COMS 573 Homework 2

Boudhayan Banerjee (715909222)

27 March, 2016

1. a. $set.seed(1)$
   $x1 = runif(100)$
   $x2 = 0.5 * x1 + rnorm(100)/10$
   $y = 2 + 2 * x1 + 0.3 * x2 + rnorm(100)$

   Form of the linear model,

   $$Y = 2 + 2X_1 + 0.3X_2 + \epsilon$$

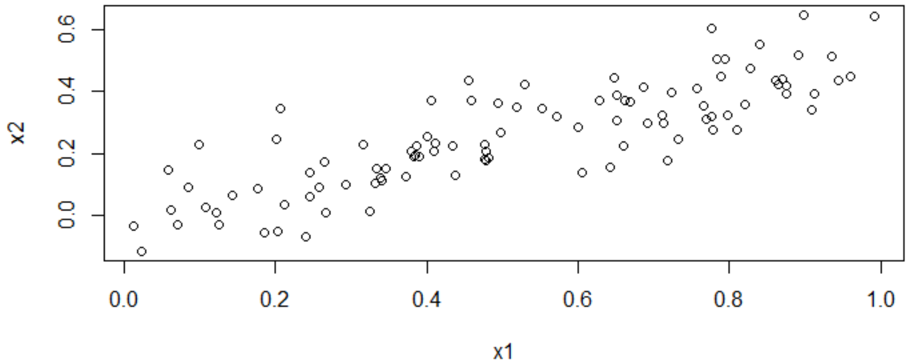
   Regression coefficients,

   $$\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3$$

   b. To find correlation between x1 and x2 we run the following command:
   $cor(x1, x2)$

   The correlation is, 0.8351212.

   Scatter-plot :
   We display the scatter plot we run the command, $plot(x1, x2)$.



c. To fit a least square regression to predict Y using x1 and x2 we need to use following command:

$lm.fit = lm(y\ x1 + x2)$
$summary(lm.fit)$

Results:

```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
     Min       1Q   Median       3Q      Max
-2.75207 -0.59159 -0.03182  0.64792  2.26200

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4688     0.2121  11.641   <2e-16 ***
x1            1.3295     0.6596   2.016   0.0466 *
x2            0.2163     1.0369   0.209   0.8352
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.966 on 97 degrees of freedom
Multiple R-squared:  0.1408,    Adjusted R-squared:  0.1231
F-statistic: 7.947 on 2 and 97 DF,  p-value: 0.0006365
```

Regression coefficients are,

$$\hat{\beta}_0 = 2.4688, \hat{\beta}_1 = 1.3295, \hat{\beta}_2 = 0.2163$$

The regression coefficients are relatively close to the true coefficients. But the standard error of this model is high.

We can reject null hypothesis $H_0 : \beta_1 = 0$, as the p value is 0.0466. Which is much less than 5%.

We can not reject null hypothesis $H_0 : \beta_2 = 0$. Because the p value is 0.8352 which is more than standard cut-off 5%.

d. To fit a least square regression to predict Y using only x1 we need to use following command:

$lm.fit = lm(y\ x1)$
$summary(lm.fit)$

Results:

```
Call:
lm(formula = y ~ x1)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.75787 -0.56934 -0.04168  0.64766  2.24483 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.4649     0.2102  11.725  < 2e-16 ***
x1            1.4445     0.3610   4.001 0.000123 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9612 on 98 degrees of freedom
Multiple R-squared:  0.1404,	Adjusted R-squared:  0.1316 
F-statistic: 16.01 on 1 and 98 DF,  p-value: 0.0001226
```

Yes, we can reject the null hypothesis for the regression coefficient as the p-value 0.000123 for its t-statistic is very small.

e. To fit a least square regression to predict Y using only x2 we need to use following command:

$lm.fit = lm(y\ x2)$

$summary(lm.fit)$

Results:

```
Call:
lm(formula = y ~ x2)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.90820 -0.63486  0.05152  0.62718  2.24278 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.7084     0.1783  15.186  < 2e-16 ***
x2            1.9618     0.5792   3.387  0.00102 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9809 on 98 degrees of freedom
Multiple R-squared:  0.1048,	Adjusted R-squared:  0.09566 
F-statistic: 11.47 on 1 and 98 DF,  p-value: 0.001018
```

Yes, we can reject the null hypothesis for the regression coefficient as the p-value 0.00102 for its t-statistic is very small.

f. The results obtained in part c,d,e does not contradict each other. From part b we can see that x1 and x2 are collinear. Therefore when we try to fit the linear regression model with both x1 and x2 it is difficult to spot individual effect.
But when we fit the same model separately on x1 and x2 then the relationship between Y and each predictor is more visible.

g. We modify our model to accommodate another observation. We run the following commands.

$x1 = c(x1, 0.1)$
$x2 = c(x2, 0.8)$
$y = c(y, 6)$
$lm.fitm = lm(y\ x1 + x2)$
$summary(lm.fitm)$

Result:

```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.7825 -0.6509  0.0189  0.6767  2.3316 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.5751     0.2137  12.048   <2e-16 ***
x1            0.3346     0.5471   0.612   0.5422    
x2            1.8798     0.8293   2.267   0.0256 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9928 on 98 degrees of freedom
Multiple R-squared:  0.1454,	Adjusted R-squared:  0.128 
F-statistic: 8.336 on 2 and 98 DF,  p-value: 0.0004535
```

$lm.fitmx1 = lm(y\, x1)$
$summary(lm.fitmx1)$

Result:

```
Call:
lm(formula = y ~ x1)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.8881 -0.5963 -0.0340  0.6105  3.2772 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.5977     0.2179  11.921  < 2e-16 ***
x1            1.2513     0.3760   3.327  0.00123 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.013 on 99 degrees of freedom
Multiple R-squared:  0.1006,	Adjusted R-squared:  0.0915 
F-statistic: 11.07 on 1 and 99 DF,  p-value: 0.001231
```

$lm.fitmx2 = lm(y\, x2)$
$summary(lm.fitmx2)$

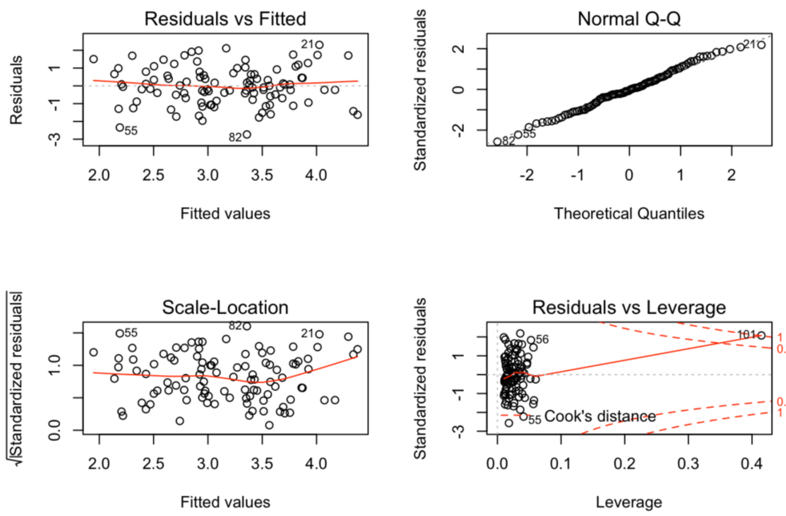
Result:

```
Call:
lm(formula = y ~ x2)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.83733 -0.62236  0.04366  0.61993  2.31160 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.6486     0.1762   15.03  < 2e-16 ***
x2            2.2547     0.5567    4.05 0.000102 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9897 on 99 degrees of freedom
Multiple R-squared:  0.1421,    Adjusted R-squared:  0.1335 
F-statistic:  16.4 on 1 and 99 DF,  p-value: 0.0001019
```

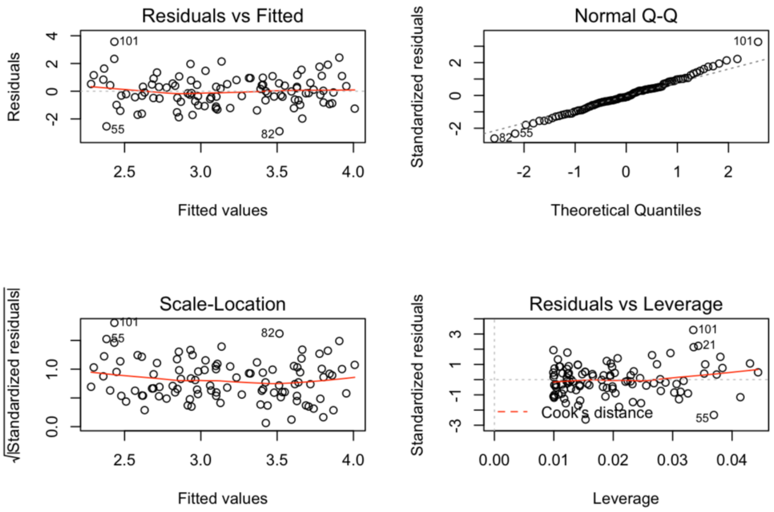
OBSERVATION :

In the first model, x1 becomes statistically insignificant and x2 becomes statistically significant because of the change in p-values between the two linear regressions.

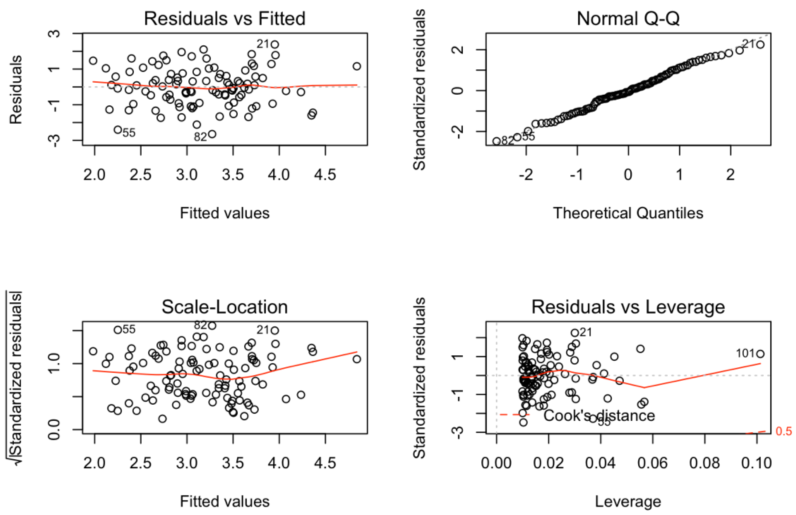
par(mfrow=c(2,2))
plot(lm.fitm)



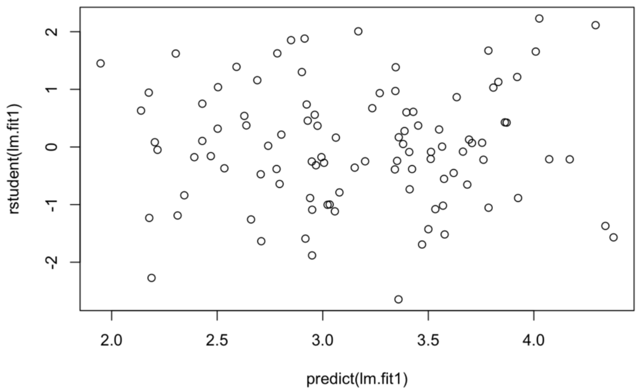
par(mfrow=c(2,2))
plot(lm.fitmx1)



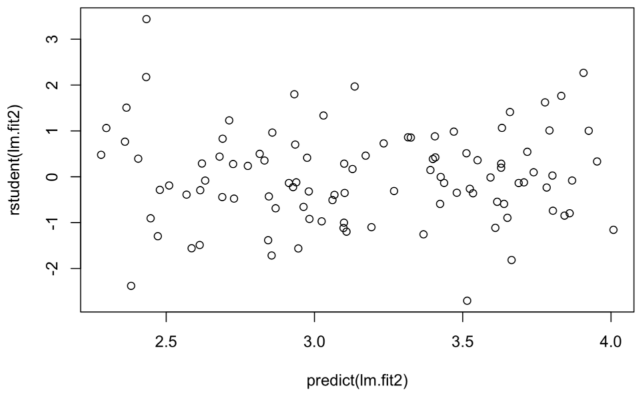

```
par(mfrow=c(2,2))
plot(lm.fitmx2)
```



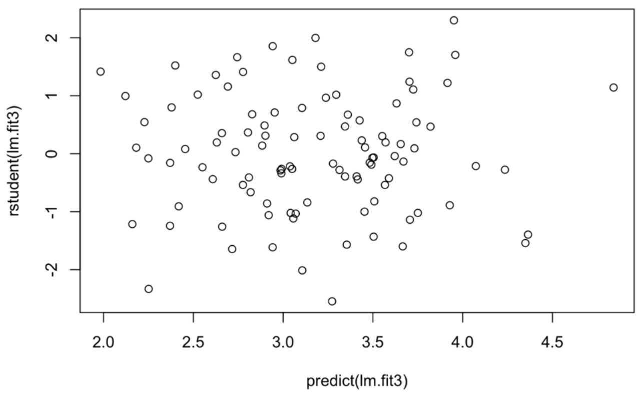

```
plot(predict(lm.fitm), rstudent(lm.fitm))
```



plot(predict(lm.fitmx1), rstudent(lm.fitmx1))



plot(predict(lm.fitmx2), rstudent(lm.fitmx2))



In the model where we use both x1 and x2 as predictor the last point is high lever-

age point.
In the model where we use only x1 as predictor the last point is an outlier.
In the model where we use only x2 as predictor the last point is a high leverage point.

2. To use Bayes classifier, we have to find the class (k) for which,

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)}{\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} \exp(-\frac{1}{2\sigma_l^2}(x-\mu_l)^2)}$$

is largest. As the log function is monotonically increasing, it is equivalent to finding k for which,

$$\log(p_k(x)) = \frac{\log(\pi_k) + \log(\frac{1}{\sqrt{2\pi}\sigma_k}) + -\frac{1}{2\sigma_k^2}(x-\mu_k)^2}{\log(\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} \exp(-\frac{1}{2\sigma_l^2}(x-\mu_l)^2))}$$

is largest. Thus we have,

$$\log(p_k(x)) \log(\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} \exp(-\frac{1}{2\sigma_l^2}(x-\mu_l)^2)) = \log(\pi_k) + \log(\frac{1}{\sqrt{2\pi}\sigma_k}) + -\frac{1}{2\sigma_k^2}(x-\mu_k)^2$$

Therefore,

$$\delta(x) = \log(\pi_k) + \log(\frac{1}{\sqrt{2\pi}\sigma_k}) + -\frac{1}{2\sigma_k^2}(x-\mu_k)^2$$

As we can see that $\delta(x)$ is a quadratic function of x. Therefore the Bayes classifier is not linear but quadratic.

3. We have,

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)}{\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)}$$

Now class k contains both yes and no. Therefore we have,

$$p_{yes}(x) = \frac{\pi_{yes} \exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2)}{\sum \pi_l \exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)}$$

It is given that 80% companies issued dividend, hence $\pi_{yes} = 0.80$ and $\pi_{no} = 0.20$
It is also given that $\sigma^2 = 36$

Also mean value of X for companies that issued dividend is μ_{yes} = 10
Therefore substituting values in the original equation we get,

$$= \frac{\pi_{yes}\exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2)}{\pi_{yes}\exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2)+\pi_{no}\exp(-\frac{1}{2\sigma^2}(x-\mu_{no})^2)}$$

$$= \frac{0.80\exp(-\frac{1}{2*36}(x-10)^2)}{0.80\exp(-\frac{1}{2*36}(x-10)^2)+0.20\exp(-\frac{1}{2*36}x^2)}$$

For x=4 we have,

$$p_{yes}(4) = \frac{0.80\exp(-\frac{1}{2*36}(4-10)^2)}{0.80\exp(-\frac{1}{2*36}(4-10)^2)+0.20\exp(-\frac{1}{2*36}4^2)} = 75.2\%$$

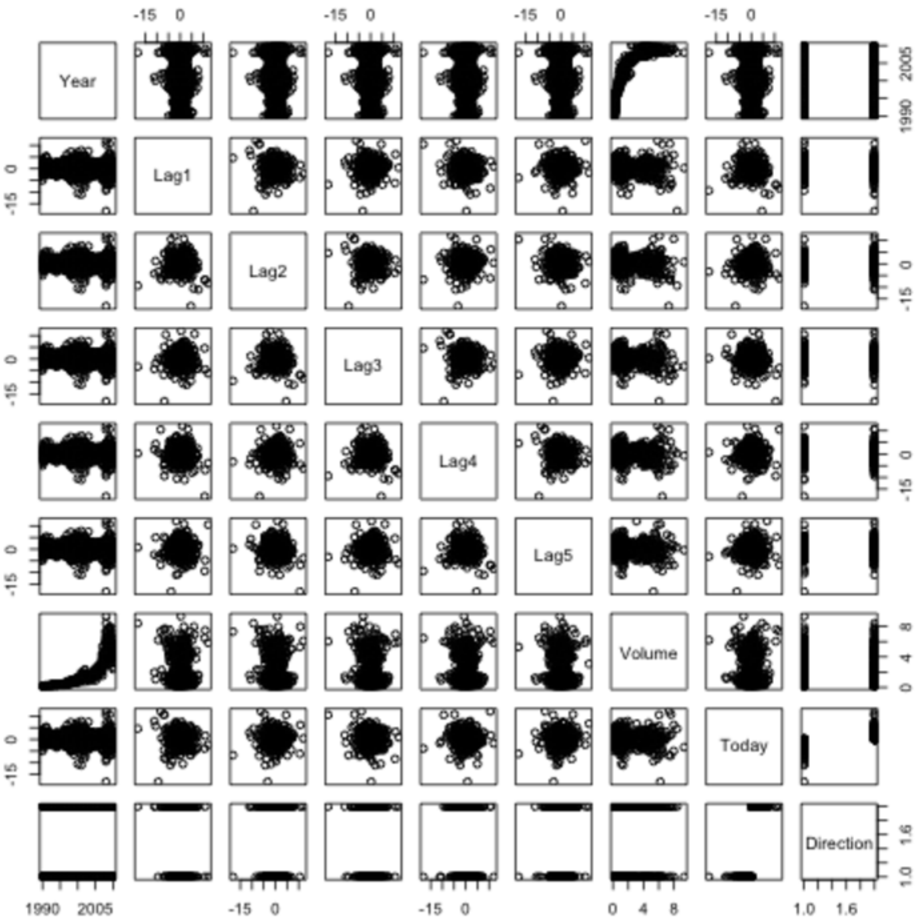
4. a. First we produce the summary of the weekly data set,

library(ISLR)
summary(Weekly)

```
      Year           Lag1               Lag2               Lag3         
 Min.   :1990   Min.   :-18.195   Min.   :-18.195   Min.   :-18.195  
 1st Qu.:1995   1st Qu.: -1.154   1st Qu.: -1.154   1st Qu.: -1.158  
 Median :2000   Median :  0.241   Median :  0.241   Median :  0.241  
 Mean   :2000   Mean   :  0.151   Mean   :  0.151   Mean   :  0.147  
 3rd Qu.:2005   3rd Qu.:  1.405   3rd Qu.:  1.409   3rd Qu.:  1.409  
 Max.   :2010   Max.   : 12.026   Max.   : 12.026   Max.   : 12.026  
      Lag4               Lag5              Volume            Today         
 Min.   :-18.195   Min.   :-18.195   Min.   :0.087   Min.   :-18.195  
 1st Qu.: -1.158   1st Qu.: -1.166   1st Qu.:0.332   1st Qu.: -1.154  
 Median :  0.238   Median :  0.234   Median :1.003   Median :  0.241  
 Mean   :  0.146   Mean   :  0.140   Mean   :1.575   Mean   :  0.150  
 3rd Qu.:  1.409   3rd Qu.:  1.405   3rd Qu.:2.054   3rd Qu.:  1.405  
 Max.   : 12.026   Max.   : 12.026   Max.   :9.328   Max.   : 12.026  
 Direction 
 Down:484  
 Up  :605  
```

Next we produce the graphical representations,

pairs(Weekly)



cor(Weekly[, -9])

```
         Year      Lag1      Lag2     Lag3      Lag4      Lag5    Volume
Year   1.00000 -0.032289 -0.03339 -0.03001 -0.031128 -0.030519  0.84194
Lag1  -0.03229  1.000000 -0.07485  0.05864 -0.071274 -0.008183 -0.06495
Lag2  -0.03339 -0.074853  1.00000 -0.07572  0.058382 -0.072499 -0.08551
Lag3  -0.03001  0.058636 -0.07572  1.00000 -0.075396  0.060657 -0.06929
Lag4  -0.03113 -0.071274  0.05838 -0.07540  1.000000 -0.075675 -0.06107
Lag5  -0.03052 -0.008183 -0.07250  0.06066 -0.075675  1.000000 -0.05852
Volume  0.84194 -0.064951 -0.08551 -0.06929 -0.061075 -0.058517  1.00000
Today  -0.03246 -0.075032  0.05917 -0.07124 -0.007826  0.011013 -0.03308
          Today
Year   -0.032460
Lag1   -0.075032
Lag2    0.059167
Lag3   -0.071244
Lag4   -0.007826
Lag5    0.011013
Volume -0.033078
Today   1.000000
```

It is evident from the results that Year and Volume are correlated.

b. *attach(Weekly)*
glm.fit = glm(Direction Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly, family = binomial)
summary(glm.fit)

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

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.695  -1.256   0.991   1.085   1.458

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.2669     0.0859    3.11   0.0019 **
Lag1         -0.0413     0.0264   -1.56   0.1181
Lag2          0.0584     0.0269    2.18   0.0296 *
Lag3         -0.0161     0.0267   -0.60   0.5469
Lag4         -0.0278     0.0265   -1.05   0.2937
Lag5         -0.0145     0.0264   -0.55   0.5833
Volume       -0.0227     0.0369   -0.62   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

Number of Fisher Scoring iterations: 4
```

Lag2 is the predictor that has some statistical significance as it's p-value is 0.0296, which is less than 5% cut off.

c. *glm.probs = predict(glm.fit, type = "response")*
glm.pred = rep("Down", length(glm.probs))
glm.pred[glm.probs > 0.5] = "Up"
table(glm.pred, Direction)

```
        Direction
glm.pred Down  Up
    Down   54  48
    Up    430 557
```

Percentage of correct predictions on the training data is $(54+557)/(54+557+48+430) = 56.106\%$.

For those weeks when market goes up the percentage of correct predictions is $557/(557+48) = 92.066\%$

.

For those weeks when market goes down the percentage of correct predictions is $54/(430+54) = 11.157\%$.

d. *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
```

Now, *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
```

Percentage of correct prediction on training data = $(9+56)/104 = 62.5\%$.

Correct prediction percentage:
For week when the market goes up = $56/(56+5) = 91.80\%$
For week when the market goes down = $9/(9+34) = 20.93\%$

e. *library(MASS)*
lda.fit = lda(Direction ~ Lag2, data = Weekly, subset = train)
lda.pred = predict(lda.fit, Weekly.0910)
table(lda.pred$class, Direction.0910)

```
       Direction.0910
        Down Up
  Down     9  5
  Up      34 56
```

mean(lda.pred$class == Direction.0910)

we get mean = 0.625. Therefore percentage of correct prediction on training data is 62.5%.

Correct prediction percentage:
For week when the market goes up = $56/(56+5) = 91.80\%$
For week when the market goes down = $9/(9+34) = 20.93\%$

f. *qda.fit = qda(Direction ~ Lag2, data = Weekly, subset = train)*
qda.class = predict(qda.fit, Weekly.0910)$class
table(qda.class, Direction.0910)

```
         Direction.0910
qda.class Down Up
     Down    0  0
     Up     43 61
```

mean(qda.class == Direction.0910)

we get mean = 0.5865. Therefore percentage of correct prediction on training data is 58.65%.

Correct prediction percentage:
For week when the market goes up = $61/(61) = 100\%$
For week when the market goes down = $0/(43) = 0\%$

g. In case of both QDA and LDA we have the assumption that all the classes come from a multivariate normal. There are couple of things we need to check before using QDA.
a. We should ensure that the class means are significantly different.
Here we have single class, hence we should have no problem using QDA.

b. We should also check if we have sufficient number of data points in the training data set.

In this case we have 985 data points.Therefore we can use QDA.

To ensure that all the classes come from a multivariate normal we should perform HZ.test() before using QDA.

h. *library(class)*
train.X = as.matrix(Lag2[train])
test.X = as.matrix(Lag2[!train])
train.Direction = Direction[train]
set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 1)
table(knn.pred, Direction.0910)

mean(knn.pred == Direction.0910)

```
        Direction.0910
knn.pred Down Up
    Down   21 30
    Up     22 31
```

we get mean = 0.5. Therefore percentage of correct prediction on training data is 50%.

Correct prediction percentage:
For week when the market goes up = 31/(61) = 50.82%
For week when the market goes down = 21/(43) = 48.84%

i. Logistic regression and LDA both provides best result on this data with correct prediction percentage of 62.5% i.e 37.5% error rate.

j. Naive Bayes can be considered a better classifier due to it's robustness. But whether it can actually deliver better result depends on various other factors.