CS 5821
February 9, 2017
Mariia Kravtsova
Assignment 3

5. a. QDA will perform better on the training set because it is more flexible than LDA, but LDA will do better on testing set, since there will be less overfitting.

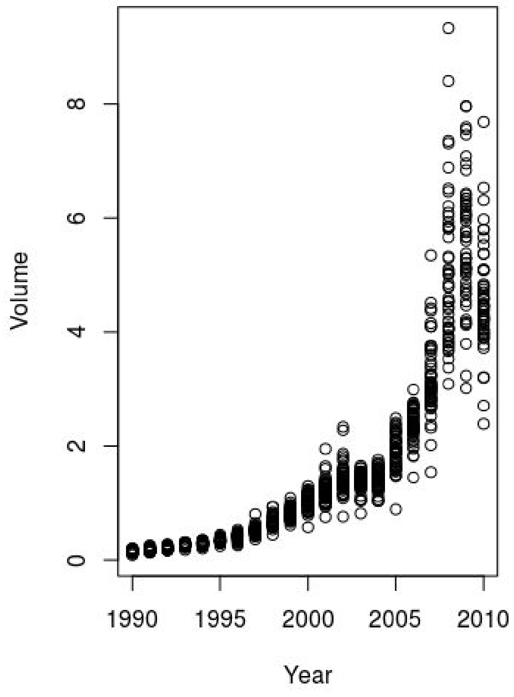
   b. In general we expect the QDA to perform better than LDA on training and testing sets. However, it it does depend how non-linear the decision boundary really is.

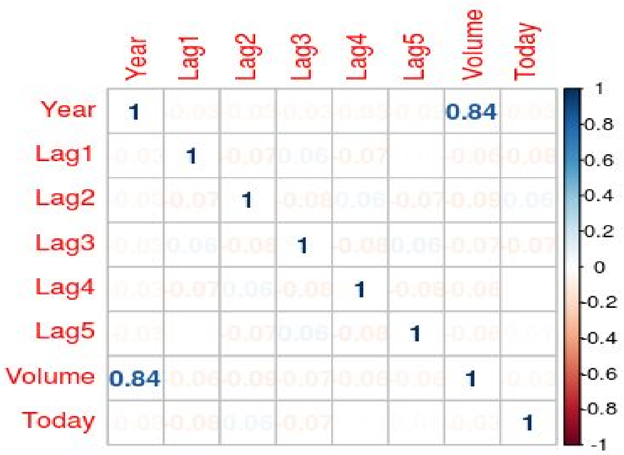
   c. Increasing the sample size will provide us with higher accuracy for the QDA, since it is a recommended classifier for bigger models because of its flexibility.

   d. False, because the QDA's flexibility will just result in overfitting on a smaller sample sizes.
8. We know that when K=1 in the KNN model there is no misclassification on the training set. So the test error in this case is 36%, which is 6% higher than the test error rate of logistic regression. Hence, the logistic regression is a better choice of classification.
9. Using the quantity of odds p(X)/[1 − or + p(X)], we get:

   a. Fraction of 26.8% of people defaulting on their credit card payment.

   b. The odds that she will default are 19%.
10. a. There is a strong correlation between Volume and Year. Volume appears to grow as time goes.





b. With the small p-value the Day 2(Lag2) is statistically significant.

c. Confusion matrix is:

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

We can notice a big mistake made by the poorly fitted model, we can see that it can predict when the market goes up 557(557+48) = 92.1% of the time. But when the market goes down it is only right 54/(430+54) = 11.2% of the time. Hence, the model is only right about 56.1% of the time. This is slightly higher than random guessing.

d. Confusion matrix is:

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

Slight improvement with correct predictions made 62.5% of the time. When the market goes up it is correct 91.8% of the time and when the market goes down it is correct 21% of the time.

e. The results are the same as in d.

f. Confusion matrix is:

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

QDA provides correct predictions 58.7% of the time. For weeks when market goes up the model is always right, and it is never right when the market goes down. So, the market is always up, which is not exactly how it works in real life.

g.

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

KNN provides correctness 50% of the time, just as random guessing it will be about half correct when the market is up or down.

h. The only models providing reasonable prediction rates are LDA and Logistic Regression, and their results are actually the same. So those will be our best models.

i. Increasing the k values made only a small difference, so did QDA with square root. It seems that the LDA and Logistic Regression are still the best bet.

```
> knn.pred = knn(train.X, test.X, train.Direction, k=10)
> table(knn.pred, Direction.0910)
        Direction.0910
knn.pred Down Up
    Down   17 18
    Up     26 43
> mean(knn.pred == Direction.0910)
[1] 0.5769231
>
> knn.pred = knn(train.X, test.X, train.Direction, k=50)
> table(knn.pred, Direction.0910)
        Direction.0910
knn.pred Down Up
    Down   20 22
    Up     23 39
> mean(knn.pred == Direction.0910)
[1] 0.5673077
>
> knn.pred = knn(train.X, test.X, train.Direction, k=100)
> table(knn.pred, Direction.0910)
        Direction.0910
knn.pred Down Up
    Down    9 12
    Up     34 49
> mean(knn.pred == Direction.0910)
[1] 0.5576923
>
> qda.fit = qda(Direction~Lag2+sqrt(abs(Lag2)), subset=train)
> qda.class = predict(qda.fit, Weekly.0910)$class
> table(qda.class, Direction.0910)
         Direction.0910
qda.class Down Up
     Down   12 13
     Up     31 48
> mean(qda.class == Direction.0910)
[1] 0.5769231
>
> lda.fit = lda(Direction ~ Lag2:Lag1, subset=train)
> lda.pred = predict(lda.fit, Weekly.0910)
> mean(lda.pred$class == Direction.0910)
```

```
[1] 0.5769231
>
> glm.fit = glm(Direction~Lag2:Lag1, family=binomial, subset=train)
> glm.probs = predict(glm.fit, Weekly.0910, type="response")
> glm.pred = rep("Down", length(glm.probs))
> glm.pred[glm.probs>.5] = "Up"
> Direction.0910 = Direction[!train]
> table(glm.pred, Direction.0910)
        Direction.0910
glm.pred Down Up
    Down    1  1
    Up     42 60
> mean(glm.pred == Direction.0910)
[1] 0.5865385
```