Farhana Afroze
UB Person# 50158114

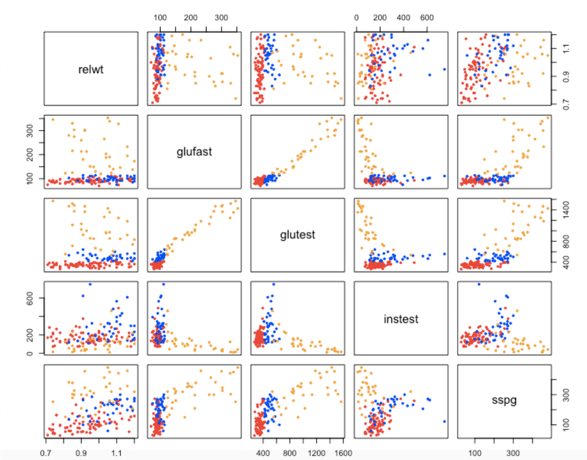
**Problem#1:**

**1(a):**

First, we will look at diabetes data set. In the data set we have 145 rows with 6 columns. There are no missing values in the dataset. Now would like to see summary of the dataset:

```
> summary(Diabetes)
     relwt           glufast        glutest          instest           sspg                   group
 Min.   :0.7100   Min.   : 70   Min.   : 269.0   Min.   : 10.0   Min.   : 29.0   Normal           :76
 1st Qu.:0.8800   1st Qu.: 90   1st Qu.: 352.0   1st Qu.:118.0   1st Qu.:100.0   Chemical_Diabetic:36
 Median :0.9800   Median : 97   Median : 413.0   Median :156.0   Median :159.0   Overt_Diabetic   :33
 Mean   :0.9773   Mean   :122   Mean   : 543.6   Mean   :186.1   Mean   :184.2
 3rd Qu.:1.0800   3rd Qu.:112   3rd Qu.: 558.0   3rd Qu.:221.0   3rd Qu.:257.0
 Max.   :1.2000   Max.   :353   Max.   :1568.0   Max.   :748.0   Max.   :480.0
>
```

Here we see first 5 columns are numeric data and group is the categorical data. In the group data we have three category classes-normal, Chemical_Diabetic and Overt_Diabetic. In the numeric dataset we see data values have different range and we see relwt median is 0.9800 while sspg median is 159.0 so there are big differences between the data. Next we would like to see pairwise scatterplots for all five features with three different classes.




Here we see glufast and glutest is correlated so it has unusual variances and it has elliptical shape so it's not multivariate normal. We see sspg with glutest has elliptical shape and it's correlated so it's nor multivariate normal. We see relwt with sspg is not multivariate normal either as we can see we can see all those 3 classes are correlated. We see most of the variable is correlated and has elliptical shape. As we know if variables are correlated so variance will be distorted so it won't be multivariate normal. We can see sspg and glutest has less correlation so the classes are well separated as it has more variance so it will go into multivariate normal. We see instest with relwt is correlated high as we see all the three classes are mixing up. So, it's also not multivariate normal.

**1(b).**

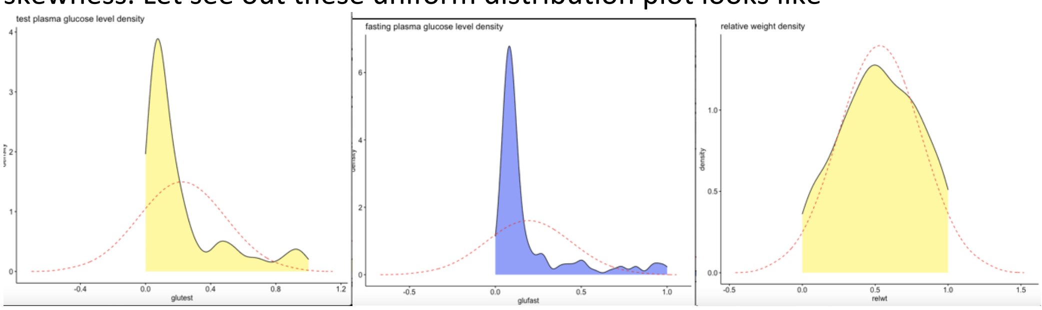
At first, we will split our diabetes dataset into training and testing, 80% for training and 20% for testing. After splitting train data has 6 columns with 116 entries and test data has 29 entries with 6 columns. Next we will normalize our dataset so the value ranges between 0 and 1.

| | relwt | glufast | glutest | instest | sspg |
|---|---|---|---|---|---|
| 1 | 0.61224490 | 0.08833922 | 0.093918399 | 0.172086721 | 0.485587583 |
| 2 | 0.91836735 | 0.57597173 | 0.703618168 | 0.085365854 | 0.951219512 |
| 3 | 0.55102041 | 0.08480565 | 0.016936105 | 0.113821138 | 0.093126386 |
| 4 | 0.26530612 | 0.07067138 | 0.078521940 | 0.257452575 | 0.141906874 |
| 5 | 0.32653061 | 0.08480565 | 0.033872209 | 0.257452575 | 0.452328160 |
| 6 | 0.71428571 | 0.09187279 | 0.150885296 | 0.307588076 | 0.181818182 |
| 7 | 0.48979592 | 0.07067138 | 0.083910701 | 0.170731707 | 0.039911308 |
| 8 | 0.83673469 | 0.07773852 | 0.209391840 | 0.410569106 | 0.527716186 |
| 9 | 0.51020408 | 0.08127208 | 0.082371055 | 0.250677507 | 0.170731707 |
| 10 | 0.40816327 | 0.10600707 | 0.062355658 | 0.285907859 | 0.199556541 |
| 11 | 0.53061224 | 0.05653710 | 0.095458045 | 0.142276423 | 0.124168514 |
| 12 | 0.06122449 | 0.08127208 | 0.093918399 | 0.285907859 | 0.164079823 |
| 13 | 0.93877551 | 0.10600707 | 0.075442648 | 0.233062331 | 0.055432373 |

Next we would like to see skewness of our data. We want each variable or feature to be normally distributed.

```
> skewness(norm_train)
      relwt     glufast     glutest     instest        sspg 
-0.07860834  2.13217891  1.67508016  1.94719209  0.60290794 
>
```

Here we see glufast is high positively skewed. Then we see gluetest and instest has high positive skewness. Let see out these uniform distribution plot looks like-




As we see gluetest and gluefast is not normally distiributes and relwt is almost normally distributed. Relwt has little negative skewness. So now we would like to transform our data so it get better and get normally distributed. So after sqrt transformation skewness of the data looks like this.

```
> skewness(norm_train)
      relwt     glufast     glutest     instest        sspg 
-0.07860834  0.57331128  0.22691640  0.36026190  0.60290794 
>
```

So here we see glufast, glutest and instest got so much better. This was for training data. We found skewness for our testing data. Before transformation testing set skewness looks like this:

```
> skewness(norm_test)
      relwt     glufast     glutest     instest        sspg 
-0.09682163  2.84083107  2.46239418  1.34108888  1.11696481 
>
```

After transformation testing set skewness will look like this:

```
> skewness(norm_test)
      relwt     glufast     glutest     instest        sspg 
-0.09682163  0.34206376 -0.02408516  0.02899955  0.13700712 
>
```

As we see dataset got much better.

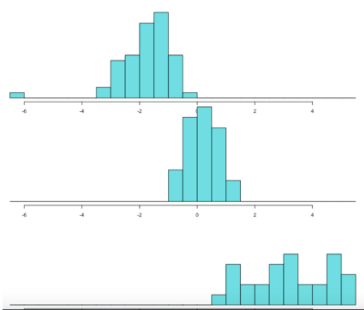
**LDA ON TRAINING AND TESTING:**

**TRAINING DATA:**

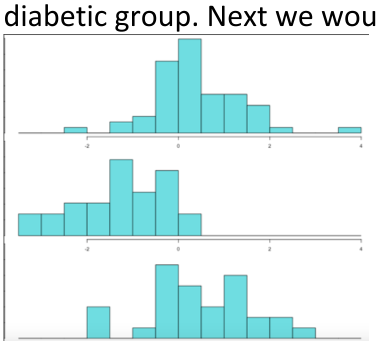
Next we will compute LDA.  So, after applying LDA the summary looks like this:

```
> lda_model
Call:
lda(category_train ~ ., data = new_train)

Prior probabilities of groups:
          Normal Chemical_Diabetic    Overt_Diabetic
       0.4913793         0.2500000         0.2586207

Group means:
                      relwt   glufast   glutest   instest      sspg
Normal            0.4375224 0.5054011 0.4786799 0.4531961 0.1891314
Chemical_Diabetic 0.6903589 0.5565556 0.6413410 0.5863332 0.3921554
Overt_Diabetic    0.5551020 0.8212609 0.8594547 0.3258845 0.6297118

Coefficients of linear discriminants:
              LD1       LD2
relwt   0.1501790 -1.815952
glufast 3.4712378  4.392048
glutest 8.4454607 -6.323467
instest 0.0740351 -5.420856
sspg    1.5717446  1.398540

Proportion of trace:
   LD1    LD2
0.8841 0.1159
```

So here we see prior probabilities of being in normal group is 49%, being in chemical_diabetic is 25% and overt_diabetic is 25.8%. Here we see LD1 is 88% which is good, meaning it's separating classes good and then we see LD2 is 11.5% which does seem poor and shows there are overlapping between classes. Next we predicted our lda model into training data to see how it did.

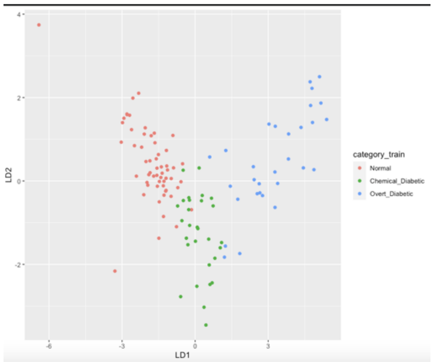


This one is LD1, as we see normal group and overt_diabetic group is well separated where chemical_diabetic and normal group has little overlap and so does between overt and chemical diabetic group. Next we would like to see LD2:



Here in LD2 we see there are overlap between all three classes. So it's not predicting good.

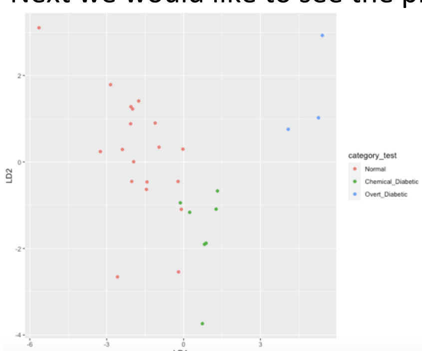
Next we would like to see in the plot how all the classes are separated .




As we see from LD1 3 classes are pretty well separated. There are some red and green classes overlapped but it did good on the training data. For LD2 as we see more overlapping between classes.

**Testing data:**

Next we would like to see the plot to see how lda model did on testing data.




As we see there is overlapping between normal and chemical_diabetic classes but it did pretty good on classifying the dataset into 3 classes. Let's check the accuracy now. The accuracy we got- 0.8965 which is pretty good.

**QDA MODEL AND PREDICTION:**

Next we would like to see how QDA do on classifying to the diabetes dataset. After applying QDA summary of QDA is-

```
qda(category_train ~ ., data = new_train)

Prior probabilities of groups:
           Normal Chemical_Diabetic    Overt_Diabetic
        0.4913793         0.2500000         0.2586207

Group means:
                      relwt   glufast   glutest   instest      sspg
Normal            0.4375224 0.5054011 0.4786799 0.4531961 0.1891314
Chemical_Diabetic 0.6903589 0.5565556 0.6413410 0.5863332 0.3921554
Overt_Diabetic    0.5551020 0.8212609 0.8594547 0.3258845 0.6297118
>
```

Here we see prior probabilities for those 3 classes and group means for all those variables. Next we predicted QDA on the testing data set.

```
> qda_predict_test
$class
 [1] Normal            Normal            Normal            Normal            Normal
 [6] Normal            Normal            Normal            Normal            Normal
[11] Normal            Normal            Normal            Normal            Normal
[16] Chemical_Diabetic Chemical_Diabetic Chemical_Diabetic Chemical_Diabetic Chemical_Diabetic
[21] Chemical_Diabetic Chemical_Diabetic Chemical_Diabetic Overt_Diabetic    Chemical_Diabetic
[26] Overt_Diabetic    Overt_Diabetic    Overt_Diabetic    Overt_Diabetic
Levels: Normal Chemical_Diabetic Overt_Diabetic
```

We would like to see accuracy, so we get idea about how QDA did classifying on the unseen data. The accuracy we got for QDA is 0.7931 which is less than LDA accuracy. LDA accuracy we got 0.8965.  I think LDA accuracy is higher because the dataset is small and LDA care about the equal covariance where equal covariance in QDA is not a major issue, so It performs good on larger dataset.

**1(c):**

We have given individual data where relwt is 1.86, glufast is 184, glutest is 68, instest is 122 and sspg is 544. We would like to see which class of LDA and QDA fall onto this individual data.  So first we make a data frame and then we would like to predict on our LDA model and QDA model. We are not normalizing or transforming our train and test dataset. After predicting we found:

```
$class
[1] Normal
Levels: Normal Chemical_Diabetic Overt_Diabetic

$posterior
     Normal Chemical_Diabetic Overt_Diabetic
1 0.9999998      2.220876e-07   3.981339e-13

$x
        LD1      LD2
1 -5.002032 2.714922

>
```

```
        LD1      LD2
1 -5.002032 2.714922

> predict(qda_model3, df)
$class
[1] Overt_Diabetic
Levels: Normal Chemical_Diabetic Overt_Diabetic

$posterior
        Normal Chemical_Diabetic Overt_Diabetic
1 2.313457e-36      5.818872e-59              1
```

As we see this individual new data point classified as Normal for LDA model and for QDA model it classified as Overt_Diabetic group.

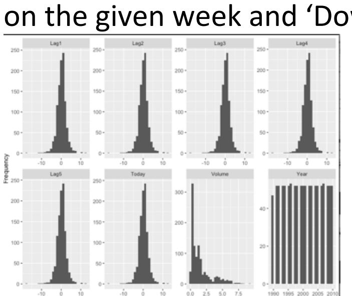
**Problem#2**

**2(a):**

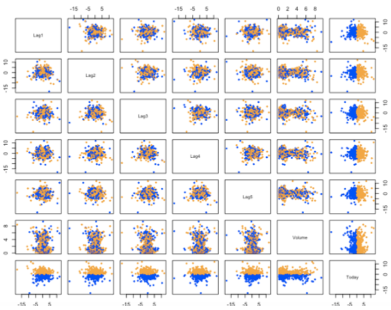
In the weekly dataset from ISLR package we have 1089 rows and 9 features. There are no missing values in the dataset. The summary of the dataset looks like this:

```
> 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
```

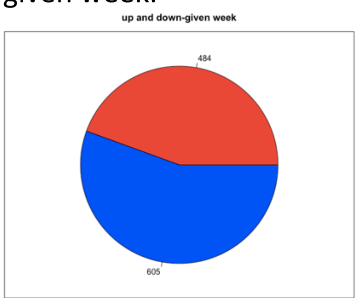
So here we see we have variable year which is numerical but there is 20 categories of year from 1990 to 2010. Lag1 to Today all are numeric variable. And we see categorical variable for ‘Direction’ features and it has ‘up’ and ‘down’ two classes. ‘Up’ if the market has positive return on the given week and ‘Down’ if the market has negative return for given week.




From histogram wee most of the features are normally distributed. Value feature is positively skewed and the year feature is different as it has 20 factor type value.

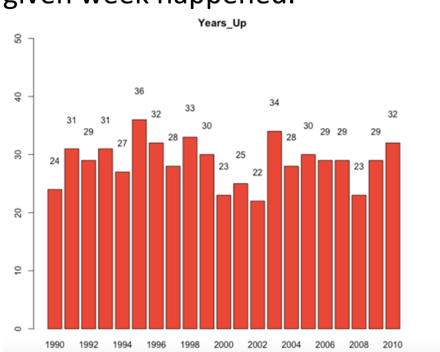


From this plot we see most of the features are co-related as we see two classes are overlapped and so classes aren't well separated. This dataset features has high correlation with each other. Next we would like to see how many up(positive class) and down(negative class) return on a given week.

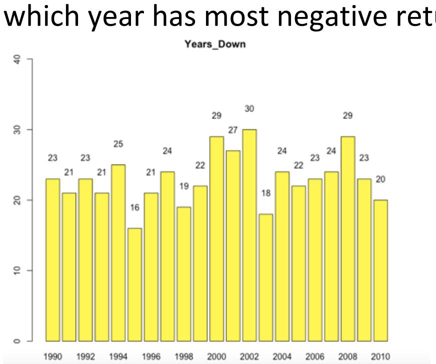



Here the red is for 'Down' class and blue is for 'Up' class. So, we have 484 down(negative) and 605 up(positive) in the dataset.

Next I would like to see which years had more positive(up) and negative(down) return on a given week happened.




We see in year 1995 had most positive(up) return and 2002 had less positive return. Next see which year has most negative return.




Here we see 2002 has most negative(down) return and 1995 had the least negative return.

**2(b).**
Now we would like to build logistic model from this dataset. We would like to drop 'year' and 'today' feature and build model using 5 lag variables, volume as independent variables while direction as a response variable which has two classes. We used the whole dataset to predict. Here is the summary of the logistic model:

```
Call:
glm(formula = Direction ~ ., family = binomial, data = New_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
```

Here we see form Lag1, Lag3, Lag4, Lag5 has negative values and so we can predict that market has negative(down) return for all those variables. So, probability is negative. For volume is also negative and it represents that probability of volume of share traded is also negative or down. Only Lag2 variable has positive return and only this Lag2 variable is significant. We can predict that the market has positive(up) return for the feature 'Lag2' and e^0.05844 = 1.06 is the odd which is likelihood of the event will occur. This means 1 unit increase of Lag2 will increase odd of being positive return by 1.06 times.

**2(C).**
Next we predicted and after predicting we would like to see confusion matrix. We chose 0.5 as our boundary. So, if it's greater than 0.5 then it's positive or return was up and if it's less than 0.5 then it's negative or return was down. So, here is our confusion matrix:

```
> conf_table
      predicted.classes1
       Down  Up
  Down   54 430
  Up     48 557
>
```

Here actual value of Down (negative return) was 54 and we correctly predicted also 54. We see actual values of Down was 430 but we predicted as Up (positive return). Next we see actual value of up is 48 but we predicted as down. Next we see 557 for up was actual value and we predicted correctly as up too. Here total negative return or down is- 484 and total positive return or up is- 605. And we predicted correctly for negative return was 54 and positive return 557. So, we predicted 987 data points as Up (positive class) but it supposed to be 605. That 430 should go in down classes but we predicted wrong and now it's in up class. Next we see the wrong prediction of 48 points as down where 48 points should go in up class. So, percentage of corrected prediction or accuracy is = 611/1089 = 0.56. So, 56% is our accuracy of correctly predicted class.

**2(d).**

Next we choose our training data set from Year 1990 to 2008. In the training data set we have 985 rows with 9 columns/features. For the testing set we choose the data from year 2009 to 2010. We have 104 rows with 9 columns or features. Next we will fit the logistic regression to this data set and picked Log2 as only predictors. So, after fitting the model summary looks like this:

```
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
```

As we see here coefficient for Lag2 is 0.05810. So, for each Lag2 value we can predict the probability is positive return. Here standard error looks small, so we are more confidence about our prediction. Next we predicted how is this logistic model is doing on unseen data using out testing data set which has data points from year 2009 to 2010. After prediction we chose a boundary 0.5, where if it's data points are greater than 0.5 it will be up class otherwise it will be down class. Here is the confusion matrix to see how correctly logistic regression predicted:

```
> conf_table_week = table(test_weekly$Direction, predicted.classes)
> conf_table_week
      predicted.classes
       Down Up
  Down    9 34
  Up      5 56
> 
```

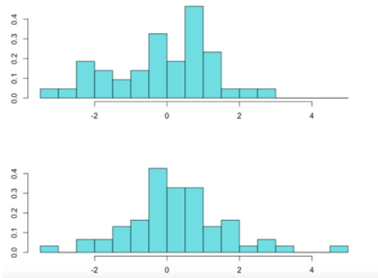
Here as we actual negative(down) return was 9 and predicted as 9 too. But 34 points we predicted as positive(up) return but it should have gone into negative return or down classes. So here we misclassified 34 data points. Next for up or positive class we predicted 56 correctly but misclassified 5 as negative return or down class. Total there is 43 negative return or down class but we predicted 9 correctly and misclassified 34 by predicting as positive. And total positive return or Up class had 61 data points, but we misclassified 5 by predicting as negative. So overall correction fraction of prediction is - 65/104 = 62.5. So, 62.5% is the accuracy of our prediction for this logistic model.

**2(e).**

Next we will fit LDA model to predict how this model performs. We used the same training set and testing set used for logistic model. After fitting the model the summary is-

```
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
> 
```

Here we see prior probabilities for down is 44.8 and up is 55.2 and we got LD1 is 44% which is not so good. We can tell there are overlapping in the dataset. Next we predicted on the test data set to see how this Lda model doing on unseen data.



Here we see the two classes are overlapped. The first histogram is down class and second histogram is up class. The Lda model is not doing so good at classifying between up and down classes. There is lots of misclassification we can see as both classes are overlapped in the plot picture. Next we will see confusion matrix to see how many it misclassified.

```
lda.class.week Down Up
          Down    9  5
          Up     34 56
>
```

Here we see Lda model correctly predicted down or negative return which is 9 and up or positive return is 56. So total correctly classified data is 65 data points out of 104 data points. As we see actual value for down is 34 but we predicted this as up and we predicted 5 data points as 'down' but actual value is up. So total misclassification rate or accuracy rate is 65/104 = 62.5. So, we can say 62.5% data was classified correctly.

**2(f).**

Next we will fit KNN model using the same training and testing set. We will use KNN with k = 1. After predicting KNN model with k = 1 value on the testing set our prediction looks like this-

```
Console  Terminal  Jobs
~/Downloads/
> knn_week = knn(knn_train, knn_test, cl = train_weekly$Direction, k=1)
> knn_week
  [1] Up   Up   Down Down Down Up   Up   Down Up   Down Up   Down Up   Up   Up   Down Down Down
 [19] Down Up   Down Down Up   Down Down Down Up   Down Down Up   Up   Up   Up   Down Up   Up
 [37] Up   Down Down Up   Down Down Down Down Up   Down Down Up   Down Up   Up   Up   Down Down
 [55] Up   Up   Down Up   Down Down Up   Down Down Down Down Down Down Up   Up   Up   Up   Up
 [73] Down Up   Up   Up   Up   Up   Down Up   Up   Up   Down Down Down Up   Up   Up   Up   Up
 [91] Up   Up   Down Down Down Up   Down Up   Down Up   Up   Down Down Up
Levels: Down Up
```

Next we will see the confusion matrix to see how well KNN predicted classes correctly. The confusion matrix-

```
> knn_conf

knn_week Down Up
    Down   21 29
    Up     22 32
>
```

Here we see total correctly classified points are 53. We see 29 we predicted as down or negative return but actual value is up or positive return. Then we predicted 22 as up class or positive return but the actual value should be down or negative return. So total misclassified points here is 29+22 = 51 and the accuracy we got 53/104 = 0.509 or 51% so KNN could classify dataset 51% correctly.

**2(g).**

For Logistic regression we got accuracy- 62.5%
For LDA we got accuracy- 62.5%
For KNN we got accuracy – 51%
As we see LDA and logistic regression has same accuracy. And KNN has least accuracy. So we say Logistic regression and LDA model are doing better than KNN model.