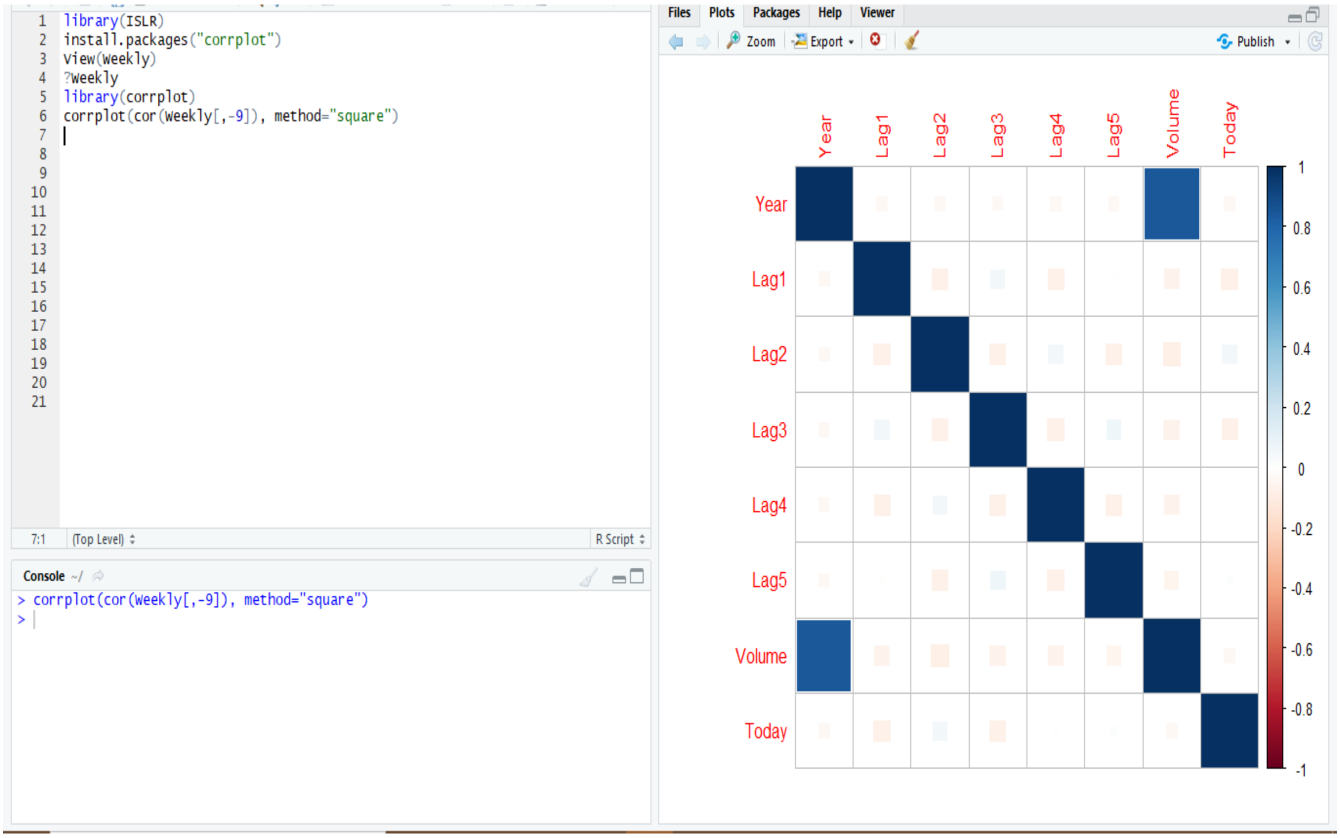
Alexa Summers, Santhoshini Sree Bolisetty, Gireesh Kumar Muppalla
CS 5565 Dr. Song

10.

(a)

The plot does not show any linear relationship between predictors



(b)

The only variable which has low p value(<0.05) is lag2. Hence, it is the only predictor to be considered as statistically significant

```
attach(Weekly)
a<-glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data = Weekly,family = binomial)
summary(a)

10:1    (Top Level) ÷
```

Console ~/

```
    Direction, Lag1, Lag2, Lag3, Lag4, Lag5, Today, Volume, Year

> a<-glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data = Weekly,family = binomial)
> summary(a)

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

> 
```

(c)

Total weekly trend:

(54+557)/(54+48+430+557)=0.5611

Up weekly trends:

557/(430+557)=0.9207

Down weekly trends:

54/(430+54)=0.1115

From the above information, we can conclude that the model predicted the up weekly trend 92.07% correctly.

```r
attach(Weekly)
fit<-glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Weekly,family=binomial)
summary(fit)
prob= predict(Weekly.fit, type='response')
pred =rep("Down", length(prob))
pred[prob > 0.5] = "Up"
table(pred, Direction)
```

32:23 (Top Level)

Console ~/

```
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

> prob= predict(Weekly.fit, type='response')
> pred =rep("Down", length(prob))
> pred[prob > 0.5] = "Up"
> table(pred, Direction)
      Direction
pred   Down  Up
  Down   54  48
  Up    430 557
>
```

(d)

From below, we can say that the model gave 62.5% accuracy rate. While the downward and upward trends gives 91.80% and 20.83% accuracy.

This means that the model is predicting downward trends way more correct than the upward trends

```
train = (Year<2009)
rows <-Weekly[!train,]
model<-glm(Direction~Lag2, data=Weekly,family=binomial, subset=train)
prob= predict(model, rows, type = "response")
pred = rep("Down", length(prob))
pred[prob > 0.5] = "Up"
Direct = Direction[!train]
table(pred, Direct)
mean(pred == Direct)
```

43:1 (Top Level)

Console ~/

```
> train = (Year<2009)
> rows <-Weekly[!train,]
> model<-glm(Direction~Lag2, data=Weekly,family=binomial, subset=train)
> prob= predict(model, rows, type = "response")
> pred = rep("Down", length(prob))
> pred[prob > 0.5] = "Up"
> Direct = Direction[!train]
> table(pred, Direct)
      Direct
pred   Down Up
  Down    9  5
  Up     34 56
> mean(pred == Direct)
[1] 0.625
>
```

(e)

The logistic and lda are giving the same accuracy rates.

```
library(MASS)
fit<-lda(Direction~Lag2, data=Weekly,family=binomial, subset=train)
pred<-predict(fit, rows)
table(pred$class, Direct)
mean(pred$class==Direct)
```

```
36:1    (Top Level) ÷
```

```
Console ~/
> model<-glm(Direction~Lag2, data=Weekly,family=binomial, subset=train)
> prob= predict(model, rows, type = "response")
> pred = rep("Down", length(prob))
> pred[prob > 0.5] = "Up"
> Direct = Direction[!train]
> table(pred, Direct)
      Direct
pred   Down Up
  Down    9  5
  Up     34 56
> fit<-lda(Direction~Lag2, data=Weekly,family=binomial, subset=train)
Error in lda(Direction ~ Lag2, data = Weekly, family = binomial, subset = train) :
  could not find function "lda"
> library(MASS)
> fit<-lda(Direction~Lag2, data=Weekly,family=binomial, subset=train)
> pred<-predict(fit, rows)
> table(pred$class, Direct)
      Direct
       Down Up
  Down    9  5
  Up     34 56
> mean(pred$class==Direct)
[1] 0.625
>
```

(f)

The qda is giving the lower accuracy compared to logistic and lda models.

```
fit = qda(Direction ~ Lag2, data = Weekly, subset = train)
rows <-Weekly[!train,]
pred = predict(fit, rows)$class
Direct = Direction[!train]
table(pred, Direct)
mean(pred == Direct)
```

44:1 (Top Level)

Console ~/

```
> fit = qda(Direction ~ Lag2, data = Weekly, subset = train)
> rows <-Weekly[!train,]
> pred = predict(fit, rows)$class
> Direct = Direction[!train]
> table(pred, Direct)
      Direct
pred   Down Up
  Down    0  0
  Up     43 61
> mean(pred == Direct)
[1] 0.5865385
> 
```

(g)

The knn model is giving a 50% accuracy.

```
library(class)
train = (Year<2009)
train1=as.matrix(Lag2[train])
Direct = Direction[!train]
test=as.matrix(Lag2[!train])
Direct1 =Direction[train]
set.seed(1)
pred=knn(train1,test,Direct1,k=1)
table(pred,Direct)
mean(pred == Direct)
```

76:1 (Top Level)

Console ~/

```
> library(class)
> train = (Year<2009)
> train1=as.matrix(Lag2[train])
> Direct = Direction[!train]
> test=as.matrix(Lag2[!train])
> Direct1 =Direction[train]
> set.seed(1)
> pred=knn(train1,test,Direct1,k=1)
> table(pred,Direct)
      Direct
pred   Down Up
  Down   21 30
  Up     22 31
> mean(pred == Direct)
[1] 0.5
>
```

(h)

From this we say that the logistic and lda models are giving the better accuracy rates(62.5%)

(i)

The below shows the logistic model, which is giving a 54.06% accuracy

```
fit<-glm(Direction~Lag2:Lag4+Lag2, data=Weekly,family=binomial, subset=train)
rows <-Weekly[!train,]
prob= predict(fit, rows, type = "response")
pred = rep("Down", length(logWeekly.prob))
pred[prob > 0.5] = "Up"
Direct = Direction[!train]
table(pred, Direct)
mean(pred == Direct)

```

```
117:1    (Top Level) ÷
```

```
Console ~/
> fit<-glm(Direction~Lag2:Lag4+Lag2, data=Weekly,family=binomial, subset=train)
> rows <-Weekly[!train,]
> prob= predict(fit, rows, type = "response")
> pred = rep("Down", length(logWeekly.prob))
> pred[prob > 0.5] = "Up"
> Direct = Direction[!train]
> table(pred, Direct)
      Direct
pred   Down  Up
  Down    5  18
  Up    219 274
> mean(pred == Direct)
[1] 0.5406977
>
```

The lda model is giving 55.2% accuracy

```
fit<-lda(Direction~Lag2:Lag4+Lag2, data=Weekly,family=binomial, subset=train)
pred<-predict(fit, rows)
table(pred$class, Direct)
mean(pred$class==Direct)

```

```
122:1    (Top Level) ÷
```

```
Console ~/
> fit<-lda(Direction~Lag2:Lag4+Lag2, data=Weekly,family=binomial, subset=train)
> pred<-predict(fit, rows)
> table(pred$class, Direct)
      Direct
       Down  Up
  Down    5  12
  Up    219 280
> mean(pred$class==Direct)
[1] 0.5523256
>
```

When k=1 and k=2, the knn model is giving accuracy rates 49.4% and 52.1%

```
Week.train=as.matrix(Lag2[train])
Week.test=as.matrix(Lag2[!train])
train.Direction =Direction[train]
set.seed(1)
Direct = Direction[!train]
Weekknn.pred=knn(Week.train,Week.test,train.Direction,k=1)
table(Weekknn.pred,Direct)
mean(Weekknn.pred == Direct)
Weekknn.pred1=knn(Week.train,Week.test,train.Direction,k=2)
mean(Weekknn.pred1 == Direct)
```

132:30 (Top Level)

Console ~/

```
> Week.train=as.matrix(Lag2[train])
> Week.test=as.matrix(Lag2[!train])
> train.Direction =Direction[train]
> set.seed(1)
> Direct = Direction[!train]
> Weekknn.pred=knn(Week.train,Week.test,train.Direction,k=1)
> table(Weekknn.pred,Direct)
            Direct
Weekknn.pred Down  Up
        Down   90 127
        Up    134 165
> mean(Weekknn.pred == Direct)
[1] 0.494186
> Weekknn.pred1=knn(Week.train,Week.test,train.Direction,k=2)
> mean(Weekknn.pred1 == Direct)
[1] 0.5213178
>
```

From this we can conclude that the lda and logistic models are giving a better accuracy rates for this data.

11.

(a)

```
library(ISLR)
attach(Auto)
summary(Auto)
mpg01 <- rep(0, length(mpg))
View(mpg01)

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

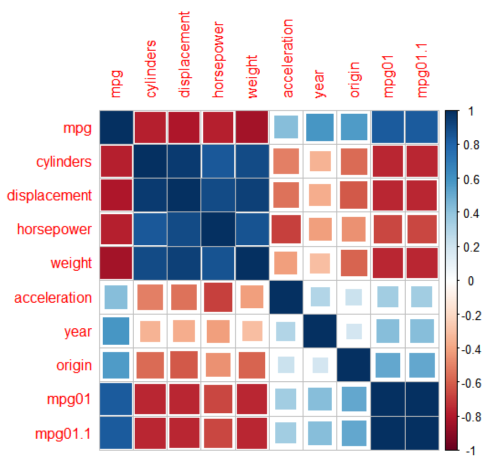
```
64:5   (Top Level)                                                   R Script
Console ~/
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.00
0
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.00
0
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804   Median :15.50   Median :76.00   Median :1.00
0
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.57
7
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.00
0
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.00
0

                 name         mpg01
 amc matador       :  5   Min.   :0.0
 ford pinto        :  5   1st Qu.:0.0
 toyota corolla    :  5   Median :0.5
 amc gremlin       :  4   Mean   :0.5
 amc hornet        :  4   3rd Qu.:1.0
 chevrolet chevette:  4   Max.   :1.0
 (Other)           :365
> mpg01 <- rep(0, length(mpg))
> mpg01 <- rep(0, length(mpg))
> mpg01[mpg > median(mpg)] <- 1
> Auto <- data.frame(Auto, mpg01)
> Auto
```

(b)

Cylinder, displacement and weight are correlating strongly with mpg01. horsepower and origin also correlate with mpg01.

```
corrplot(cor(Auto[,-9]), method="square")
```

```
67:1   (Top Level)                R Script
Console ~/
80  26          4        96.0         69   2189           18.0
  72       2                  renault 12 (sw)      1       1
81  22          4       122.0         86   2395           16.0
  72       1                  ford pinto (sw)      0       0
82  28          4        97.0         92   2288           17.0
  72       3                  datsun 510 (sw)      1       1
83  23          4       120.0         97   2506           14.5
  72       3      toyouta corona mark ii (sw)      1       1
84  28          4        98.0         80   2164           15.0
  72       1                  dodge colt (sw)      1       1
85  27          4        97.0         88   2100           16.5
  72       3         toyota corolla 1600 (sw)      1       1
86  13          8       350.0        175   4100           13.0
  73       1               buick century 350      0       0
87  14          8       304.0        150   3672           11.5
  73       1                     amc matador      0       0
88  13          8       350.0        145   3988           13.0
  73       1                chevrolet malibu      0       0
89  14          8       302.0        137   4042           14.5
  73       1                ford gran torino      0       0
90  15          8       318.0        150   3777           12.5
  73       1             dodge coronet custom      0       0
91  12          8       429.0        198   4952           11.5
  73       1         mercury marquis brougham      0       0
 [ reached 'max' / getOption("max.print") -- omitted 302 rows
 ]
> corrplot(cor(Auto[,-9]), method="square")
>
```



(c)

```r
train <- (year %% 2 == 0)
train.auto <- Auto[train,]
test.auto <- Auto[-train,]
```

```
Console ~/
> train <- (year %% 2 == 0)
> train.auto <- Auto[train,]
> test.auto <- Auto[-train,]
>
```

(d)

lda model is giving an error rate of 8.44%

```r
autolda.fit <- lda(mpg01~displacement+horsepower+weight+year+cylinders+origin, data=train.auto)
autolda.pred <- predict(autolda.fit, test.auto)
table(autolda.pred$class, test.auto$mpg01)
mean(autolda.pred$class != test.auto$mpg01)
```

```
Console ~/
> autolda.fit <- lda(mpg01~displacement+horsepower+weight+year+cylinders+origin, data=train.auto)
> autolda.pred <- predict(autolda.fit, test.auto)
> table(autolda.pred$class, test.auto$mpg01)

       0   1
  0 169   7
  1  26 189
> mean(autolda.pred$class != test.auto$mpg01)
[1] 0.08439898
>
```

(e)

Qda is giving an error rate of 9.97%

```r
autoqda.fit <- qda(mpg01~displacement+horsepower+weight+year+cylinders+origin, data=train.auto)
autoqda.pred <- predict(autoqda.fit, test.auto)
table(autoqda.pred$class, test.auto$mpg01)
mean(autoqda.pred$class != test.auto$mpg01)
```

```
Console ~/
> autoqda.fit <- qda(mpg01~displacement+horsepower+weight+year+cylinders+origin, data=train.auto)
> autoqda.pred <- predict(autoqda.fit, test.auto)
> table(autoqda.pred$class, test.auto$mpg01)

       0   1
  0 176  20
  1  19 176
> mean(autoqda.pred$class != test.auto$mpg01)
[1] 0.09974425
>
```

(f)

The logistic regression method is giving an error rate of 8.44%

```
auto.fit<-glm(mpg01~displacement+horsepower+weight+year+cylinders+origin, data=train.auto,family=binomial)
auto.probs = predict(auto.fit, test.auto, type = "response")
auto.pred = rep(0, length(auto.probs))
auto.pred[auto.probs > 0.5] = 1
table(auto.pred, test.auto$mpg01)
mean(auto.pred != test.auto$mpg01)

89:1  (Top Level)                                      R Script
```

```
Console ~/
> auto.fit<-glm(mpg01~displacement+horsepower+weight+year+cylinders+origin, data=train.auto,family=binomial)
> auto.probs = predict(auto.fit, test.auto, type = "response")
> auto.pred = rep(0, length(auto.probs))
> auto.pred[auto.probs > 0.5] = 1
> table(auto.pred, test.auto$mpg01)

auto.pred   0   1
        0 174  12
        1  21 184
> mean(auto.pred != test.auto$mpg01)
[1] 0.08439898
>
```

(g)

K=1 is giving a lower error rate compared to k=2 and k=3. This can concluded as the error rate keeps increasing with an increasing value of k.

```
train.K= cbind(displacement,horsepower,weight,cylinders,year, origin)[train,]
test.K=cbind(displacement,horsepower,weight,cylinders, year, origin)[-train,]
set.seed(1)
autok.pred=knn(train.K,test.K,train.auto$mpg01,k=1)
mean(autok.pred != test.auto$mpg01)
autok.pred=knn(train.K,test.K,train.auto$mpg01,k=2)
mean(autok.pred != test.auto$mpg01)
autok.pred=knn(train.K,test.K,train.auto$mpg01,k=3)
mean(autok.pred != test.auto$mpg01)

99:36  (Top Level)
```

```
Console ~/
> train.K= cbind(displacement,horsepower,weight,cylinders,year, origin)[train,]
> test.K=cbind(displacement,horsepower,weight,cylinders, year, origin)[-train,]
> set.seed(1)
> autok.pred=knn(train.K,test.K,train.auto$mpg01,k=1)
> mean(autok.pred != test.auto$mpg01)
[1] 0.07161125
> autok.pred=knn(train.K,test.K,train.auto$mpg01,k=2)
> mean(autok.pred != test.auto$mpg01)
[1] 0.09974425
> autok.pred=knn(train.K,test.K,train.auto$mpg01,k=3)
> mean(autok.pred != test.auto$mpg01)
[1] 0.09462916
>
```