Alex Shoop & Anton Libsch

MA 543/DS 502 – HW 2 responses

Problem 1)

First started with (4.3) and showed that it is equal to (4.2)

Problem 2) LDA and QDA

a) If the Bayes decision boundary is linear, in regards to the training set we would expect the QDA to perform better; QDA has more flexibility for a better fit to the training data.

In regards to the test set, the LDA would perform better; LDA fits with the linearity of the Bayes decision boundary, therefore this is more suitable for the test set.

b) If the Bayes decision boundary is non-linear, in regards to the training set data we expect the QDA to perform better; once again QDA has more flexibility in fitting. The same is true for the test data set, QDA would perform better than LDA.
c) As sample size n increases, between QDA and LDA, we would expect the test prediction accuracy of QDA relative to LDA to improve. This is because a more flexible method will be a better fit for the large amount of samples.
d) False. QDA can always lead to overfitting the data, with a few amount of sample data points.

Problem 3)

The average error rate of the 1 nearest neighbor (1NN) classification method is 18%, compared to the 25% average error rate of the logistic regression classification method. Although logistic regression has higher average error rate in this example, it is necessary to look at the test error rate, since the goal is to find the method that is best for classification of new observations.

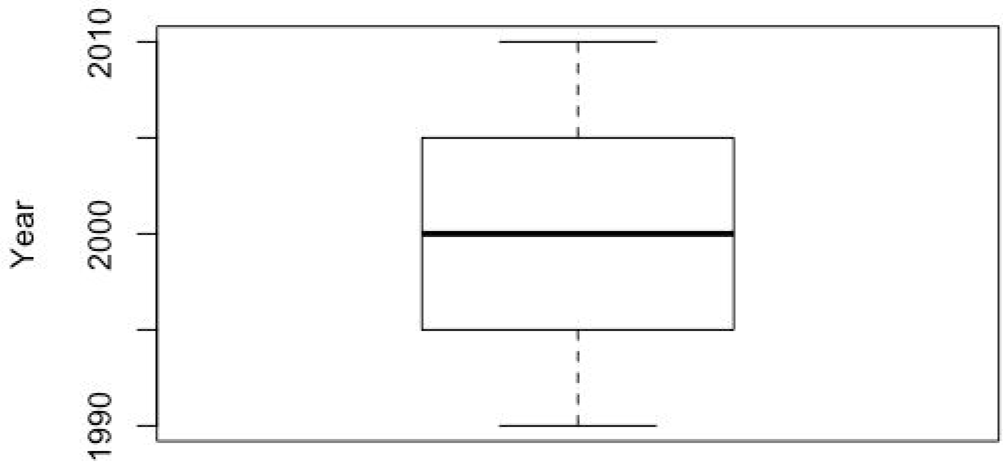
The training error rate of 1NN is always 0, meaning that the test error rate in this example is 36%. Therefore, we should prefer logistic regression as a classification method for new observations.

Problem 4)

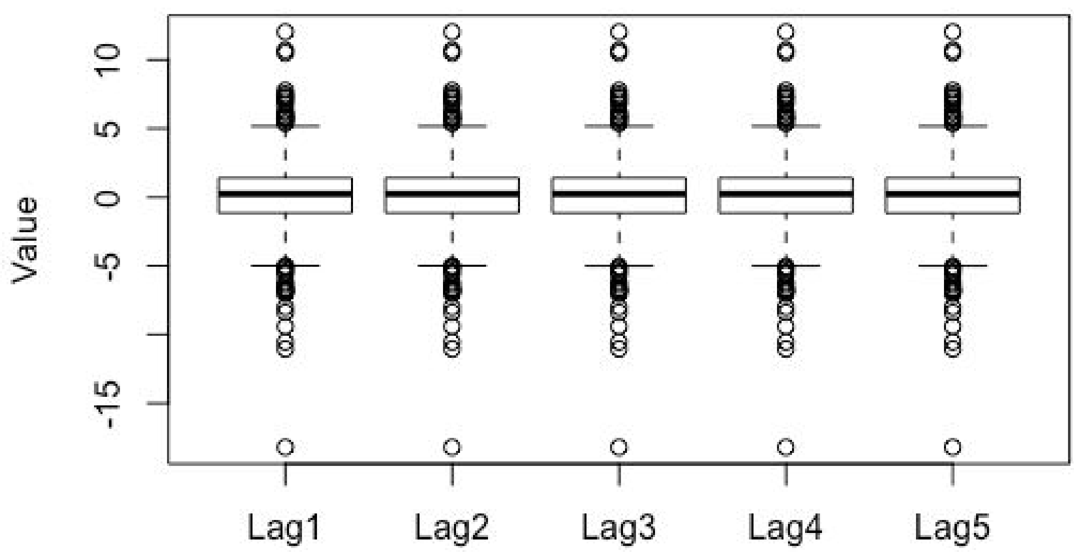
a)

```
      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
```

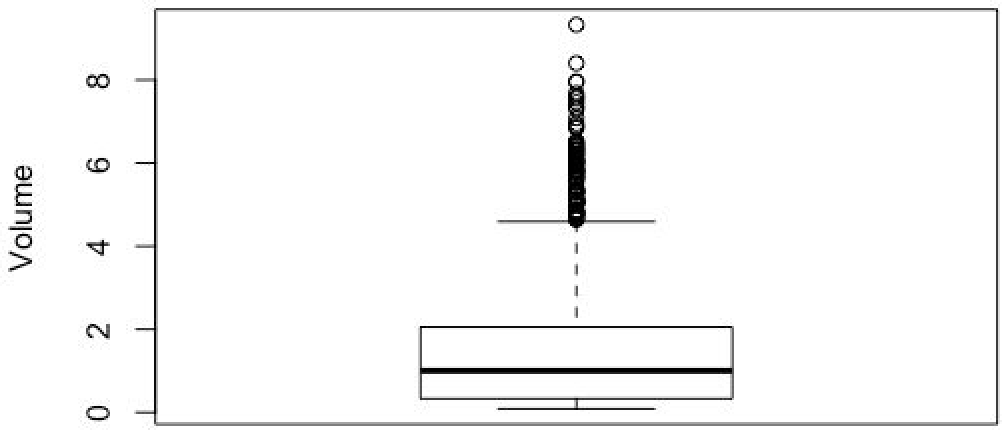
The “Year” variable is symmetric with no outliers.



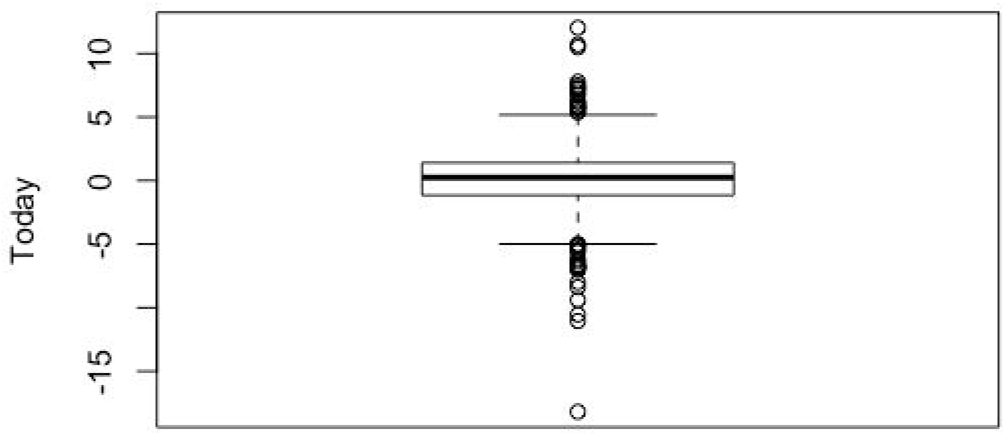
Each of the “Lag” variables have very similar distributions, centers, and spreads; this is thanks to the repeated % returns for each week (ie, week1’s Lag1 is the same as week2’s Lag2). They are all approximately symmetric with many outliers. Their distributions appear to be perfectly symmetric in the box plots, but there is their medians are a bit larger than their means, showing some left skew.



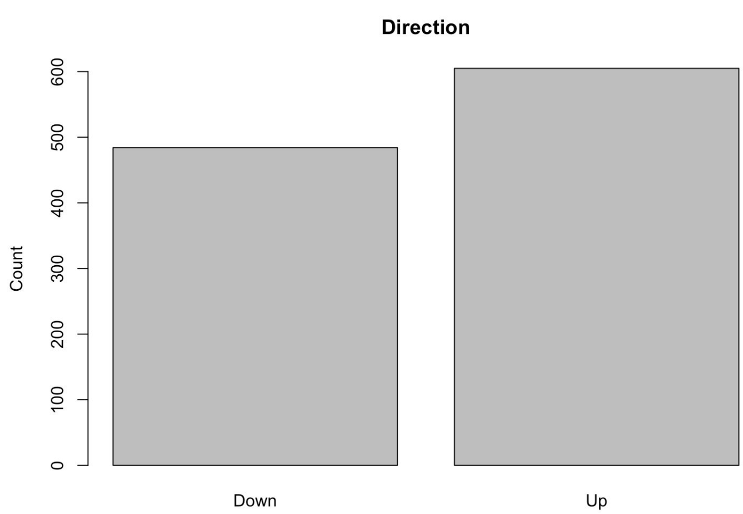
The “Volume” variable is very skewed right, as shown in the box plot and by the mean being higher than the median. There are also outliers.



The “Today” variable appears is approximately symmetric, but the mean being smaller than the median shows that there is some left skew. This distribution has many outliers.



Looking at the “Direction” variable, the visual boxplot indicates that there were more weeks where the stock % returns went UP.



b)

```
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
```

Explanation:
Lag2 is the only predictor that is significant. Based on the table of results, Lag2 has a p-value of 0.0296, which indicates that it is statistically significant. The positive coefficient of Lag2 indicates that if the market has a positive return 2 weeks ago, then is likely to go up this week.

c)
Confusion Matrix:

```
logit4.pred Down  Up
       Down   54  48
       Up    430 557
```

Thus, our confusion matrix shows that, for eg., our model predicted 54 “Down” directions correctly and 557 “Up” directions correctly.
Overall fraction of correct predictions: (54+557)/(54+430+48+557) = **0.561**

Of the model’s errors, a large majority (430) were in predicting “Up”, whereas in reality it should have predicted “Down”. The model incorrectly predicted “Down” 48 times.

There appears to be a bias towards predicting “Up” direction, which is the fault of the logistical regression.

d)

```
Call:
glm(formula = Direction ~ Lag2, family = "binomial", data = Weekly.sub)

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
```

Correction Matrix:

```
             Direction.d.greater
 logit4d.pred Down Up
         Down    9  5
         Up     34 56
```

Overall Correct Predictions: (9+56)/(9+56+5+34) = **0.625**

Therefore, our new logistical regression model actually correctly predicted about 62.5% of the 2009 and 2010 weekly stock directions.

e)

```
Call:
lda(Direction ~ Lag2, data = Weekly, subset = train.d.less)

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
```

Confusion matrix:

```
          Direction.d.greater
 lda.class Down Up
      Down    9  5
      Up     34 56
```

Overall Correct Predictions: (9+56)/(9+56+5+34) = **0.625**

As can be seen, LDA is the same as our logistical regression model fit.

f)

```
Call:
qda(Direction ~ Lag2, data = Weekly, subset = train.d.less)

Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843

Group means:
        Lag2
Down -0.03568254
Up    0.26036581
```

Confusion matrix:

```
          Direction.d.greater
 qda.class Down Up
      Down    0  0
      Up     43 61
```

Overall Correct Predictions: (0+61)/(0+61+0+43) = **0.587**

QDA failed to give a better accuracy of prediction.

g)
Confusion matrix for KNN w/ K = 1:

```
          Direction.d.greater
knn.pred Down Up
    Down   21 30
    Up     22 31
```

Overall Correct Predictions: (21+31)/(21+31+30+22) = **0.5**

Using the KNN method with K = 1, we only get 50% prediction accuracy, which is the lowest out of all the methods so far.

h)
From the above used methods (logistical regression, LDA, QDA, KNN w/ K = 1), we discovered that both logistical regression fit and LDA provided the most accurate predictions for the 2009 & 2010 test dataset (both model fits gave an accuracy of 62.5%).

i)
Now, we want to predict the Direction, using predictors of separate weekly Lag (Lag5) and also the interaction between Lag3 and Lag4.

Logistical regression

```
Call:
glm(formula = Direction ~ Lag5 + Lag3 * Lag4, family = "binomial",
    data = Weekly, subset = train.d.less)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.490  -1.260   1.033   1.094   1.477

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.253e-01  6.489e-02   3.473 0.000515 ***
Lag5        -3.916e-02  2.889e-02  -1.356 0.175174
Lag3         1.136e-05  3.070e-02   0.000 0.999705
Lag4        -2.333e-02  2.938e-02  -0.794 0.427002
Lag3:Lag4    1.333e-02  8.153e-03   1.635 0.101961
---
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: 1349.8  on 980  degrees of freedom
AIC: 1359.8

Number of Fisher Scoring iterations: 3
```

Confusion matrix:

```
             Direction.d.greater
logit4i.pred Down Up
        Down    4  7
        Up     39 54
```

Overall correct predictions: (4+54)/(4+54+7+39) = **0.55769**

LDA

Call:
lda(Direction ~ Lag5 + Lag3 * Lag4, data = Weekly, subset = train.d.less)

Prior probabilities of groups:

```
     Down        Up 
0.4477157 0.5522843 
```

Group means:

```
           Lag5       Lag3       Lag4    Lag3:Lag4
Down 0.21409297 0.17080045 0.15925624 -0.991496916
Up   0.04548897 0.08404044 0.09220956  0.007162822
```

Coefficients of linear discriminants:

```
                  LD1
Lag5      -0.27306444
Lag3       0.01081840
Lag4      -0.14658615
Lag3:Lag4  0.08228727
```

Confusion matrix:

```
            Direction.d.greater
lda4i.class Down Up
       Down    3  5
       Up     40 56
```

Overall correct predictions: (3+56)/(3+56+5+40) = **0.5673**

QDA

Call:
qda(Direction ~ Lag5 + Lag3 * Lag4, data = Weekly, subset = train.d.less)

Prior probabilities of groups:

```
     Down        Up 
0.4477157 0.5522843 
```

Group means:

```
           Lag5       Lag3       Lag4    Lag3:Lag4
Down 0.21409297 0.17080045 0.15925624 -0.991496916
Up   0.04548897 0.08404044 0.09220956  0.007162822
```

Confusion matrix:

```
            Direction.d.greater
qda4i.class Down Up
       Down   13 28
       Up     30 33
```

Overall correct predictions: (13+33)/(13+33+28+30) = **0.4423**

KNN (K = 100)
Confusion matrix:

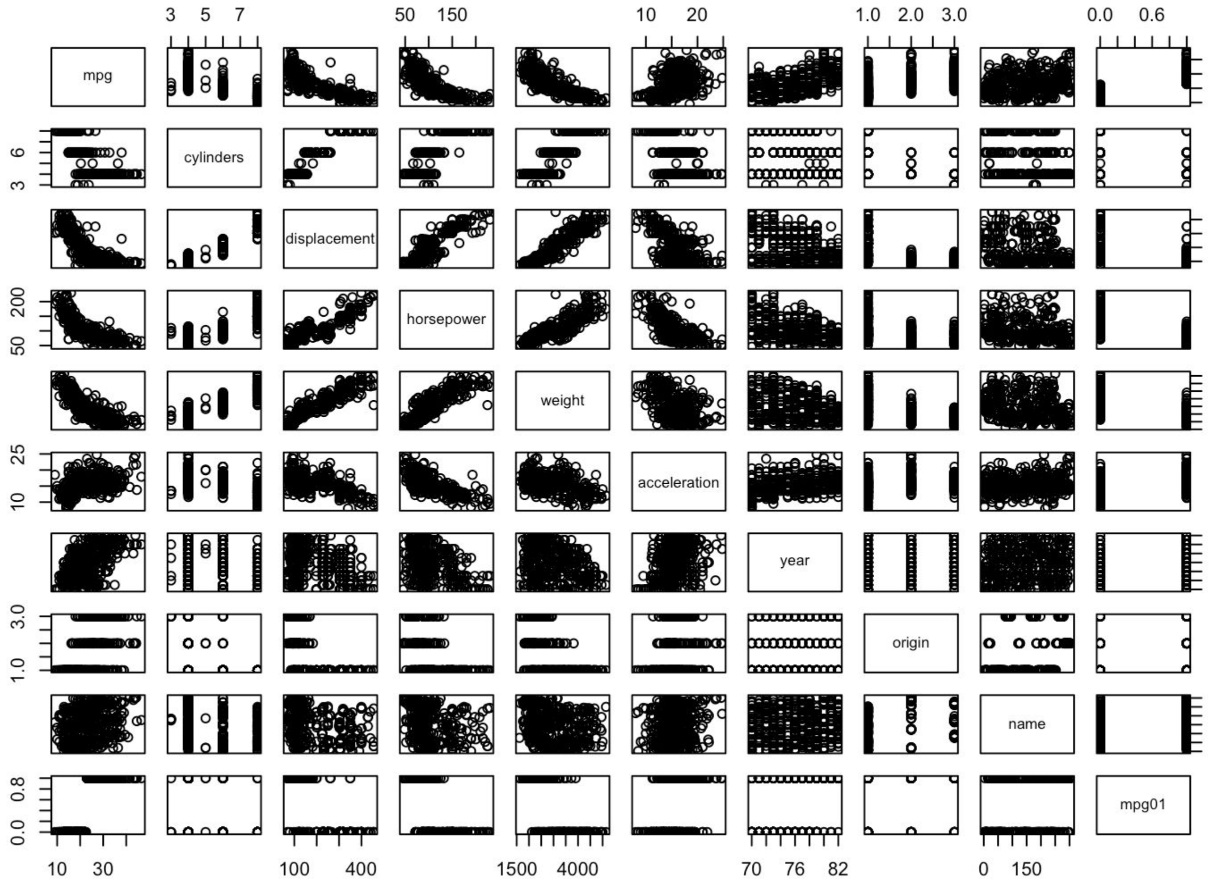
```
           Direction.d.greater
knn.i.pred Down Up
      Down   11 18
      Up     32 43
```

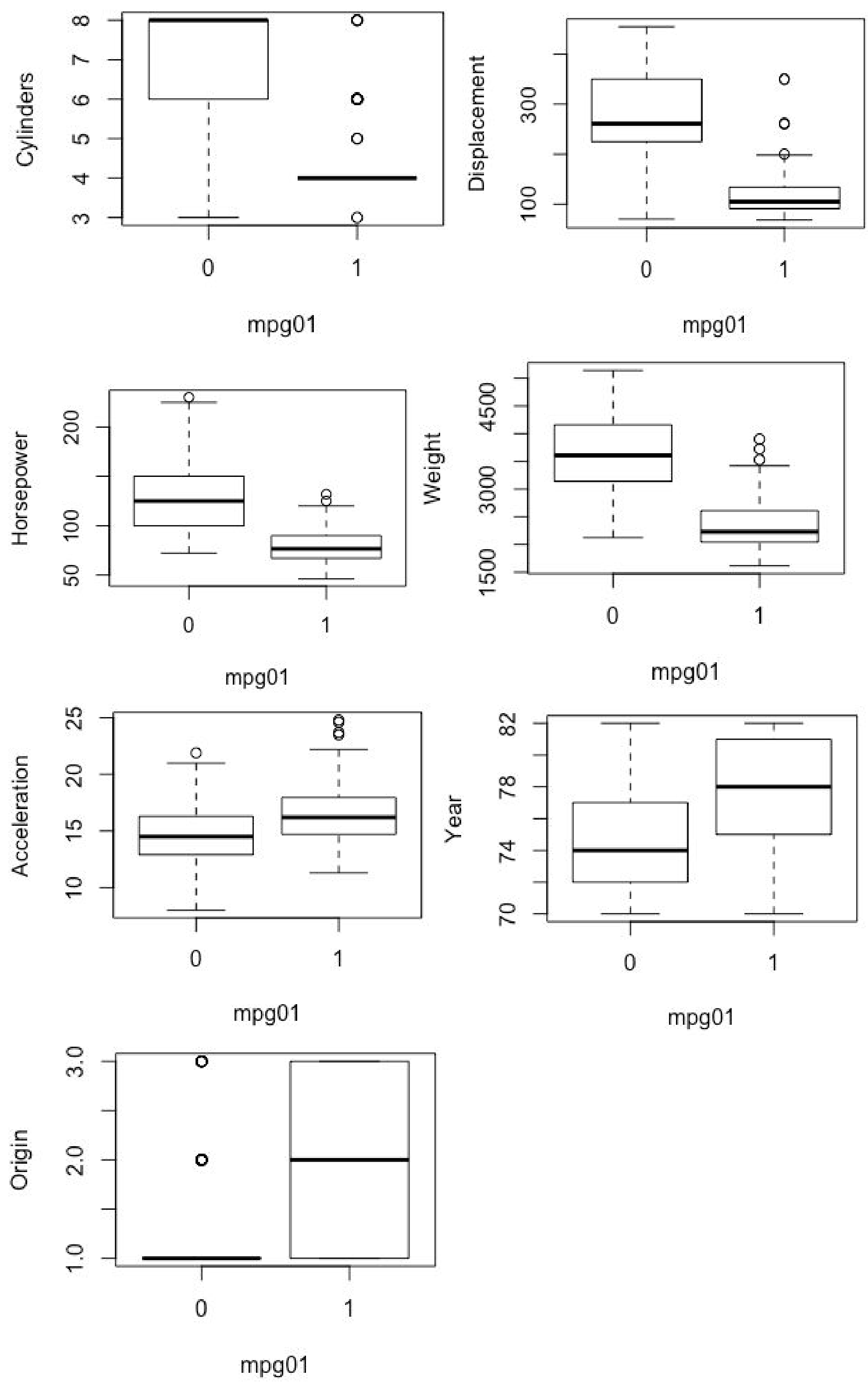
Overall correct predictions: (11+43)/(11+43+18+32) = **0.5192**

Conclusion
After performing all 4 prediction methods on predictor variables: Lag5 + Lag3*Lag4,
we discover that LDA provided a slightly better prediction.
It is still reasonable to deduce that both logistical regression and LDA provide the best test error rate.

Problem 5)

b)
* see bottom-most row and right-most column for “mpg01” scatterplots.


mpg
cylinders
displacement
horsepower
weight
acceleration
year
origin
name
mpg01



Pairs() output shows rough big picture view, and any bulk groupings of mpg01 to the other variables.

The box plots tell us that there are certain variables better associated with mpg01, and in fact would be good in predicting the mpg01 target variable. Specifically: cylinders, displacement, horsepower, and weight.

From our findings, looking at the four chosen predictor variables, none of the interquartile ranges (IQR) overlap. In other words, for eg. if the predictor displacement unit is 100, although it is within the outliers range for mpg01 = 0, because it is within the IQR of mpg01 = 1, there is a high likelihood that a displacement unit of 100 will predict mpg01 to be 1.

d)

```
Call:
lda(mpg01 ~ cylinders + displacement + horsepower + weight, data = Autompg01.df,
   subset = train.c.old)

Prior probabilities of groups:
        0         1
0.6635514 0.3364486

Group means:
  cylinders displacement horsepower   weight
0  6.830986     282.0775  134.02817 3672.106
1  4.055556     105.5347   78.45833 2228.125

Coefficients of linear discriminants:
                       LD1
cylinders    -0.3505402344
displacement -0.0057104148
horsepower    0.0145160830
weight       -0.0009436055
```

Confusion matrix:

```
           mpg01.c.new
 lda.class   0   1
         0  48  13
         1   6 111
```

Overall correct predictions: (48+111)/(48+111+13+6) = **0.8933**

Test error rate = 10.67%

e)

```
Call:
qda(mpg01 ~ cylinders + displacement + horsepower + weight, data = Autompg01.df,
   subset = train.c.old)

Prior probabilities of groups:
        0         1
0.6635514 0.3364486

Group means:
```

```
  cylinders displacement horsepower   weight
0  6.830986     282.0775  134.02817 3672.106
1  4.055556     105.5347   78.45833 2228.125
```

Confusion matrix:

```
          mpg01.c.new
qda.class   0   1
        0  50  20
        1   4 104
```

Overall correct predictions: (50+104)/(50+104+20+4) = **0.8652**
Test error rate = 13.48%

f)

```
Call:
glm(formula = mpg01 ~ cylinders + displacement + horsepower + 
    weight, data = Autompg01.df, subset = train.c.old)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-0.92489  -0.17335   0.08104   0.22947   0.71965  

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.426e+00  1.395e-01  10.221  < 2e-16 ***
cylinders    -8.179e-02  4.077e-02  -2.006 0.046161 *  
displacement -1.332e-03  7.997e-04  -1.666 0.097205 .  
horsepower    3.387e-03  1.141e-03   2.969 0.003340 ** 
weight       -2.202e-04  6.243e-05  -3.526 0.000518 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.09328779)

    Null deviance: 47.776  on 213  degrees of freedom
Residual deviance: 19.497  on 209  degrees of freedom
AIC: 106.62

Number of Fisher Scoring iterations: 2
```

Confusion matrix:

```
                 mpg01.c.new
logit.mpg01.pred   0   1
               0  48  13
               1   6 111
```

Overall correct predictions: (48+111)/(48+111+13+6) = **0.8933**
Test error rate = 10.67%

g)
Confusion matrix (KNN, K = 1)

	mpg01.c.new	
knn.pred	0	1
0	51	30
1	3	94

Overall correct predictions = **0.8146**
Test error rate = 18.54%

Confusion matrix (KNN, K = 10)

	mpg01.c.new	
knn.pred	0	1
0	50	27
1	4	97

Overall correct predictions = **0.8258**
Test error rate = 17.42%

Confusion matrix (KNN, K = 100)

	mpg01.c.new	
knn.pred	0	1
0	52	28
1	2	96

Overall correct predictions = **0.8315**
Test error rate = 16.85%

Therefore, a KNN prediction method with K = 100 is the most accurate.

Problem 6)

Problem 7)

a)

$\frac{n-1}{n}$

b)

$\frac{n-1}{n}$

Neither the value of j nor the bootstrap observation of interest has an effect on the probability, which is why the answer is the same as that of a). This is because each value of j has an equal chance of being selected, and the sampling is done with replacement.

c)

$\frac{n-1}{n}$ $\longrightarrow$ P(Any given bootstrap observation is not the j^{th} observation from the original data)

$(\frac{n-1}{n})^n$ $\longrightarrow$ P(Every bootstrap observation is not the j^{th} observation from the original data)

$(\frac{n-1}{n})^n = (\frac{n}{n} - \frac{1}{n})^n = (1 - \frac{1}{n})^n$

d)

P(The j^{th} observation is in the bootstrap sample)

= 1-P(The j^{th} observation is not in the bootstrap sample)
= $1-(1-\frac{1}{n})^{n}$

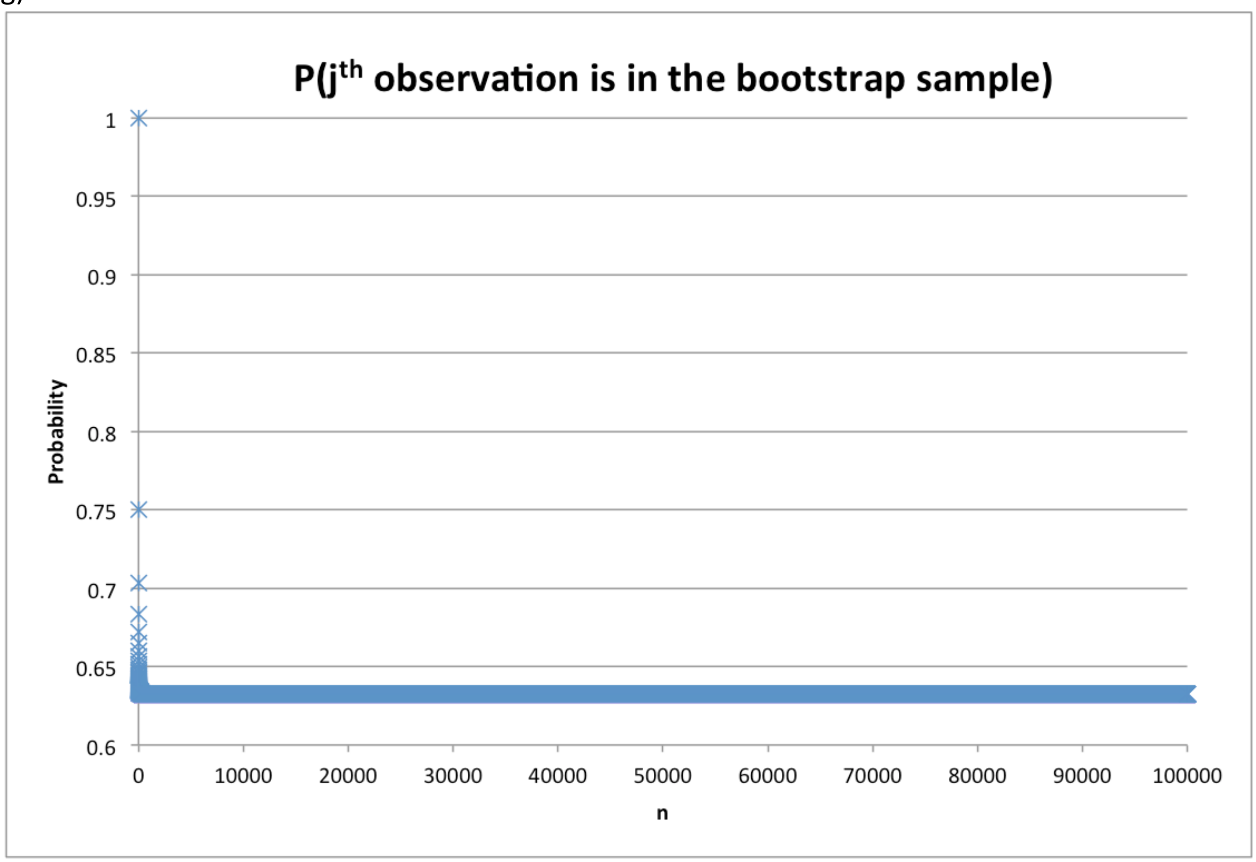
n=5:
$1-(1-\frac{1}{5})^{5}$ = **.672**

e)

n=100:
$1-(1-\frac{1}{100})^{100}$ = **.634**

f)

n=10000
$1-(1-\frac{1}{10000})^{10000}$ = **.632**

g)


As can be seen, as n gets very large, it converges to about 0.6321, which basically the same as the probability for when n = 10,000.

h)

The result we get is **.6319**, which is very close to the calculated probability in e) of .634. The jth value does not matter, since each observation has an equal chance of being selected and the sampling is done with replacement.

Problem 8)

a)

```
Call:
glm(formula = default ~ income + balance, family = "binomial",
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4725  -0.1444  -0.0574  -0.0211   3.7245

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1579.0  on 9997  degrees of freedom
AIC: 1585

Number of Fisher Scoring iterations: 8
```

----

b)

Confusion matrix of validation set approach:

```
            default.b.actual
logit.b.pred   No  Yes
         No  4805  115
         Yes   28   52
```

Test error rate = 2.86%

c)

Ran b) 3 times on random sampling of training set & test set:

Test error rate = 2.36%

Test error rate = 2.8%
Test error rate = 2.68%

Based on our runs, the test error rate seems to hover around **2.6%**.

d)
Confusion matrix:

```
             default.d.actual
logit.d.pred   No  Yes
         No  4811  109
         Yes   23   57
```

Test error rate = 2.64%

When we included the student dummy variable, the test error rate did not really improve by much in comparison to without the dummy var.

Problem 9)
a)

```
> summary(Default)
 default     student        balance           income     
 No :9667   No :7056   Min.   :   0.0   Min.   :  772  
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340  
                       Median : 823.6   Median :34553  
                       Mean   : 835.4   Mean   :33517  
                       3rd Qu.:1166.3   3rd Qu.:43808  
                       Max.   :2654.3   Max.   :73554  
```

```
> summary(logit.default.fit)

Call:
glm(formula = default ~ income + balance, family = "binomial",
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4725  -0.1444  -0.0574  -0.0211   3.7245

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1579.0  on 9997  degrees of freedom
AIC: 1585

Number of Fisher Scoring iterations: 8
```

From our observation of the summary() and glm() outputs, the estimated std errors for the income and balance coefficients are **4.985e-06** and **2.274e-04** respectively.

c)
ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = Default, statistic = boot.fn, R = 100)

Bootstrap Statistics :

	original	bias	std. error
t1*	-1.154047e+01	9.699111e-02	4.101121e-01
t2*	2.080898e-05	6.715005e-08	4.127740e-06
t3*	5.647103e-03	-5.733883e-05	2.105660e-04

----

d)
The bootstrap function returned an estimated std error for income and balance are **4.1277e-06** and **2.10566e-04** respectively.
When comparing with the glm() function output, the estimated standard errors are approximately the same.