Assignment 3

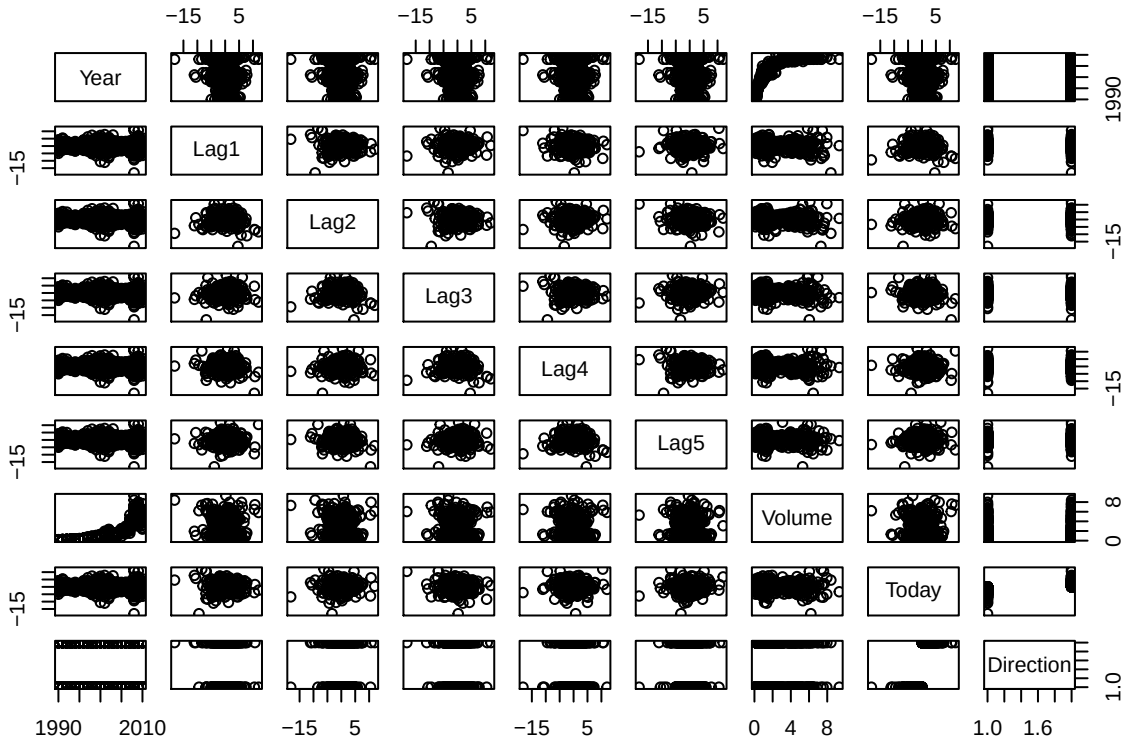
Mayank Jaggi

February 27, 2018

Problem 1

a)

```
library(ISLR)
attach(Weekly)
pairs(Weekly)
```




```
cor(Weekly[1:8])
```

```
##                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
```

Based on the graphical and numerical summaries, there is a correlation between "Volume" and "Year" and between "Today" and "Direction"

b)

```
logist.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,family=binomial)
summary(logist.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial)
##
## 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
```

From the summary, based on the p-value for each predictor, Lag2 is statistically significant. The other predictors have a higher p-value (>0.05), so they fail to reject null hypothesis and are statistically not significant.

c)

```
logist.probs=predict(logist.fit,type = "response") #probability for each
dim(Weekly)
```

```
## [1] 1089    9
```

```
logist.pred=rep("Down",1089)
logist.pred[logist.probs>0.5]="Up"
table(logist.pred,Direction)
```

```
##            Direction
## logist.pred Down  Up
##        Down   54  48
##        Up    430 557
```

```
A= mean(logist.pred==Direction)                    #accuracy
E= 1-A                                             #error rate
print(A)
```

```
## [1] 0.5610652
```

```
print(E)
```

```
## [1] 0.4389348
```

Accuracy= 0.561 Error Rate= 0.439 Sensitivity=557/(48+557)=0.921 Specificity=54/(54+430)=0.112

Error rate is high and since this is training error rate, test error rate would be much larger and so the model is a poor fit to the data.

d) Logistic Regression

```
train=(Year<2009)
Weekly.test=Weekly[!train,]
Direction.2009=Direction[!train]
logist.fit1=glm(Direction~Lag2,family = binomial, subset=train)
logist.probs1=predict(logist.fit1,Weekly.test,type="response")     #probability for each observation
logist.pred1=rep("Down",104)
logist.pred1[logist.probs1>0.5]="Up"
table(logist.pred1,Direction.2009)
```

```
##             Direction.2009
## logist.pred1 Down Up
##         Down    9  5
##         Up     34 56
```

```
A1= mean(logist.pred1==Direction.2009)                  #accuracy
E1= 1-A1                                                #error rate
print(A1)
```

```
## [1] 0.625
```

```
print(E1)
```

```
## [1] 0.375
```

Accuracy= 0.625 Error Rate= 0.375 Sensitivity=56/(56+5)=0.918 Specificity=9/(34+9)=0.209

e) LDA

```
library(MASS)
lda.fit=lda(Direction~Lag2,data=Weekly,subset=Year<2009)
lda.pred=predict(lda.fit,Weekly.test)
table(lda.pred$class,Direction.2009)
```

```
##       Direction.2009
##        Down Up
##   Down    9  5
##   Up     34 56
```

```
A2=mean(lda.pred$class==Direction.2009)                #accuracy
E2=1-A2                                                #error rate
print(A2)
```

```
## [1] 0.625
```

```
print(E2)
```

```
## [1] 0.375
```

Accuracy= 0.625 Error Rate= 0.375 Sensitivity=56/(56+5)=0.918 Specificity=9/(34+9)=0.209

The results of logistic regression and LDA are almost identical.

f) QDA

```
qda.fit=qda(Direction~Lag2,data=Weekly,subset=Year<2009)
qda.pred=predict(qda.fit,Weekly.test)
table(qda.pred$class,Direction.2009)
```

```
##       Direction.2009
##        Down Up
##   Down    0  0
##   Up     43 61
```

```
A3=mean(qda.pred$class==Direction.2009)                #accuracy
E3=1-A3                                                #error rate
print(A3)
```

```
## [1] 0.5865385
```

```
print(E3)
```

```
## [1] 0.4134615
```

Accuracy= 0.587 Error Rate= 0.413 Sensitivity=61/(61+0)=1 Specificity=0/(0+43)=0

The QDA model fails to idenify all the weeks when the market went down, whereas correctly identifies all the weeks when the market went up.

g) KNN

```
library(class)
train.X=Lag2[train]
```

```
test.X=Lag2[!train]
train.Direction=Direction[train]
set.seed(1)
knn.pred=knn(data.frame(train.X),data.frame(test.X),train.Direction,k=1)
table(knn.pred,Direction.2009)
```

```
##         Direction.2009
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

```
mean(knn.pred==Direction.2009)
```

```
## [1] 0.5
```

Accuracy= (21+31)/(21+22+30+31)=0.5 Error Rate= 1-Accuracy=0.5 Sensitivity=31/(30+31)=0.508 Specificity=21/(21+22)=0.488

h)

Comparing the test error rates, we see that logistic regression and LDA have the minimum error rates followed by QDA and KNN. So logistic regression and LDA appears to provide the best results.

i)

I Lag1:Lag2

1) Logistic regression

```
#Since Lag1 and Lag2 are the most significant coeffecients we perform all analysis using these two pred
logist.fit4=glm(Direction~Lag1:Lag2,family=binomial,subset = train)
logist.probs1=predict(logist.fit4,Weekly.test,type="response")     #probability for each observation
logist.pred1=rep("Down",104)
logist.pred1[logist.probs1>0.5]="Up"
table(logist.pred1,Direction.2009)
```

```
##             Direction.2009
## logist.pred1 Down Up
##         Down    1  1
##         Up     42 60
```

```
A7= mean(logist.pred1==Direction.2009)                    #accuracy
print(A7)
```

```
## [1] 0.5865385
```

2) LDA

```
library(MASS)
lda.fit2=lda(Direction~Lag1:Lag2,data=Weekly,subset=train)
lda.pred2=predict(lda.fit2,Weekly.test)
table(lda.pred2$class,Direction.2009)
```

```
##       Direction.2009
##        Down Up
```

```
##   Down    0  1
##   Up     43 60
```

```
A8=mean(lda.pred$class==Direction.2009)                    #accuracy
print(A8)
```

```
## [1] 0.625
```

3) QDA

```
qda.fit2=qda(Direction~Lag1:Lag2,data=Weekly,subset=train)
qda.pred2=predict(qda.fit2,Weekly.test)
table(qda.pred2$class,Direction.2009)
```

```
##       Direction.2009
##        Down Up
##   Down   16 32
##   Up     27 29
```

```
A9=mean(qda.pred2$class==Direction.2009)                    #accuracy
print(A9)
```

```
## [1] 0.4326923
```

4)KNN

```
#Varying k from 1 to 104
n=length(Direction.2009)
mean.knn.pred2=c()
for (i in 1:n){
  knn.pred2=knn(data.frame(train.X),data.frame(test.X),train.Direction,k=i)
  mean.knn.pred2[i]=mean(knn.pred2==Direction.2009)
}
mean.knn.pred2
```

```
##   [1] 0.5000000 0.4615385 0.5480769 0.5384615 0.5480769 0.5769231 0.5480769
##   [8] 0.5769231 0.5576923 0.5865385 0.5576923 0.6057692 0.5865385 0.5673077
##  [15] 0.5865385 0.5384615 0.5865385 0.5673077 0.5576923 0.6057692 0.5576923
##  [22] 0.5865385 0.5673077 0.5576923 0.5288462 0.5480769 0.5192308 0.5288462
##  [29] 0.5576923 0.5480769 0.5384615 0.5480769 0.5480769 0.5769231 0.5769231
##  [36] 0.5288462 0.5673077 0.5576923 0.5576923 0.5865385 0.5576923 0.5384615
##  [43] 0.5480769 0.5769231 0.5480769 0.5673077 0.6153846 0.5961538 0.5865385
##  [50] 0.5673077 0.5673077 0.5673077 0.5576923 0.5480769 0.5576923 0.5576923
##  [57] 0.5576923 0.5865385 0.5961538 0.5865385 0.6057692 0.5769231 0.5769231
##  [64] 0.5769231 0.5865385 0.5769231 0.5480769 0.6057692 0.5673077 0.5865385
##  [71] 0.5576923 0.6250000 0.5865385 0.5576923 0.5673077 0.5673077 0.5961538
##  [78] 0.5769231 0.6057692 0.6153846 0.5673077 0.5961538 0.5865385 0.5673077
##  [85] 0.5961538 0.5769231 0.5961538 0.5961538 0.5865385 0.5865385 0.5865385
##  [92] 0.5769231 0.5769231 0.5769231 0.5961538 0.6057692 0.5865385 0.5673077
##  [99] 0.5865385 0.5480769 0.5673077 0.5673077 0.5673077 0.5961538
```

```
max(mean.knn.pred2)
```

```
## [1] 0.625
```

```
which(mean.knn.pred2==max(mean.knn.pred2))
```

```
## [1] 72
```

k=47 gives the highest accuracy

```
#Confusion matrix for 47
knn.pred2=knn(data.frame(train.X),data.frame(test.X),train.Direction,k=47)
table(knn.pred2,Direction.2009)
```

```
##          Direction.2009
## knn.pred2 Down Up
##      Down   23 20
##      Up     20 41
```

II) Lag2+Lag1^2

1) Logistic regression

```
logist.fit5=glm(Direction~I(Lag1^2)+Lag2,family=binomial,subset = train)
logist.probs2=predict(logist.fit5,Weekly.test,type="response")     #probability for each observation
logist.pred2=rep("Down",104)
logist.pred2[logist.probs2>0.5]="Up"
table(logist.pred2,Direction.2009)
```

```
##             Direction.2009
## logist.pred2 Down Up
##         Down    8  2
##         Up     35 59
```

```
A9= mean(logist.pred2==Direction.2009)                    #accuracy
print(A9)
```

```
## [1] 0.6442308
```

2) LDA

```
lda.fit3=lda(Direction~I(Lag1^2)+Lag2,data=Weekly,subset=train)
lda.pred3=predict(lda.fit3,Weekly.test)
table(lda.pred3$class,Direction.2009)
```

```
##       Direction.2009
##        Down Up
##   Down    8  2
##   Up     35 59
```

```
A10=mean(lda.pred3$class==Direction.2009)                 #accuracy
print(A10)
```

```
## [1] 0.6442308
```

3) QDA

```
qda.fit3=qda(Direction~I(Lag1^2)+Lag2,data=Weekly,subset=train)
qda.pred3=predict(qda.fit3,Weekly.test)
table(qda.pred3$class,Direction.2009)
```

```
##       Direction.2009
##        Down Up
```

```
##   Down   32 40
##   Up     11 21
```

```
A11=mean(qda.pred3$class==Direction.2009)                    #accuracy
print(A11)
```

```
## [1] 0.5096154
```

Amongst all the methods the best result is shown by Logistic regression/LDA with predictors (Lag1)^2 and Lag2.

Problem 2

```
library(ROCR)
```

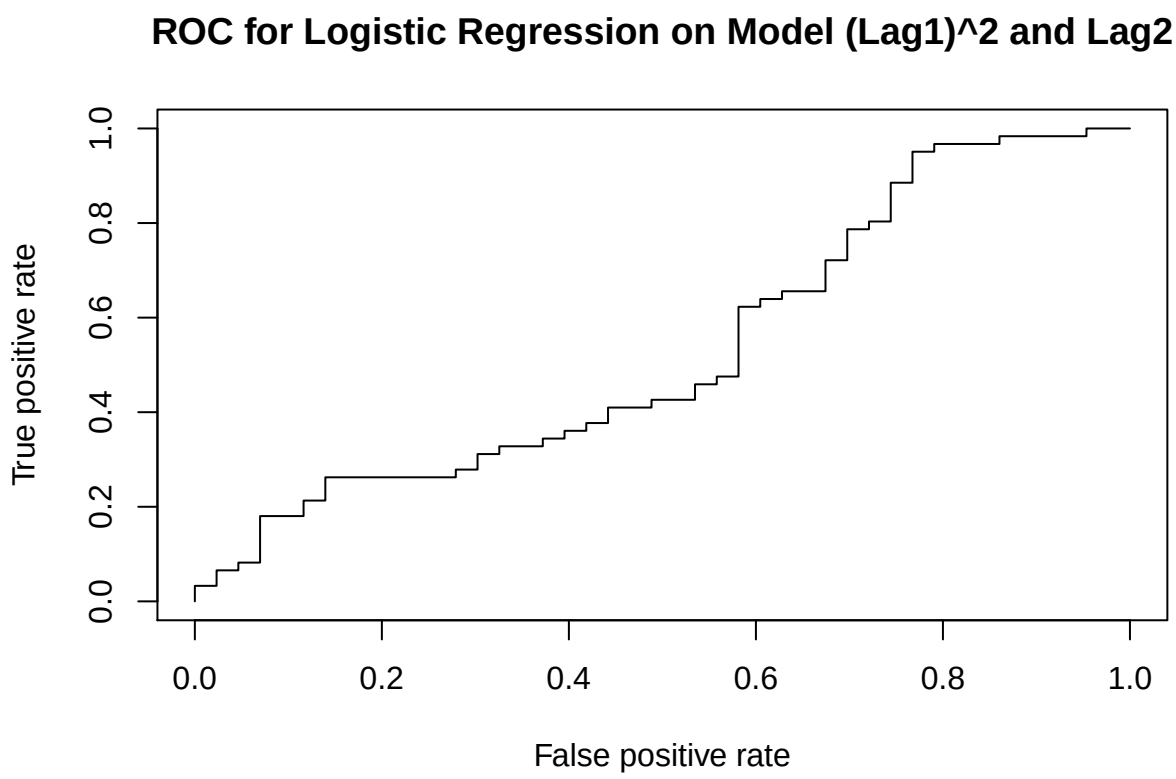
```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

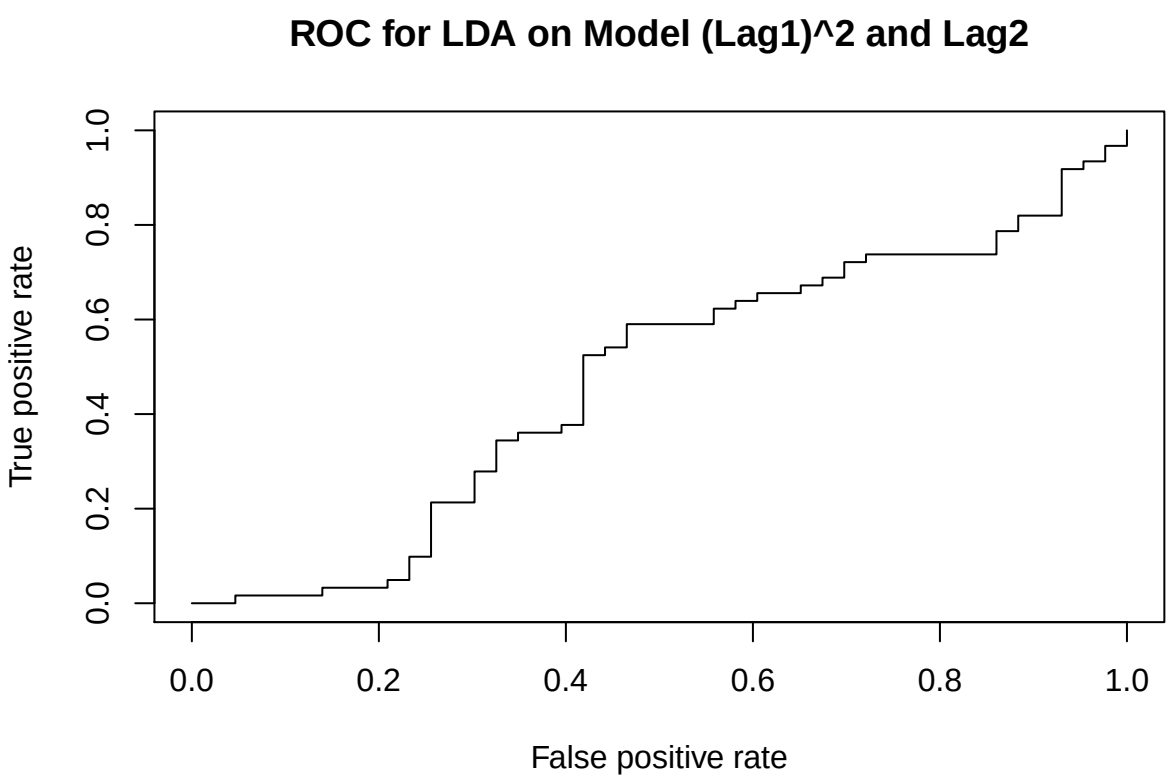
```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
Append.Direction.2009=rep(0,length(Direction.2009))
Append.Direction.2009[Direction.2009=='Up']=1

#for logistic regression
roc.logist.pred=prediction(logist.probs2,Append.Direction.2009)
roc.logist.perf=performance(roc.logist.pred,'tpr','fpr')
plot(roc.logist.perf,main="ROC for Logistic Regression on Model (Lag1)^2 and Lag2")
```




```
#for LDA
roc.lda.pred=prediction(lda.pred3$posterior[,1],Append.Direction.2009)
roc.lda.perf=performance(roc.lda.pred,'tpr','fpr')
plot(roc.lda.perf,main="ROC for LDA on Model (Lag1)^2 and Lag2")
```


ROC for LDA on Model (Lag1)^2 and Lag2
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

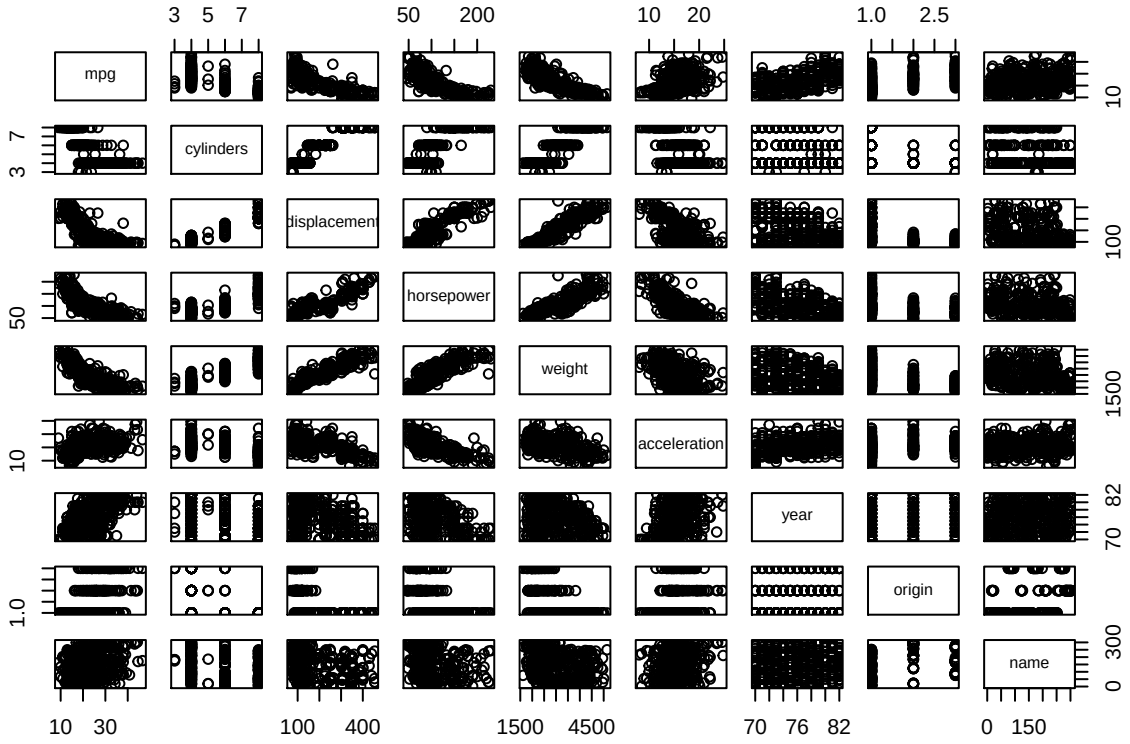
Problem 3

a)

```
attach(Auto)
mpg01=rep(0, length(mpg))
mpg01[mpg > median(mpg)]= 1
Auto1= data.frame(Auto, mpg01)
```

b)

```
pairs(Auto)                    #graphical representation
```

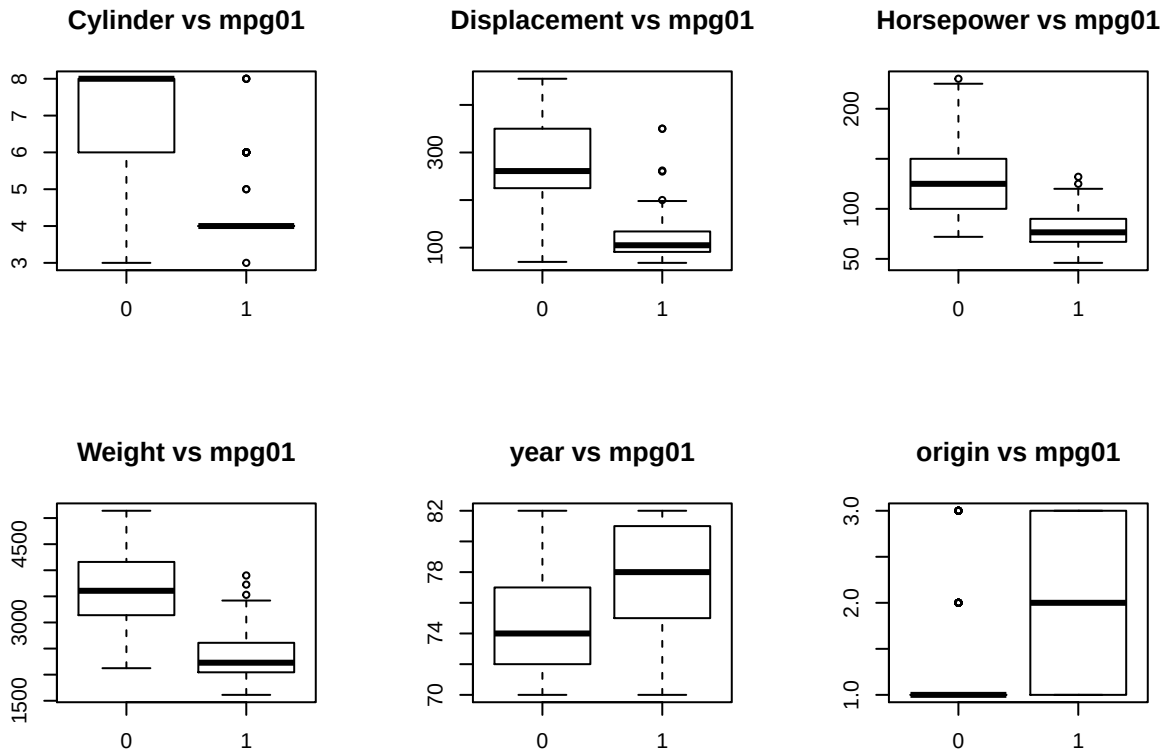



```
cor(Auto1[,-9])
```

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

```
## mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566
##              acceleration       year     origin      mpg01
## mpg             0.4233285  0.5805410  0.5652088  0.8369392
## cylinders      -0.5046834 -0.3456474 -0.5689316 -0.7591939
## displacement   -0.5438005 -0.3698552 -0.6145351 -0.7534766
## horsepower     -0.6891955 -0.4163615 -0.4551715 -0.6670526
## weight         -0.4168392 -0.3091199 -0.5850054 -0.7577566
## acceleration    1.0000000  0.2903161  0.2127458  0.3468215
## year            0.2903161  1.0000000  0.1815277  0.4299042
## origin          0.2127458  0.1815277  1.0000000  0.5136984
## mpg01           0.3468215  0.4299042  0.5136984  1.0000000
```

```
par(mfrow=c(2,3))
boxplot(cylinders~mpg01,main="Cylinder vs mpg01")
boxplot(displacement~mpg01,main="Displacement vs mpg01")
boxplot(horsepower~mpg01,main="Horsepower vs mpg01")
boxplot(weight~mpg01,main="Weight vs mpg01")
boxplot(year~mpg01,main="year vs mpg01")
boxplot(origin~mpg01,main="origin vs mpg01")
```



Based on the above boxplots mpg01 has cylinder, displacement, horsepower, weight as significant predictors.This can be double checked with the aforementioned correlation matrix.

c)

```
smp_size=floor(0.75*nrow(Auto1))
set.seed(1)
train1=sample(seq_len(nrow(Auto1)),size=smp_size)
Auto1.train=Auto1[train1,]                          #train data
Auto1.test=Auto1[-train1,]                          #test data
mpg01.train=mpg01[train1]
mpg01.test=mpg01[-train1]
```

d)LDA

```
lda.fit1=lda(mpg01~cylinders+displacement+horsepower+weight,data=Auto1.train)
lda.pred1=predict(lda.fit1,Auto1.test)
lda.class=lda.pred1$class
table(lda.class,mpg01.test)
```

```
##          mpg01.test
## lda.class  0  1
##         0 41  1
##         1  5 51
```

```
A4=mean(lda.class==mpg01.test)                          #accuracy
E4=1-A4                                                 #error rate
print(E4)
```

```
## [1] 0.06122449
```

Test error rate=0.061

e)QDA

```
qda.fit1=qda(mpg01~cylinders+displacement+horsepower+weight,data=Auto1.train)
qda.pred1=predict(qda.fit1,Auto1.test)
table(qda.pred1$class,mpg01.test)
```

```
##    mpg01.test
##      0  1
##   0 42  4
##   1  4 48
```

```
A5=mean(qda.pred1$class==mpg01.test)                  #accuracy
E5=1-A5                                                 #error rate
print(E5)
```

```
## [1] 0.08163265
```

Test error rate=0.082

f)Logistic regression

```
logist.fit3=glm(mpg01~cylinders+displacement+horsepower+weight,family = binomial, data=Auto1.train)
logist.probs2=predict(logist.fit3,Auto1.test,type="response")     #probability for each observation

logist.pred1=rep("0",length(logist.probs2))
logist.pred1[logist.probs2>0.5]="1"
table(logist.pred1,mpg01.test)
```

```
##             mpg01.test
## logist.pred1  0  1
##            0 39  3
##            1  7 49
```

```
A6= mean(logist.pred1==mpg01.test)                    #accuracy
E6= 1-A6                                                  #error rate
print(E6)
```

```
## [1] 0.1020408
```

Test error rate=0.102

g)KNN

1)k=1

```
train.X1=cbind(Auto1.train$cylinders,Auto1.train$displacement,Auto1.train$horsepower,Auto1.train$weight)
test.X1=cbind(Auto1.test$cylinders,Auto1.test$displacement,Auto1.test$horsepower,Auto1.test$weight)
set.seed(1)
knn.pred1=knn(train.X1,test.X1,mpg01.train,k=1)
table(knn.pred1,mpg01.test)
```

```
##          mpg01.test
## knn.pred1  0  1
##         0 38  8
##         1  8 44
```

Test Error rate= (8+8)/(8+8+38+44)=0.163

2)k=10

```
train.X1=cbind(Auto1.train$cylinders,Auto1.train$displacement,Auto1.train$horsepower,Auto1.train$weight)
test.X1=cbind(Auto1.test$cylinders,Auto1.test$displacement,Auto1.test$horsepower,Auto1.test$weight)
set.seed(1)
knn.pred1=knn(train.X1,test.X1,mpg01.train,k=10)
table(knn.pred1,mpg01.test)
```

```
##          mpg01.test
## knn.pred1  0  1
##         0 35  2
##         1 11 50
```

Test Error rate=(2+11)/(2+11+35+50)=0.133

3)k=100

```
train.X1=cbind(Auto1.train$cylinders,Auto1.train$displacement,Auto1.train$horsepower,Auto1.train$weight)
test.X1=cbind(Auto1.test$cylinders,Auto1.test$displacement,Auto1.test$horsepower,Auto1.test$weight)
set.seed(1)
knn.pred1=knn(train.X1,test.X1,mpg01.train,k=100)
table(knn.pred1,mpg01.test)
```

```
##          mpg01.test
## knn.pred1  0  1
##         0 41  3
##         1  5 49
```

Test Error rate=(3+5)/(3+5+41+49)=0.082

k=100 performs best for the data