Vandana Anand
Kratika Agrawal
DS502 - HW2

Question 1:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

$$\frac{p(X)}{1-p(X)} = \frac{\dfrac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}{1-\dfrac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}$$

$$= \frac{\dfrac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}{\dfrac{1+e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}-\dfrac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}} = \frac{\dfrac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}{\dfrac{1}{1+e^{\beta_0+\beta_1 X}}}$$

$$= \frac{e^{\beta_0+\beta_1 X}}{\cancel{1+e^{\beta_0+\beta_1 X}}} \cdot \frac{\cancel{1+e^{\beta_0+\beta_1 X}}}{1} = e^{\beta_0+\beta_1 X}$$

So, $\frac{p(X)}{1-p(X)} = \boxed{e^{\beta_0+\beta_1 X}}$ ✓

Question 2:

a) If the Bayes decision boundary is linear, QDA would perform better on the training set because it is more flexible, which would create a closer fit to the data. On the test set, LDA would perform better because QDA is more flexible, which could overfit the linear data.
b) If the Bayes decision boundary is nonlinear, QDA would perform better on both the training and test sets.
c) QDA would produce a better test prediction accuracy as the sample size increases because it is more flexible so it has the ability to fit more data points closely and variance would not be a problem.

d) False. A more flexible model such as QDA would be too flexible for linear data and overfit it, thus leading to a worse error rate.

Question 3:

Given,

For Logistic Regression: Training Error = 20% & Test Error = 30%

For KNN with k=1, average of Training and Test Error = 18%

i.e. (e_train + e_test)/2 = 18%

Since Training error in KNN for k=1 is 0, therefore:

(0 + e_test)/2 = 18% => e_test = 36%

It implies that the Test Error for KNN with k=1 is 36%, while Test Error for Logistic Regression is 30%.

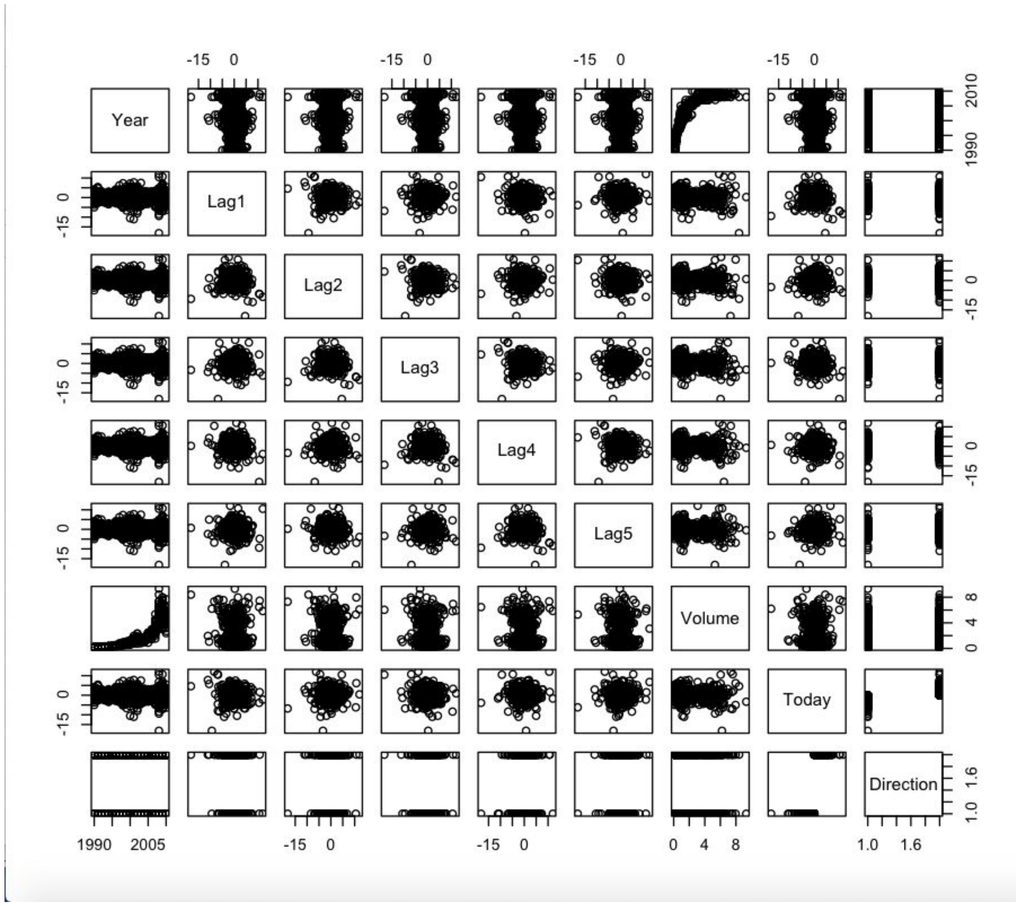
Thus we choose **Logistic Regression** over KNN with k=1 as it has smaller test error.

Question 4:

a) > library(ISLR)

```
> summary(Weekly)
      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
```

> plot(Weekly)




```
> Weekly[,]
   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
1  1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
2  1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
3  1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
4  1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
5  1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
6  1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
7  1990 -1.372  1.178  0.712  3.514 -2.576 0.1517220  0.807        Up
8  1990  0.807 -1.372  1.178  0.712  3.514 0.1323100  0.041        Up
9  1990  0.041  0.807 -1.372  1.178  0.712 0.1439720  1.253        Up
10 1990  1.253  0.041  0.807 -1.372  1.178 0.1336350 -2.678      Down
11 1990 -2.678  1.253  0.041  0.807 -1.372 0.1490240 -1.793      Down
12 1990 -1.793 -2.678  1.253  0.041  0.807 0.1357900  2.820        Up
13 1990  2.820 -1.793 -2.678  1.253  0.041 0.1398980  4.022        Up
14 1990  4.022  2.820 -1.793 -2.678  1.253 0.1643420  0.750        Up
15 1990  0.750  4.022  2.820 -1.793 -2.678 0.1756480 -0.017      Down
```

From these observations, there seems to be a relationship between the Year and Volume. Overall, as the year progresses, the volume increases.

b)

```
> glm.fits=glm(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5 + Volume, family=binomial, data = Weekly)
> summary(glm.fits)

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
```

The predictor Lag2 has the smallest p-value so it seems to be the statically significant.

c)

```
> glm.probs=predict(glm.fits,type="response")
> glm.pred=rep("Down",length(glm.probs))
> glm.pred[glm.probs>0.5]="Up"
> table(glm.pred,Weekly$Direction)

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

The diagonal elements of the confusion matrix indicate the correct predictions, meaning that 54+557 = 611 correct predictions. Dividing this by 1089 (the amount of data) we get 611/1089 = 56.11% which the logistic regression correctly predicted. This means that 43.89% is what the logistic regression predicts wrong, which is a high error rate and very optimistic. When the market goes up, the model predicts it correctly 557/(48+557) = 92.07% of the time whereas when the market goes down, the model predicts it only 54/(430+54) = 11.16% of the time.

d)

```
> train=(Weekly$Year<2009)
> Weekly.20092010=Weekly[!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
```

```
> glm.probs2=predict(fit.glm2, Weeklydata, type="response")
> pred.glm2=rep("Down",length(glm.probs2))
> pred.glm2[glm.probs2>0.5]="Up"
> table(pred.glm2,Weeklydata$Direction)

pred.glm2 Down Up
     Down    9  5
     Up     34 56
```

The logistic regression correctly predicted (9+56)/104 = 62.5%. This means that 37.50% is what the logistic regression predicts wrong. When the market goes up, the model predicts it correctly 56/(56+5) = 91.80% of the time whereas when the market goes down, the model predicts it only 9/(9+54) = 14.29% of the time.

e)

```
> library(MASS)
> lda.fit = lda(Direction ~ Lag2, data = Weekly, subset=train)
> lda.pred = predict(lda.fit, Weeklydata)
> table(lda.pred$class, Weeklydata$Direction)

       Down Up
  Down    9  5
  Up     34 56
```

f)

```
> qda.fit = qda(Direction ~ Lag2, data = Weekly, subset=train)
> qda.pred = predict(qda.fit, Weeklydata)$class
> table(qda.pred, Weeklydata$Direction)

qda.pred Down Up
    Down    0  0
    Up     43 61
```

(0+61)/104 = 58.65% is correctly predicted by QDA.

g)

```
> library(class)
> train.X = as.matrix(Weekly$Lag2[train])
> test.X = as.matrix(Weekly$Lag2[!train])
> train.Direction = Weekly$Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k=1)
> table(knn.pred, Weeklydata$Direction)

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

(21+31)/104 = 50% is correctly predicted by KNN.

h) We can observe that Logistic Regression and LDA have the same and lowest error rates, then QDA, then KNN.

i) Logistic Regression with Lag2*Volume

```
> train=(Weekly$Year<2009)
> Weeklydata=Weekly[!train,]
> fit.glm2=glm(Direction ~ Lag2*Volume, data = Weekly, family=binomial, subset=train)
> summary(fit.glm2)

Call:
glm(formula = Direction ~ Lag2 * Volume, family = binomial, data = Weekly,
    subset = train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.438  -1.263   1.022   1.086   1.521

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.27007    0.09024   2.993  0.00277 **
Lag2         0.05036    0.03998   1.260  0.20781
Volume      -0.05436    0.05279  -1.030  0.30317
Lag2:Volume  0.00151    0.01328   0.114  0.90945
---
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: 1349.4  on 981  degrees of freedom
AIC: 1357.4

Number of Fisher Scoring iterations: 4

> glm.probs2=predict(fit.glm2, Weeklydata, type="response")
> pred.glm2=rep("Down",length(glm.probs2))
> pred.glm2[glm.probs2>0.5]="Up"
> table(pred.glm2,Weeklydata$Direction)

pred.glm2 Down Up
     Down   20 25
     Up     23 36
> mean(pred.glm2 == Weeklydata$Direction)
[1] 0.5384615
```

QDA with Lag2*Volume

```
> qda.fit = qda(Direction ~ Lag2*Volume, data = Weekly, subset=train)
> qda.pred = predict(qda.fit, Weeklydata)$class
> table(qda.pred, Weeklydata$Direction)

qda.pred Down Up
    Down   37 49
    Up      6 12
> mean(qda.pred == Weeklydata$Direction)
[1] 0.4711538
```

KNN with K=32 and K=100

```
> library(class)
> train.X = as.matrix(Weekly$Lag2[train])
> test.X = as.matrix(Weekly$Lag2[!train])
> train.Direction = Weekly$Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k=32)
> table(knn.pred, Weeklydata$Direction)

knn.pred Down Up
    Down   21 24
    Up     22 37
> #KNN with k = 32
> library(class)
> train.X = as.matrix(Weekly$Lag2[train])
> test.X = as.matrix(Weekly$Lag2[!train])
> train.Direction = Weekly$Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k=32)
> table(knn.pred, Weeklydata$Direction)

knn.pred Down Up
    Down   21 24
    Up     22 37
> mean(knn.pred == Weeklydata$Direction)
[1] 0.5576923
>
> #KNN with k = 100
> library(class)
> train.X = as.matrix(Weekly$Lag2[train])
> test.X = as.matrix(Weekly$Lag2[!train])
> train.Direction = Weekly$Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k=100)
> table(knn.pred, Weeklydata$Direction)

knn.pred Down Up
    Down   10 11
    Up     33 50
> mean(knn.pred == Weeklydata$Direction)
[1] 0.5769231
```

The highest correct prediction is the KNN with K =100, then K=32, logistic regression, and then QDA.

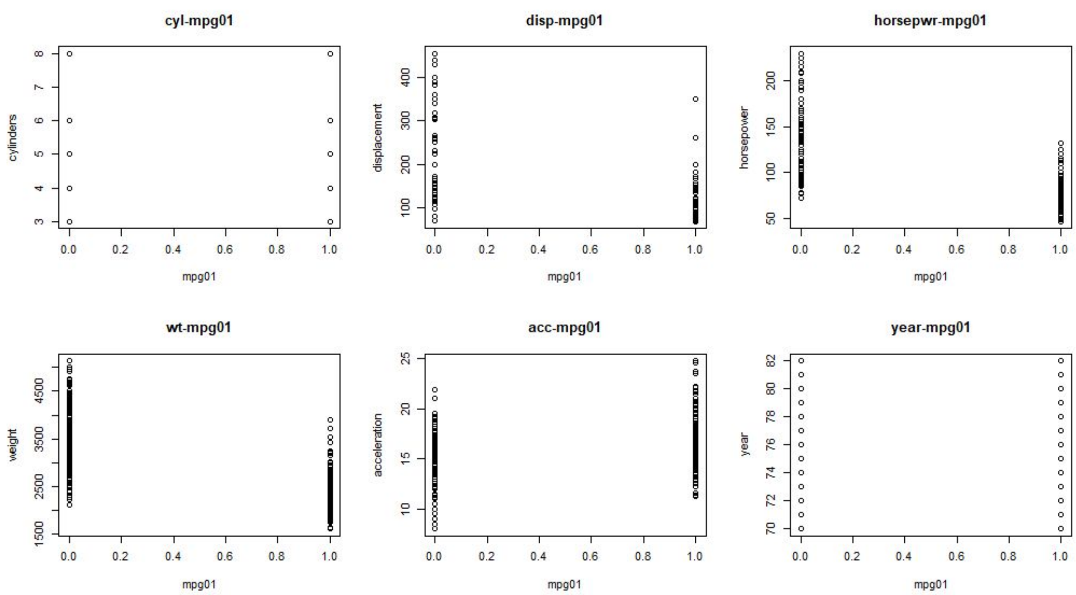
Question 5:

a. Summary:

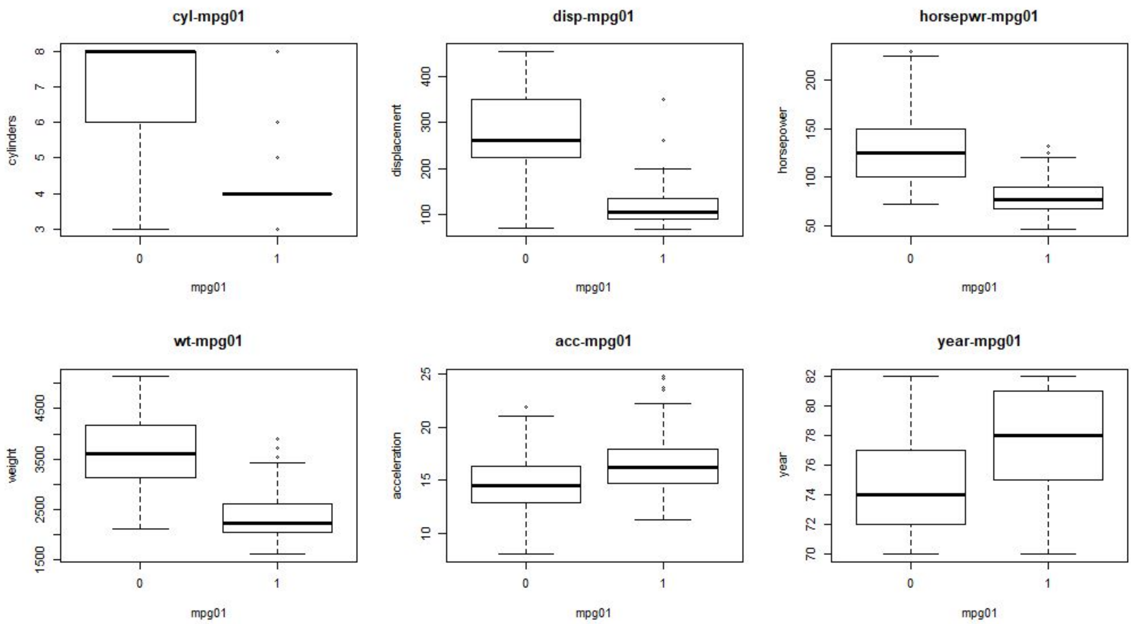
```
> summary(Auto)
      mpg          cylinders      displacement     horsepower        weight      acceleration  
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00  
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78  
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804   Median :15.50  
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean   :15.54  
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02  
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80  
                                                                                               
      year           origin                      name    
 Min.   :70.00   Min.   :1.000   amc matador       :  5  
 1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5  
 Median :76.00   Median :1.000   toyota corolla    :  5  
 Mean   :75.98   Mean   :1.577   amc gremlin       :  4  
 3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4  
 Max.   :82.00   Max.   :3.000   chevrolet chevette:  4  
                                 (Other)           :365  
>
```

```
> table(mpg01)
mpg01
  0   1 
196 196 
```

b. Scatter-plot of Quantitative parameters with mpg01



Boxplot of Quantitative parameters with mpg01:



Visualizing both Scatter-plot and Boxplot of various parameters with mpg01 makes it clear that there is an inverse relation of cylinders, displacement, horsepower and weight with mpg01. Thus these features(cylinders, displacement, horsepower, weight) seem most useful in predicting mpg01.

c. Split is done as 75% Training Dataset and 25% Test Dataset

d. LDA:

```
> lda.fit
Call:
lda(mpg01 ~ cylinders + displacement + horsepower + weight, data = training_set)

Prior probabilities of groups:
        0         1 
0.4829932 0.5170068 

Group means:
  cylinders displacement horsepower   weight
0  6.887324     277.9014  131.26056 3672.627
1  4.190789     114.8388   78.27632 2335.776

Coefficients of linear discriminants:
                       LD1
cylinders    -0.5115464475
displacement -0.0019414455
horsepower    0.0050823545
weight       -0.0009344792
```

```
> table(lda.pred$class,test_set$mpg01)

     0  1
  0 43  2
  1 11 42
```

Test Error:

```
> mean(lda.pred$class != test_set$mpg01)
[1] 0.1326531
```

LDA Test Error = 13.26%

e. QDA:

```
> qda.fit
Call:
qda(mpg01 ~ cylinders + displacement + horsepower + weight, data = training_set)

Prior probabilities of groups:
        0         1 
0.4829932 0.5170068 

Group means:
  cylinders displacement horsepower   weight
0  6.887324     277.9014  131.26056 3672.627
1  4.190789     114.8388   78.27632 2335.776
```

```
> table(qda.pred$class,test_set$mpg01)

     0  1
  0 44  2
  1 10 42
```

Test Error

```
> mean(qda.pred$class != test_set$mpg01)
[1] 0.122449
```

QDA Test Error = 12.24%

f. Logistic Regression:

```
> glm.fit

Call:  glm(formula = as.factor(mpg01) ~ cylinders + displacement + horsepower + 
    weight, family = binomial, data = training_set)

Coefficients:
 (Intercept)     cylinders  displacement    horsepower        weight  
   13.153560     -0.289753     -0.005771     -0.043662     -0.002307  

Degrees of Freedom: 293 Total (i.e. Null);  289 Residual
Null Deviance:      406.5 
Residual Deviance: 156.9        AIC: 166.9
```

```
> table(glm.pred,test_set$mpg01)

glm.pred  0  1
       0 58 40
```

Test Error

```
> mean(glm.pred!=test_set$mpg01)
[1] 0.4081633
```

Logistic Regression Test Error = 40.8%

g. KNN:

k=1:

```
> knn.pred=knn(train.X,test.X,train.mpg01,k=1)
> table(knn.pred,test_set$mpg01)

knn.pred  0  1
       0 45  9
       1  6 38
```

Test Error:

```
> mean(knn.pred!=test_set$mpg01)
[1] 0.1530612
```

=15.3%

k=10:

```
> knn.pred=knn(train.X,test.X,train.mpg01,k=10)
> table(knn.pred,test_set$mpg01)

knn.pred  0  1
       0 45  4
       1  6 43
```

Test Error:

```
> mean(knn.pred!=test_set$mpg01)
[1] 0.1020408
```

=10.2%

k=50:

```
> knn.pred=knn(train.X,test.X,train.mpg01,k=50)
> table(knn.pred,test_set$mpg01)

knn.pred  0  1
       0 45  5
       1  6 42
```

Test Error:

```
> mean(knn.pred!=test_set$mpg01)
[1] 0.1122449
```

=11.22%

k=100:

```
> knn.pred=knn(train.X,test.X,train.mpg01,k=100)
> table(knn.pred,test_set$mpg01)

knn.pred  0  1
       0 46  6
       1  5 41
```

Test Error:

```
> mean(knn.pred!=test_set$mpg01)
[1] 0.1122449
```

=11.22%

As per this experiment, for k=10, the model has the least test error. Thus, k=10 KNN model is the best.

Question 6:

For minimizing risk, we minimize variance.
Variance is given by:

$$Var(\alpha X + (1-\alpha)Y) \quad — ①$$

as per the property of variance that

$$Var(aX) = a^2 V(X)$$

also $Var(A+B) = Var(A) + Var(B) + 2Cov(A,B)$

Therefore equation ① becomes:

$$\Rightarrow \alpha^2 Var(X) + (1-\alpha)^2 Var(Y) + 2\alpha(1-\alpha) Cov(X,Y)$$

$\Rightarrow$ Let $Var(X) = \sigma_X^2$

$Var(Y) = \sigma_Y^2$

$Cov(X,Y) = \sigma_{XY}$

$$\therefore \alpha^2\sigma_X^2 + (1-\alpha)^2\sigma_Y^2 + 2(\alpha)(1-\alpha)\sigma_{XY}$$

$$\Rightarrow \alpha^2\sigma_X^2 + (1+\alpha^2-2\alpha)\sigma_Y^2 + 2(\alpha-\alpha^2)\sigma_{XY}$$

$$\Rightarrow \alpha^2\sigma_X^2 + (1+\alpha^2-2\alpha)\sigma_Y^2 + (2\alpha-2\alpha^2)\sigma_{XY}$$

For minimizing variance, differentiating wrt α,

$$\frac{\partial Var(\alpha X + (1-\alpha)Y)}{\partial \alpha} = 2\alpha\sigma_X^2 + (2\alpha - 2)\sigma_Y^2 + 2\sigma_{XY} - 4\alpha\sigma_{XY}$$

setting differential term to zero

$$0 = 2\alpha\sigma_X^2 + 2\alpha\sigma_Y^2 - 2\sigma_Y^2 + 2\sigma_{XY} - 4\alpha\sigma_{XY}$$

$$2\sigma_Y^2 - 2\sigma_{XY} = 2\alpha(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})$$

$$\alpha = \frac{2(\sigma_Y^2 - \sigma_{XY})}{2(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})}$$

$$\boxed{\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}}$$

Question 7:

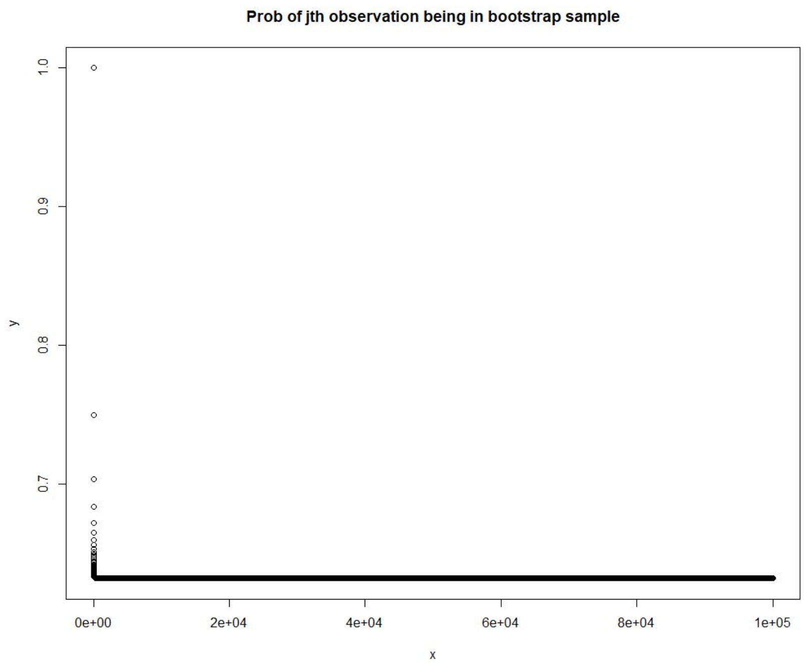
Total observations = n

a. Probability of selecting jth observation in first bootstrap observation from the sample is: 1/n
   Thus, the probability of not selecting jth observation in first bootstrap observation is = (1 - 1/n)
b. Since, bootstrap is a sampling with replacement type of technique, thus the probability of not selecting jth observation in second bootstrap observation is also: (1 - 1/n)
c. Probability of not selecting jth observation in first bootstrap observation = (1 - 1/n)
   Probability of not selecting jth observation in second bootstrap observation = (1 - 1/n)
   Probability of not selecting jth observation in third bootstrap observation = (1 - 1/n)
   .
   .
   .
   .
   Probability of not selecting jth observation in nth bootstrap observation = (1 - 1/n)

   Therefore, total probability of not selecting jth observation at all is $(1 - 1/n)^n$

d. Probability of jth observation to be in the bootstrap sample = $1 - (1 - 1/n)^n$
   For n=5
   Prob(jth observation in bootstrap sample) = $1-(1 - 1/5)^5$ = 0.67232
e. For n=100
   Prob(jth observation in bootstrap sample) = $1-(1 - 1/100)^{100}$ = 0.63396
f. For n=10,000
   Prob(jth observation in bootstrap sample) = $1-(1 - 1/10000)^{10000}$ = 0.63214

g. x = 1:100000
y = 1 - (1 - 1/x)^x
plot(x, y,main="Prob of jth observation being in bootstrap sample")



As the number of observations slightly increases from zero, the probability drops sharply towards 60%. We also observed this scenario in answers d,e,f on this question.

h.

```
> store=rep (NA , 10000)
> for (i in 1:10000) {store[i]=sum(sample (1:100 , rep =TRUE)==4) >0}
> mean(store)
[1] 0.6428
```

Mean comes out to be 0.6428

Thus, the probability of selecting jth observation from n observation set, has a mean value of 64.28%, which is approximately equal to $1 - (1 - 1/100)^{100}$, i.e. $1 - (1 - 1/n)^n$

Question 8:

a.

```
> glm.fit

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

Coefficients:
(Intercept)       income      balance
 -1.154e+01    2.081e-05    5.647e-03

Degrees of Freedom: 9999 Total (i.e. Null);  9997 Residual
Null Deviance:      2921
Residual Deviance: 1579         AIC: 1585
```

b.

ii. Fitting the model

```
> glm.fit2

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

Coefficients:
(Intercept)       income      balance
 -1.090e+01    1.622e-05    5.365e-03

Degrees of Freedom: 4999 Total (i.e. Null);  4997 Residual
Null Deviance:      1484
Residual Deviance: 854.5        AIC: 860.5
```

iii. Making prediction on Validation set:

```
> table(glm.pred,Default[-train,'default'])

glm.pred   No  Yes
     No  4819  101
     Yes   18   62
```

iv. Evaluating misclassifications in Validation set:

```
> mean(glm.pred!=Default[-train,'default'])
[1] 0.0238
```

Total of 2.38% data has been mis-classified in the Validation set.

c. Calling the Logistic Regression lr_eval() function thrice, yields:

```
> lr_eval()
[1] 0.0286
> lr_eval()
[1] 0.0274
> lr_eval()
[1] 0.0278
```

Thus, we see that there is a slight difference in Validation Error when performed three times. It shows that based the observations randomly chosen for training and validation set has some impact on validation error evaluation.

d. Predicting default using income, balance, and student dummy variable.

```
> table(glm.pred4,Default[-train3,'default'])

glm.pred4    No  Yes
      No   4805  112
      Yes    25   58
> mean(glm.pred4 != Default[-train3,]$default)
[1] 0.0274
```

Here, the validation error is 2.74%.Thus, compared to the case above, we see that the validation error has not reduced for some of the cases, i.e. for some of the Training & Validation Set combinations, error has reduced, while didn't change for some. Thus addition of student variable for prediction doesn't have much of an impact in this experiment.

Question 9:

a) On next page

```
> library(ISLR)
> 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  
> set.seed(1)
> glm.fit= glm(default ~ income + balance, family=binomial, data= Default)
> summary(glm.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
```

The glm estimates of the standard errors for B0, B1, and B2 are:

B1 -> 0.4348

B2 -> 4.985 * 10^-6

B3 -> 2.274 * 10^-4

b)

```
boot.fn = function(data, index_obs){
    glmfit2 = glm(default ~ income + balance, data=data,
 family="binomial", subset=index)
    return coef(fit)
}
```

c)

```
> boot.fn = function(data, index_obs){
+   glmfit2 = glm(default ~ income + balance, data=data, family="binomial", subset=index_obs)
+   return (coef(glmfit2))
+ }
> library(boot)
> boot(Default, boot.fn, 50)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Default, statistic = boot.fn, R = 50)


Bootstrap Statistics :
         original        bias     std. error
t1* -1.154047e+01  3.000586e-02 4.212271e-01
t2*  2.080898e-05  7.435755e-08 4.692165e-06
t3*  5.647103e-03 -1.170042e-05 2.287838e-04
>
```

The estimates of the standard errors for B0, B1, and B2 are:
B1 -> 0.42123
B2 -> 4.69217 * 10^-6
B3 -> 2.28784 * 10^-4

d) Both of the estimates are very similar.