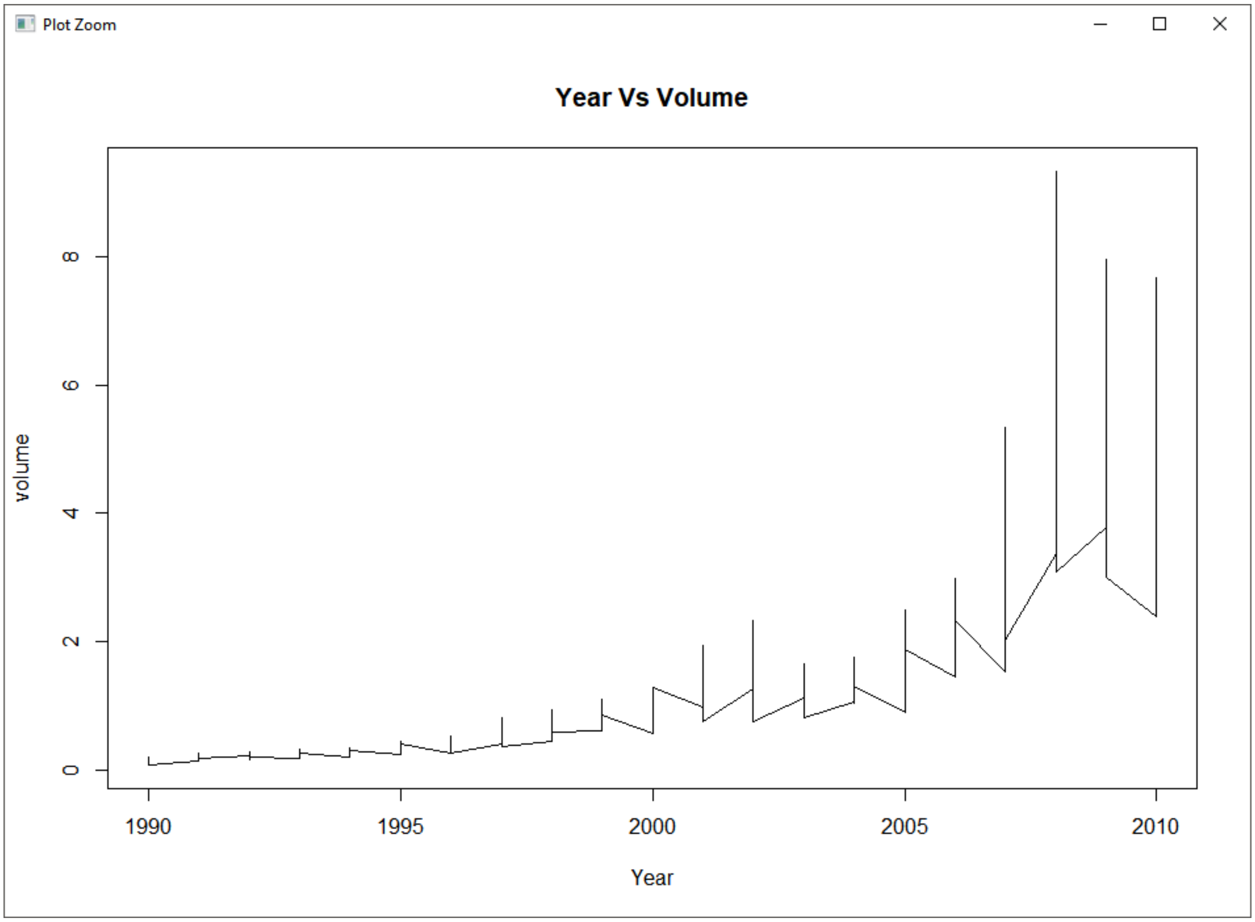
```
> cor(Weekly[, -9])
              Year         Lag1        Lag2        Lag3         Lag4         Lag5      Volume
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923 -0.030519101  0.84194162
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876 -0.008183096 -0.06495131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



The correlations between the "lag" variables and today's returns are close to zero. The only substantial correlation is between "Year" and "Volume". When we plot "Volume", we see that it is increasing over time.

Between number of up's and down's

Year v/s Direction



(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?

Ans.

```
> fit.glm <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = bin
omial)
> summary(fit.glm)

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

> 
```

It would seem that “Lag2” is the only predictor statistically significant as its p-value is less than 0.05.

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

Ans.

```
> probs <- predict(fit.glm, type = "response")
> pred.glm <- rep("Down", length(probs))
> pred.glm[probs > 0.5] <- "Up"
> table(pred.glm, Direction)
        Direction
pred.glm Down  Up
    Down   54  48
    Up    430 557
```

We may conclude that the percentage of correct predictions on the training data is (54+557)/1089 which is equal to 56.1065197%. In other words, 43.8934803% is the training error rate, which is often overly optimistic. We could also say that for weeks when the market goes up, the model is right 92.0661157% of the time (557/(48+557)). For weeks when the market goes down, the model is right only 11.1570248% of the time (54/(54+430)).

(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 the overall fraction of correct predictions for the held-out data (that is, the data from 2009 to 2010).

```
> train <- (Year < 2009)
> Weekly.20092010 <- Weekly[!train, ]
> Direction.20092010 <- Direction[!train]
> fit.glm2 <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)
> summary(fit.glm2)

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
```

```
> probs2 <- predict(fit.glm2, Weekly.20092010, type = "response")
> pred.glm2 <- rep("Down", length(probs2))
> pred.glm2[probs2 > 0.5] <- "Up"
> table(pred.glm2, Direction.20092010)
         Direction.20092010
pred.glm2 Down Up
     Down    9  5
     Up     34 56
```

In this case, we may conclude that the percentage of correct predictions on the test data is (9+56)/104 which is equal to 62.5%. In other words, 37.5% is the test error rate. We could also say that for weeks when the market goes up, the model is right 91.8032787% of the time (56/(56+5)). For weeks when the market goes down, the model is right only 20.9302326% of the time (9/(9+34)).

(e) Repeat (d) using LDA.

Ans.

Here we use “MASS” library to use the LDA and QDA

```
> 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.20092010
> table(pred.lda$class, Direction.20092010)
      Direction.20092010
       Down Up
  Down    9  5
  Up     34 56
```

In this case, we may conclude that the percentage of correct predictions on the test data is 62.5%. In other words, 37.5% is the test error rate. We could also say that for weeks when the market goes up, the model is right 91.8032787% of the time. For weeks when the market goes down, the model is right only 20.9302326% of the time. These results are very close to those obtained with the logistic regression model which is not surprising.

(f) Repeat (d) using QDA.

Ans.

```
> 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.20092010)
> table(pred.qda$class, Direction.20092010)
      Direction.20092010
       Down Up
  Down    0  0
  Up     43 61
```

In this case, we may conclude that the percentage of correct predictions on the test data is 58.6538462%. In other words, 41.3461538% is the test error rate. We could also say that for weeks when the market goes up, the model is right 100% of the time. For weeks when the market goes down, the model is right only 0% of the time. We may note, that QDA achieves a correctness of 58.6538462% even though the model chooses “Up” the whole time!

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

Ans.

```
> library(class)
> train.X <- as.matrix(Lag2[train])
> test.X <- as.matrix(Lag2[!train])
> train.Direction <- Direction[train]
> set.seed(1)
> pred.knn <- knn(train.X, test.X, train.Direction, k = 1)
> table(pred.knn, Direction.20092010)
        Direction.20092010
pred.knn Down Up
    Down   21 30
    Up     22 31
```

In this case, we may conclude that the percentage of correct predictions on the test data is 50%. In other words, 50% is the test error rate. We could also say that for weeks when the market goes up, the model is right 50.8196721% of the time. For weeks when the market goes down, the model is right only 48.8372093% of the time.

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

Ans.

If we compare the test error rates, we see that logistic regression and LDA have the minimum error rates, followed by QDA and KNN.

(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 classifier.

Ans.

```
> Lag2:Lag3
[1]  1.572  0.572 -0.428 -1.428 -2.428 -3.428
Warning messages:
1: In Lag2:Lag3 :
  numerical expression has 1089 elements: only the first used
2: In Lag2:Lag3 :
  numerical expression has 1089 elements: only the first used
```

```
> fit.glm3 <- glm(Direction ~ Lag2:Lag1, data = Weekly, family = binomial, subset = train)
> probs3 <- predict(fit.glm3, Weekly.20092010, type = "response")
> pred.glm3 <- rep("Down", length(probs3))
> pred.glm3[probs3 > 0.5] = "Up"
> table(pred.glm3, Direction.20092010)
         Direction.20092010
pred.glm3 Down Up
     Down    1  1
     Up     42 60
```

```
> mean(pred.glm3 == Direction.20092010)
[1] 0.5865385
```

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 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.

Ans.

>attach(Auto)

>mpg01 <- rep(0, length(mpg))

>mpg01[mpg > median(mpg)] <- 1

>Auto <- data.frame(Auto, mpg01)

```
> attach(Auto)
The following objects are masked _by_ .GlobalEnv:

    mpg01, name

The following objects are masked from Auto (pos = 3):

    acceleration, cylinders, displacement, horsepower, mpg, name, origin, weight, year

> mpg01 <- rep(0, length(mpg))
> mpg01[mpg > median(mpg)] <- 1
> Auto <- data.frame(Auto, mpg01)
```

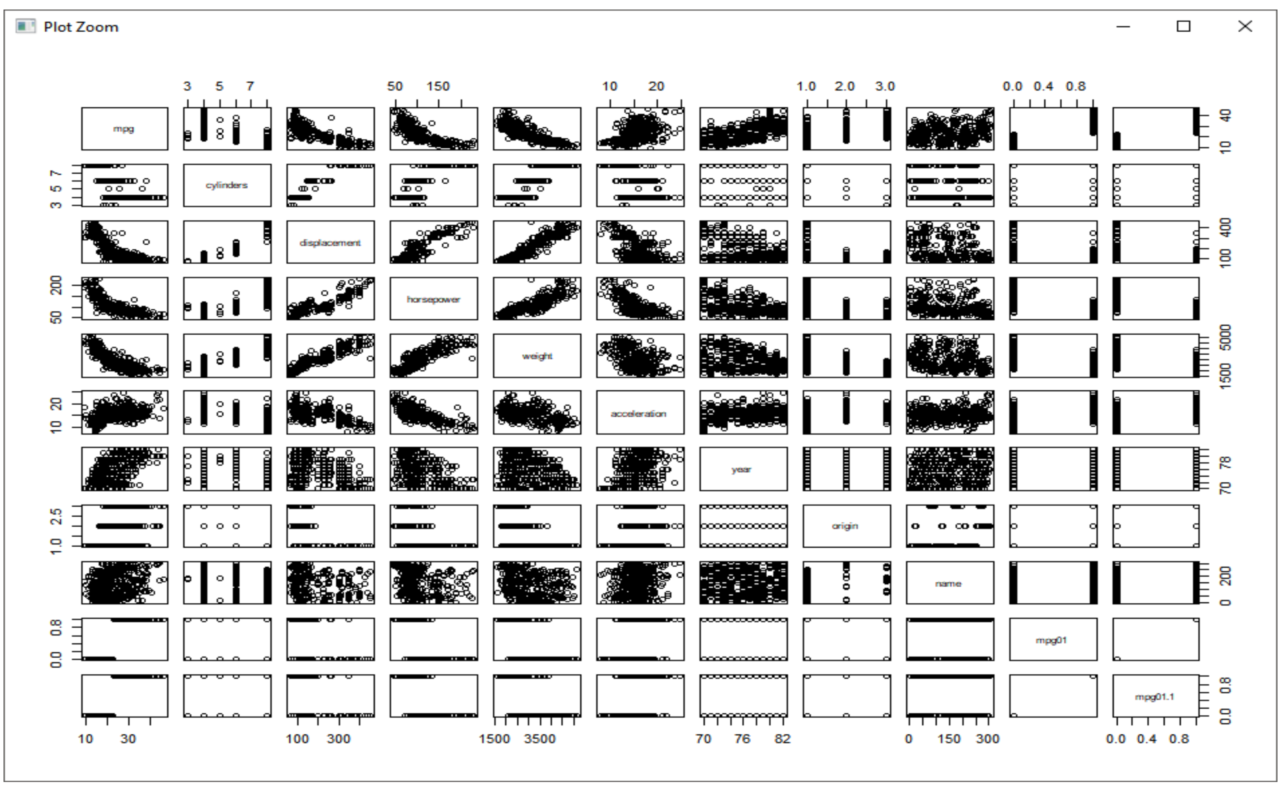
(b) Explore the data graphically in order to investigate the association between "mpg01" and the other features. Which of the other features seem most likely to be useful in predicting "mpg01"? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

Ans.

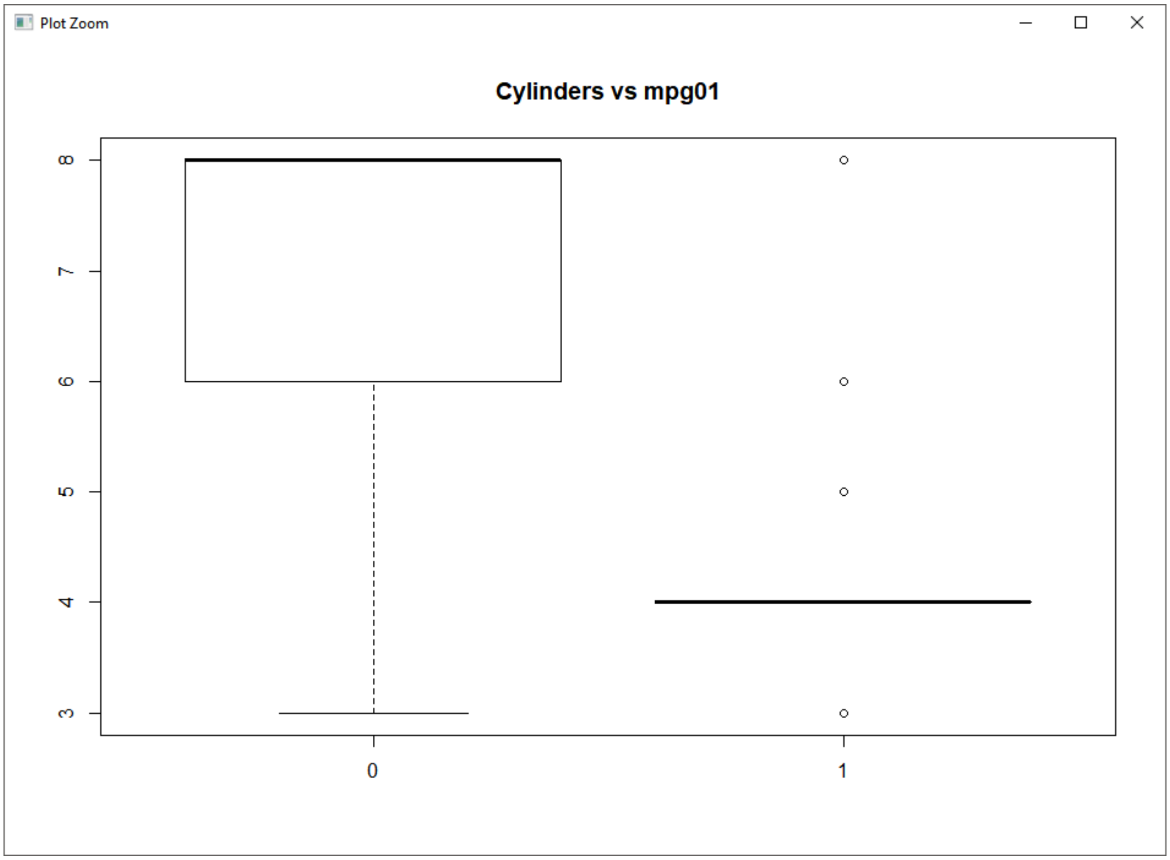
>cor(Auto[, -9])

```
> cor(Auto[, -9])
                    mpg  cylinders displacement horsepower     weight acceleration       year       o
rigin
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285  0.5805410  0.56
52088
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834 -0.3456474 -0.56
89316
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -0.3698552 -0.61
45351
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -0.4163615 -0.45
51715
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -0.3091199 -0.58
50054
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000  0.2903161  0.21
27458
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161  1.0000000  0.18
15277
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458  0.1815277  1.00
00000
mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215  0.4299042  0.51
36984
mpg01.1       0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215  0.4299042  0.51
36984
                  mpg01    mpg01.1
mpg           0.8369392  0.8369392
cylinders    -0.7591939 -0.7591939
displacement -0.7534766 -0.7534766
horsepower   -0.6670526 -0.6670526
weight       -0.7577566 -0.7577566
acceleration  0.3468215  0.3468215
year          0.4299042  0.4299042
origin        0.5136984  0.5136984
mpg01         1.0000000  1.0000000
mpg01.1       1.0000000  1.0000000
> 
```

>pairs(Auto)

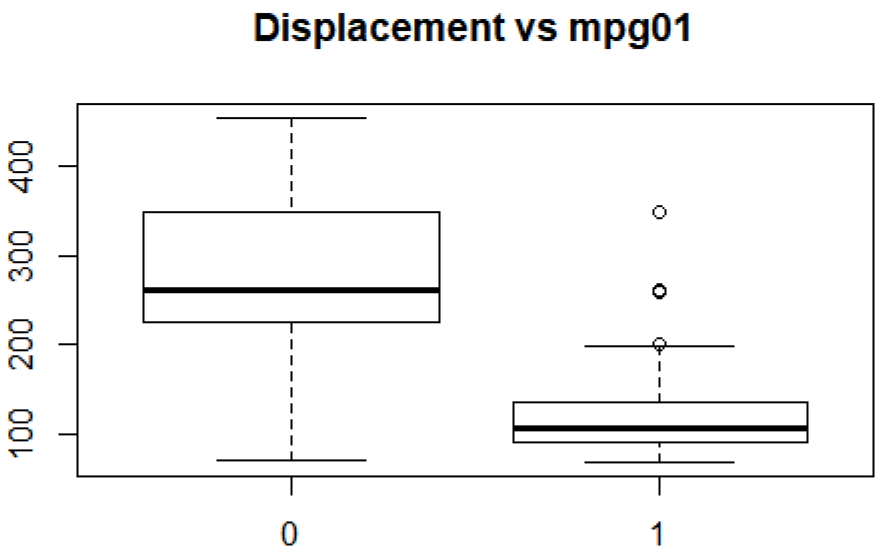


>boxplot(cylinders ~ mpg01, data = Auto, main = "Cylinders vs mpg01")

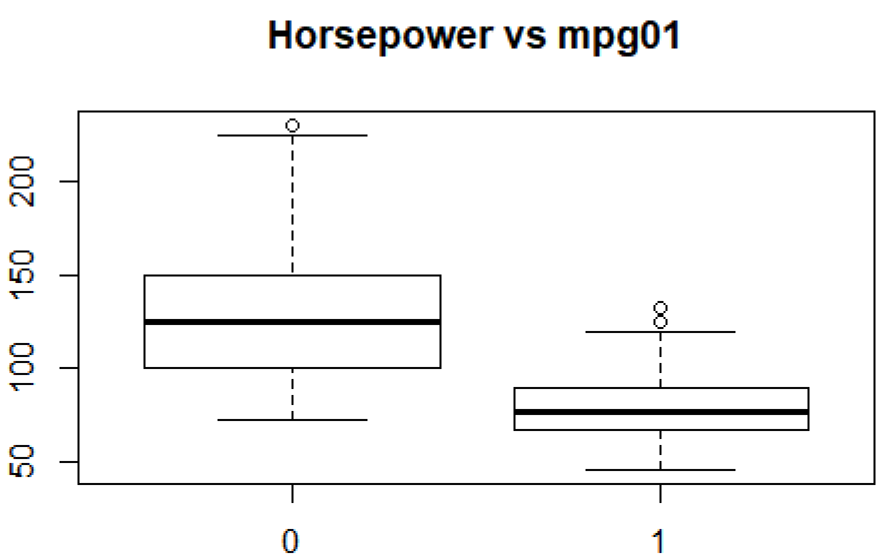


>boxplot(displacement ~ mpg01, data = Auto, main = "Displacement vs mpg01")

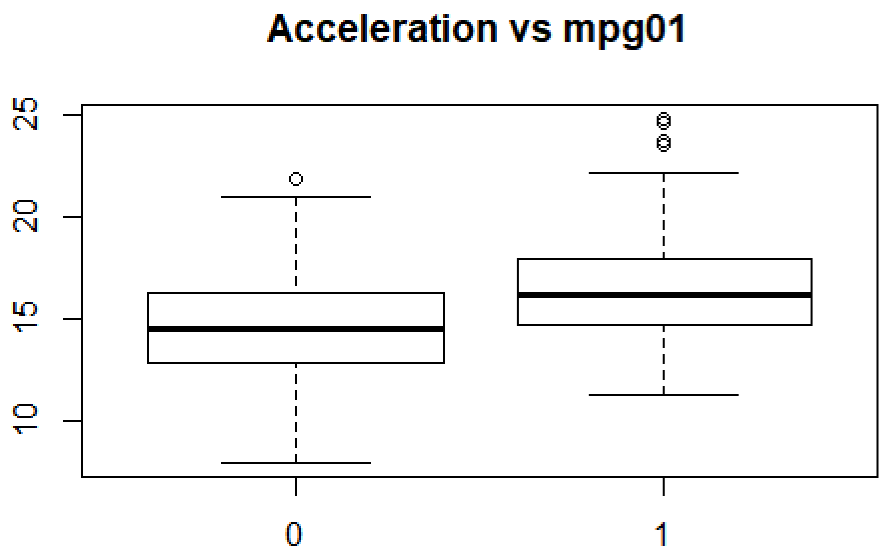
here we get a box plot for the values of 0 and 1 when plotted between Displacement and mpg01




```
>boxplot(horsepower ~ mpg01, data = Auto, main = "Horsepower vs mpg01")
## here we get a box plot for the values of 0 and 1 when plotted between Horsepower and mpg01
```

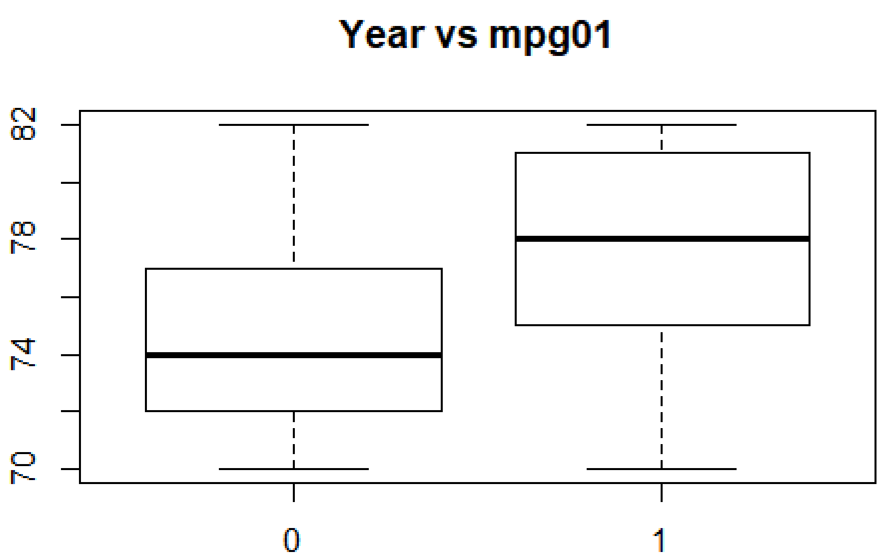



```
>boxplot(acceleration ~ mpg01, data = Auto, main = "Acceleration vs mpg01")
## here we get a box plot for the values of 0 and 1 when plotted between Acceleration and mpg01
```



>boxplot(year ~ mpg01, data = Auto, main = "Year vs mpg01")

here we get a box plot for the values of 0 and 1 when plotted between Year and mpg01



We may conclude that there exists some association between “mpg01” and “cylinders”, “weight”, “displacement” and “horsepower”.

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

Ans.

>train <- (year %% 2 == 0)

>Auto.train <- Auto[train,]

>Auto.test <- Auto[!train,]

>mpg01.test <- mpg01[!train]

(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?

Ans.

>fit.lda <- lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = train)

>fit.lda

```
> library(MASS)
> fit.lda <- lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = trai
n)
> fit.lda
Call:
lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
    subset = train)

Prior probabilities of groups:
        0         1
0.4571429 0.5428571

Group means:
  cylinders   weight displacement horsepower
0  6.812500 3604.823     271.7396  133.14583
1  4.070175 2314.763     111.6623   77.92105

Coefficients of linear discriminants:
                       LD1
cylinders    -0.6741402638
weight       -0.0011465750
displacement  0.0004481325
horsepower    0.0059035377
```

For LDA to work we should import the library named "mass"

>pred.lda <- predict(fit.lda, Auto.test)

>table(pred.lda$class, mpg01.test)

```
> pred.lda <- predict(fit.lda, Auto.test)
> table(pred.lda$class, mpg01.test)
   mpg01.test
     0  1
  0 86  9
  1 14 73
```

>mean(pred.qda$class != mpg01.test)

```
> mean(pred.lda$class != mpg01.test)
[1] 0.1263736
```

We may conclude that we have a test error rate of 12.63736%.

(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?

Ans.

>fit.qda <- qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = train)

>fit.qda

```
> fit.qda <- qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = trai
n)
> fit.qda
Call:
qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
    subset = train)

Prior probabilities of groups:
        0         1
0.4571429 0.5428571

Group means:
  cylinders   weight displacement horsepower
0  6.812500 3604.823     271.7396  133.14583
1  4.070175 2314.763     111.6623   77.92105
```

>pred.qda <- predict(fit.qda, Auto.test)

>table(pred.qda$class, mpg01.test)

```
> pred.qda <- predict(fit.qda, Auto.test)
> table(pred.qda$class, mpg01.test)
   mpg01.test
     0  1
  0 89 13
  1 11 69
```

>mean(pred.qda$class != mpg01.test)

```
> mean(pred.qda$class != mpg01.test)
[1] 0.1318681
```

We may conclude that we have a test error rate of 13.1868132%.

(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?

Ans.

>fit.glm <- glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, family = binomial, subset = train)

>summary(fit.glm)

```
> fit.glm <- glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, family = bino
mial, subset = train)
> summary(fit.glm)

Call:
glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
    family = binomial, data = Auto, subset = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.48027  -0.03413   0.10583   0.29634   2.57584

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  17.658730   3.409012   5.180 2.22e-07 ***
cylinders    -1.028032   0.653607  -1.573   0.1158
weight       -0.002922   0.001137  -2.569   0.0102 *
displacement  0.002462   0.015030   0.164   0.8699
horsepower   -0.050611   0.025209  -2.008   0.0447 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 289.58  on 209  degrees of freedom
Residual deviance:  83.24  on 205  degrees of freedom
```

>probs <- predict(fit.glm, Auto.test, type = "response")

>pred.glm <- rep(0, length(probs))

>pred.glm[probs > 0.5] <- 1

>table(pred.glm, mpg01.test)

```
> probs <- predict(fit.glm, Auto.test, type = "response")
> pred.glm <- rep(0, length(probs))
> pred.glm[probs > 0.5] <- 1
> table(pred.glm, mpg01.test)
         mpg01.test
pred.glm  0  1
       0 89 11
       1 11 71
```

>mean(pred.glm != mpg01.test)

```
> mean(pred.glm != mpg01.test)
[1] 0.1208791
```

We may conclude that we have a test error rate of 12.08791%.

(g) Perform KNN on the training data, with several values of K, in order to predict “mpg01” using 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?

Ans.

>train.X <- cbind(cylinders, weight, displacement, horsepower)[train,]

>test.X <- cbind(cylinders, weight, displacement, horsepower)[!train,]

>train.mpg01 <- mpg01[train]

>set.seed(1)

##k=1

>pred.knn <- knn(train.X, test.X, train.mpg01, k = 1)

>table(pred.knn, mpg01.test)

```
> train.X <- cbind(cylinders, weight, displacement, horsepower)[train, ]
> test.X <- cbind(cylinders, weight, displacement, horsepower)[!train, ]
> train.mpg01 <- mpg01[train]
> set.seed(1)
> pred.knn <- knn(train.X, test.X, train.mpg01, k = 1)
> table(pred.knn, mpg01.test)
        mpg01.test
pred.knn  0  1
       0 83 11
       1 17 71
```

>mean(pred.knn != mpg01.test)

```
> mean(pred.knn != mpg01.test)
[1] 0.1538462
```

We may conclude that we have a test error rate of 15.3846154% for K=1.

K=10

>pred.knn <- knn(train.X, test.X, train.mpg01, k = 10)

>table(pred.knn, mpg01.test)

```
> pred.knn <- knn(train.X, test.X, train.mpg01, k = 10)
> table(pred.knn, mpg01.test)
        mpg01.test
pred.knn  0  1
       0 77  7
       1 23 75
```

>mean(pred.knn != mpg01.test)

```
> mean(pred.knn != mpg01.test)
[1] 0.1648352
```

We may conclude that we have a test error rate of 16.4835165% for K=10.

k=100

>pred.knn <- knn(train.X, test.X, train.mpg01, k = 100)

>table(pred.knn, mpg01.test)

```
> pred.knn <- knn(train.X, test.X, train.mpg01, k = 100)
> table(pred.knn, mpg01.test)
        mpg01.test
pred.knn  0  1
       0 81  7
       1 19 75
```

>mean(pred.knn != mpg01.test)

```
> mean(pred.knn != mpg01.test)
[1] 0.1428571
```

We may conclude that we have a test error rate of 14.2857143% for K=100. So, a K value of 100 seems to perform the best.

GitHub Link : https://github.com/Yuvesh95/R-programming