Statistical exploration of data

NWUDO CHIKAEZE FIDELIS JUNIOR

STUDENT ID: 202290064

2023-01-13

```
library(MASS)
library(qcc)

## Package 'qcc' version 2.7

## Type 'citation("qcc")' for citing this R package in publications.

library(corrplot)

## corrplot 0.92 loaded

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.1.8
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1

## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ dplyr::select() masks MASS::select()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force
all conflicts to become errors
```

QUESTION 1: Observations on two response variables are collected for two treatments. The observation vectors [x1, x2] are Treatment 1: (3,3), (1,6), (2,3) Treatment 2: (2,3), (5,1), (3,1), (2,3) a) Calculate the Spooled b) Test H0 : μ1 = μ2 employing a two sample approach with α = 0.01

```
treat1 <- matrix(c(3,1,2,3,6,3), nrow = 3)
treat1

##      [,1] [,2]
## [1,]    3    3
## [2,]    1    6
## [3,]    2    3
```

```r
treat2 <- matrix(c(2,5,3,2,3,1,1,3), ncol = 2)
treat2

##      [,1] [,2]
## [1,]    2    3
## [2,]    5    1
## [3,]    3    1
## [4,]    2    3

n1 <- 3
n2 <- 4

S1 <- cov(treat1)
S2 <- cov(treat2)

# Pooled estimate of sample covariance matrix
Spooled <- (((n1-1)/(n1+n2-2))*S1) + (((n2-1)/(n1+n2-2))*S2)
Spooled

##      [,1] [,2]
## [1,]  1.6 -1.4
## [2,] -1.4  2.0

mean1 <- colMeans(treat1)
mean2 <- colMeans(treat2)

T2 <- (t(mean1-mean2)) %*% solve(((1/n1)+(1/n2))*Spooled) %*% (mean1 - mean2)
T2

##          [,1]
## [1,] 3.870968

F <- qf(1-0.01,2,n1+n2-2-1)

T <- (((n1+n2-2)*2) / (n1+n2-2-1))*F
T

## [1] 45
```

Since my T2 (3.870968) is less than critical value (45) under the null hypothesis, we fail to reject the null hypothesis.

QUESTION 2. Generate a data set with two explanatory variables x1 and x2 from multinomial Normal distri- bution with covariance matrix σ = c(1, .2, .2, 4) in two classes with mean for Class 0 is (3,7) and and Class 1 is (6,10). For the Class 0, generate 50 observations and for Class 1, 50 observations. While generating this data, use the set.seed("99"). Find the linear discriminant function weights. Plot the data with two colors and draw the discriminant function for classification. Also plot the 4 test data (3.68, 5.65), (3.28, 5.20), (3.57, 8.82), (4.64, 7.98) and predict the test data. Use the R program also to predict the test data

```r
set.seed(99)
n <- 50

# Covariance matrix
sigma <- matrix(c(1, 0.2, 0.2, 4), nrow = 2, ncol = 2)


# Generate data for Class 0
class0 <- mvrnorm(n, c(3, 7), sigma)

# Generate data for Class 1
class1 <- mvrnorm(n, c(6, 10), sigma)

C0 <- data.frame(Y= rep(0,n),class0)
C1 <- data.frame(Y= rep(1,n),class1)


df <- rbind(C0,C1)
df
```

```
##     Y       X1        X2
## 1   0 4.591003  7.323978
## 2   0 2.445073  7.999853
## 3   0 2.684351  7.197286
## 4   0 3.453491  7.861046
## 5   0 4.023698  6.203576
## 6   0 3.092518  7.240155
## 7   0 3.406281  5.238671
## 8   0 2.677130  8.004460
## 9   0 3.626129  6.227396
## 10  0 3.570366  4.363663
## 11  0 3.062533  5.498551
## 12  0 3.235471  8.834590
## 13  0 3.540241  8.470044
## 14  0 2.412492  2.002511
## 15  0 3.686958  0.849049
## 16  0 4.324795  6.912603
## 17  0 2.744637  6.225868
## 18  0 2.733505  3.514155
## 19  0 3.501188  7.967849
## 20  0 2.639082  7.567956
## 21  0 2.565800  9.235150
## 22  0 2.621973  8.535930
## 23  0 3.789836  6.828284
## 24  0 2.653887  6.331174
## 25  0 3.848414  7.390745
## 26  0 2.933029  8.112280
## 27  0 2.356889  8.415251
## 28  0 1.554774  5.999949
## 29  0 3.446553  4.224927
```

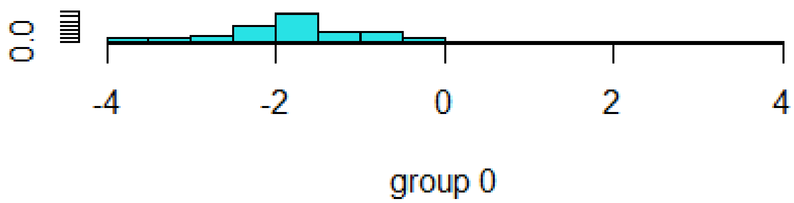
```
## 30  0 2.949271  9.814284
## 31  0 3.145881  9.747030
## 32  0 2.773762  7.919007
## 33  0 4.076785  6.634813
## 34  0 2.132644  7.314753
## 35  0 1.524525  2.490766
## 36  0 2.470628  4.291443
## 37  0 2.911350  6.609400
## 38  0 1.627097  7.227821
## 39  0 3.024405  7.180086
## 40  0 2.370740  7.689779
## 41  0 3.073293  7.262120
## 42  0 2.642413  3.652235
## 43  0 1.114960  6.566006
## 44  0 1.655482  3.972427
## 45  0 3.748626  4.180257
## 46  0 2.881293  4.283284
## 47  0 3.250555  5.133981
## 48  0 2.677161  5.281097
## 49  0 2.520572 10.357905
## 50  0 2.333101  6.732907
## 51  1 7.469773  6.840529
## 52  1 6.484840  8.961967
## 53  1 4.491158  7.664414
## 54  1 4.696785  8.821077
## 55  1 5.258125  7.143089
## 56  1 6.712782  9.617585
## 57  1 6.137532 13.180997
## 58  1 5.466893  9.572890
## 59  1 3.850817  8.991476
## 60  1 5.105644 11.189173
## 61  1 6.080613  9.502926
## 62  1 7.059296 14.021736
## 63  1 4.727180  9.580137
## 64  1 5.869102  3.988182
## 65  1 7.472393  7.503576
## 66  1 7.140383 11.864606
## 67  1 5.245036  7.667080
## 68  1 5.459430  8.330029
## 69  1 5.043570  9.242343
## 70  1 6.917700  6.269495
## 71  1 4.748965 11.276266
## 72  1 6.768502 10.710625
## 73  1 5.286645  8.152155
## 74  1 5.152890  6.717313
## 75  1 6.822309 10.003599
## 76  1 5.388397 11.634677
## 77  1 7.031621 10.057030
## 78  1 4.394722 12.939472
## 79  1 5.574675 13.042667
```

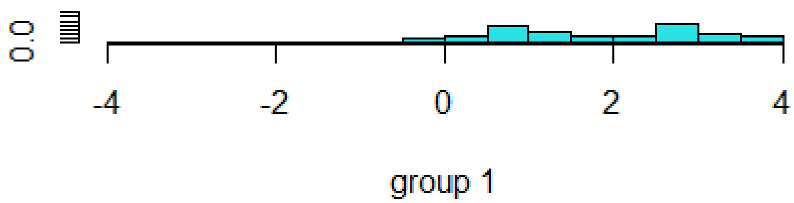
```
## 80  1 5.217067  7.553373
## 81  1 6.808589 10.766968
## 82  1 7.893824  9.439355
## 83  1 6.546334 11.621058
## 84  1 6.925997 11.791458
## 85  1 7.148120  9.052979
## 86  1 7.622595 10.033361
## 87  1 6.612334 11.858975
## 88  1 4.202222 10.946762
## 89  1 5.973736  6.299143
## 90  1 4.944677 10.292652
## 91  1 5.539114 11.104863
## 92  1 5.050976 13.318595
## 93  1 7.728366 10.494864
## 94  1 7.171268 11.447790
## 95  1 6.838125  8.861307
## 96  1 6.985123  8.553408
## 97  1 5.349125  9.530215
## 98  1 8.063451 10.465073
## 99  1 5.719048  8.727354
## 100 1 6.771416 12.933805
```

```
model <- lda(Y ~ X1 + X2, data=df)
model
```

```
## Call:
## lda(Y ~ X1 + X2, data = df)
##
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
## Group means:
##         X1       X2
## 0 2.922533 6.498367
## 1 6.059386 9.791609
##
## Coefficients of linear discriminants:
##          LD1
## X1 0.9774977
## X2 0.1835658
```

```
plot(model)
```



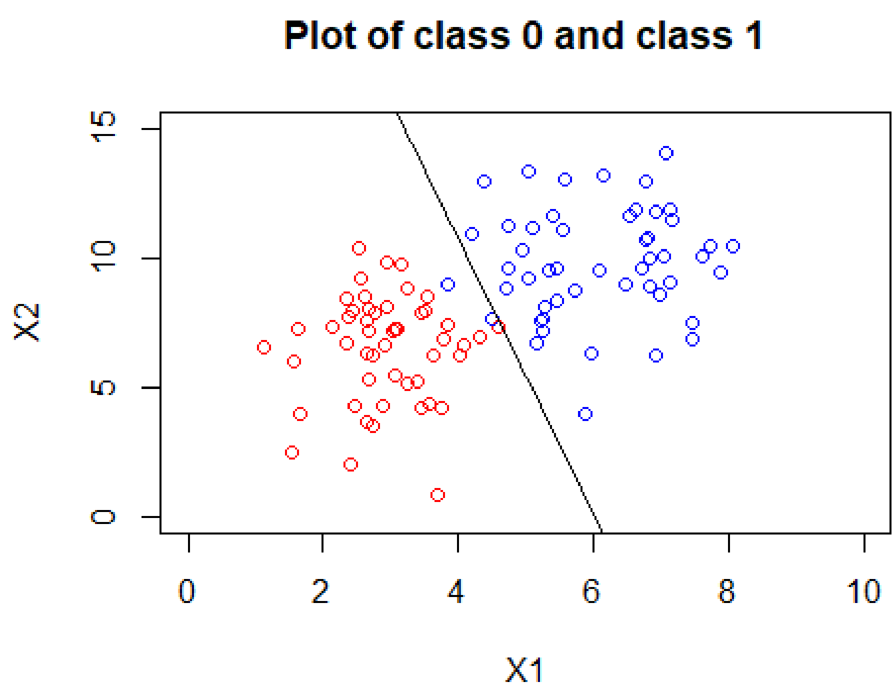



```
# linear discriminant function weights
w = model$scaling
w

##           LD1
## X1 0.9774977
## X2 0.1835658
```

The LDA output indicates that our prior probabilities are 0.5 for the two classes. In other words, 50% of the observations are both in class 1 and class 0.It also provides the group means; these are the average of each predictor within each class, and are used by LDA as estimates of the means of the classes. The coefficients of linear discriminant output provides the linear combination of X1 and X2 that are used to form the LDA decision rule.

```
X0.bar <- model$means[1,]
X1.bar <- model$means[2,]
a <- t(w)  %*% ((X0.bar + X1.bar)/2)
a

##         [,1]
## LD1 5.885044

P1 <- seq(min(df$X1), max(df$X1), 0.1)
P2 <- c(a/w[2,]) - ((w[1,]/w[2,])*P1)
P2

##  [1]  26.12236229  25.58985697  25.05735164  24.52484632  23.99234099
##  [6]  23.45983567  22.92733034  22.39482502  21.86231969  21.32981437
```

```
## [11]  20.79730904  20.26480372  19.73229839  19.19979307  18.66728774
## [16]  18.13478242  17.60227709  17.06977177  16.53726644  16.00476112
## [21]  15.47225579  14.93975047  14.40724514  13.87473982  13.34223449
## [26]  12.80972917  12.27722384  11.74471852  11.21221319  10.67970787
## [31]  10.14720254   9.61469722   9.08219189   8.54968657   8.01718124
## [36]   7.48467592   6.95217059   6.41966526   5.88715994   5.35465461
## [41]   4.82214929   4.28964396   3.75713864   3.22463331   2.69212799
## [46]   2.15962266   1.62711734   1.09461201   0.56210669   0.02960136
## [51]  -0.50290396  -1.03540929  -1.56791461  -2.10041994  -2.63292526
## [56]  -3.16543059  -3.69793591  -4.23044124  -4.76294656  -5.29545189
## [61]  -5.82795721  -6.36046254  -6.89296786  -7.42547319  -7.95797851
## [66]  -8.49048384  -9.02298916  -9.55549449 -10.08799981 -10.62050514

# Plot the above samples and color by class labels
plot(class0, xlim = c(0,10), ylim = c(0,15), xlab = "X1", ylab = "X2", col =
"red", main = "Plot of class 0 and class 1")
points(class1, col = "blue")
lines(P1,P2)
```



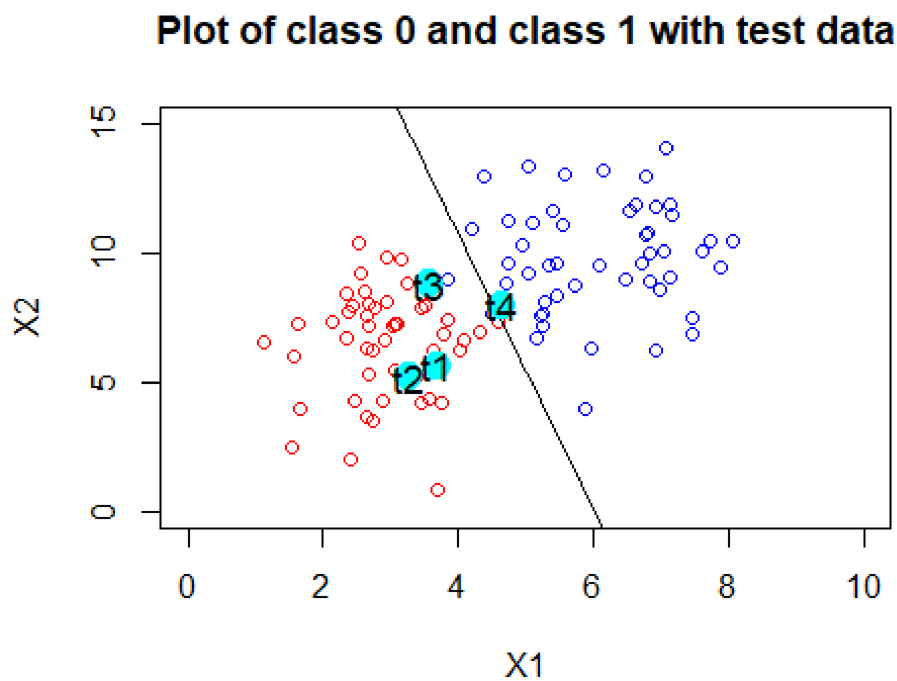
PLOTTING AND PREDICTING TEST DATA POINTS

```
#Test data set
t1 <- c(3.68,5.65)
t2 <- c(3.28, 5.20)
t3 <- c(3.57,8.82)
t4 <- c(4.64,7.98)
```

```
test <- rbind(t1,t2,t3,t4)
test_data <- data.frame(test)
test_data
```

```
##      X1   X2
## t1 3.68 5.65
## t2 3.28 5.20
## t3 3.57 8.82
## t4 4.64 7.98
```

```
# Plot the above samples and color by class labels
plot(class0, xlim = c(0,10), ylim = c(0,15), xlab = "X1", ylab = "X2", col =
"red", main = "Plot of class 0 and class 1 with test data")
points(class1, col = "blue")
lines(P1,P2)
# Add first point of the test dataset
points(t1[1],t1[2],col="cyan1", pch=19, cex=2)
text(t1[1],t1[2],labels ="t1",cex = 1.2)

# Add second point of the test dataset
points(t2[1],t2[2],col="cyan1", pch=19, cex=2)
text(t2[1],t2[2],labels ="t2",cex = 1.2)

# Add third point of the test dataset
points(t3[1],t3[2],col="cyan1", pch=19, cex=2)
text(t3[1],t3[2],labels ="t3",cex = 1.2)

# Add fourth point of the test dataset
points(t4[1],t4[2],col="cyan1", pch=19, cex=2)
text(t4[1],t4[2],labels ="t4",cex = 1.2)
```



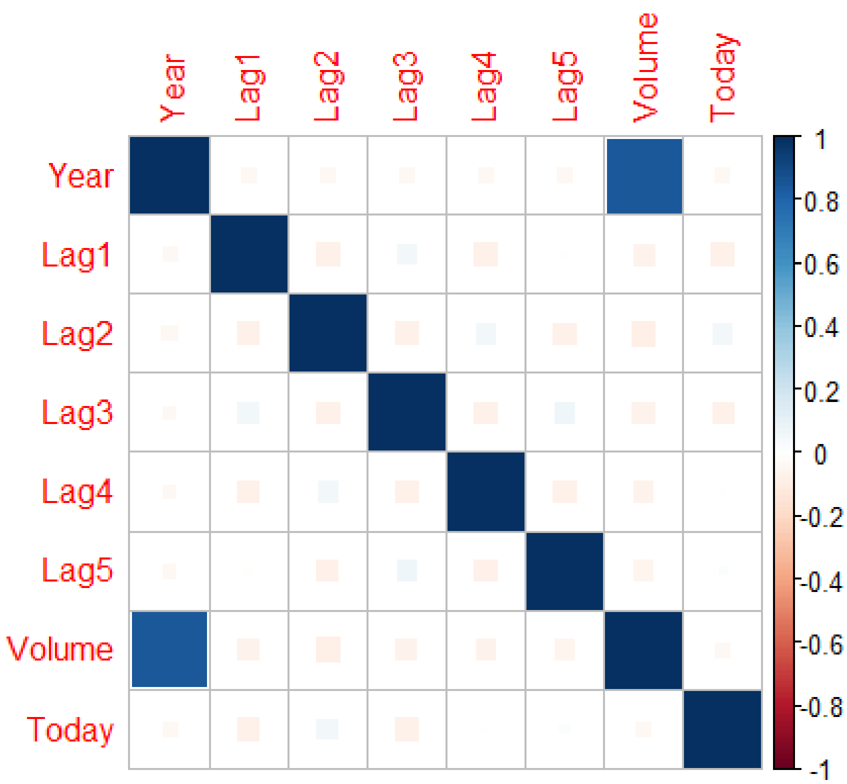

```
pred <- predict(model, test_data)
pred

## $class
## [1] 0 0 0 1
## Levels: 0 1
##
## $posterior
##            0           1
## t1 0.9899599 0.010040066
## t2 0.9982204 0.001779606
## t3 0.9453049 0.054695067
## t4 0.3956527 0.604347342
##
## $x
##          LD1
## t1 -1.2507054
## t2 -1.7243091
## t3 -0.7763265
## t4  0.1154008
```

Question 3. This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?
(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?
(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.
(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).
(e) Repeat (d) using LDA and QDA. Interpret the results.
(f) Which of these methods appears to provide the best results on this data?

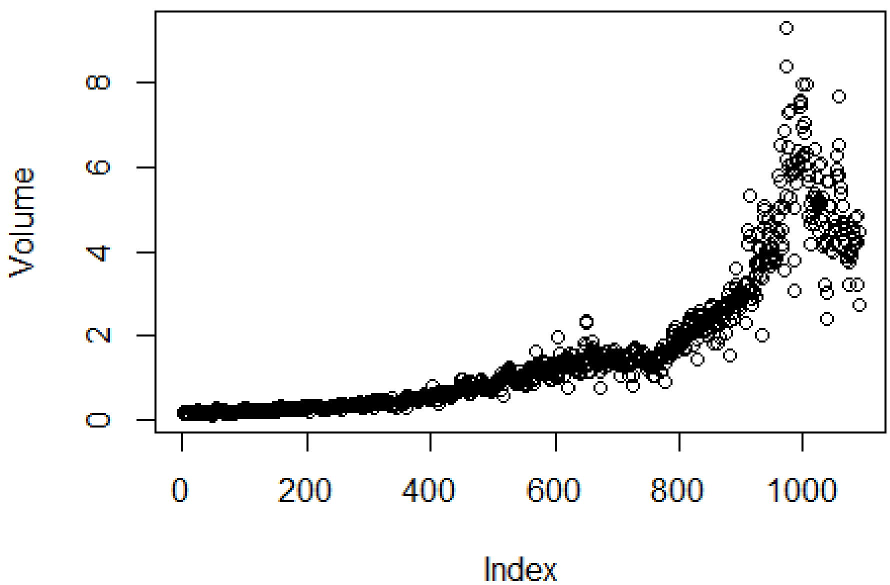
```
library(ISLR)

names(Weekly)

## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"

dim(Weekly)

## [1] 1089    9

summary(Weekly)

##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

```
cor(Weekly[,-9])

##              Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000

corrplot(cor(Weekly[,-9]), method="square")
```



- The correlations between the lag variables and today's return returns are close to zero. There appears to be little or no correlation between today's return and previous days' returns.The only substantial correlation is between Year and volume.

```r
attach(Weekly)
plot(Volume)
```



- By plotting the data, we see that volume is increasing over time.

```r
# fitting Logistic Regression
glm.fit <- glm(Direction ~ Lag1 + Lag2 +Lag3 + Lag4 + Lag5 +Volume, family =
binomial, data = Weekly)
summary(glm.fit)

##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -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
```

- In this case, the p-values for Lag2 and Intercept are statistically significant at the 5% level ($p < 0.05$), while the p-values for Lag1, Lag3, Lag4, Lag5, and Volume are not statistically significant.

```
glm.probs <- predict(glm.fit, type = "response")

# contrasts() is used to specify how the two level of outcome variable should
be coded
contrasts(Direction)

##      Up
## Down  0
## Up    1

threshold <- 0.5

# Convert probabilities to class labels
predictions <- ifelse(glm.probs >= threshold, "Up", "Down")

# Create confusion matrix
confusion_matrix <- table(predictions, Weekly$Direction)

# Print confusion matrix
print(confusion_matrix)

##
## predictions Down  Up
##        Down   54  48
##        Up    430 557

# Fraction of correct prediction
Correctpred <- (54 + 557) / (1089)
print(paste0("Fraction of correct predictions: ", Correctpred))

## [1] "Fraction of correct predictions: 0.561065197428834"
```

- The diagonal elements of the confusion matrix indicate correct predictions while the off diagonal elements represent incorrect predictions. Hence, the model correctly predicted that the market would go up for 557 days and that it would go down for 54 days, giving a total of 611 correct predictions.

- In this case, where we trained and tested the model on the same dataset, we are likely to get overly optimistic estimates of the model's performance. This is because the model has already seen the data it is being tested on, so it is an unfair advantage. The model's accuracy on the entire dataset is 56.1%. However, this is not very informative because it doesn't tell us how well the model would perform on new, unseen data. So to get a better estimate of the model's performance on new data, we split the data into training and testing set. we fit the model on the training set and evaluate its performance on the testing set.

```
# fitting the model using training from 1990-2008 with lag2 as the only
predictor and test from 2009-2010

Train <- Weekly %>% filter(Year <= 2008)
Test <- Weekly %>% filter(Year >= 2009)
glm.fit1 <- glm(Direction ~ Lag2, family = binomial, data = Train)
glm.probs1 <- predict(glm.fit1, Test, type = "response")

# Convert probabilities to class labels
prediction <- ifelse(glm.probs1 >= threshold, "Up", "Down")
# Create confusion matrix
confusionmatrix <- table(prediction, Test$Direction)

# Print confusion matrix
print(confusionmatrix)

##
## prediction Down Up
##       Down    9  5
##       Up     34 56

# Fraction of correct prediction
Correct_pred <- (9 + 56) / (104)
print(paste0("Fraction of correct predictions: ", Correct_pred))

## [1] "Fraction of correct predictions: 0.625"
```

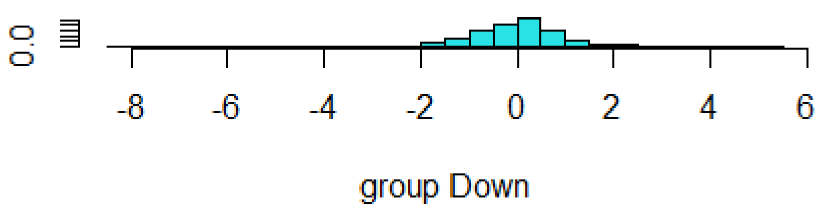
LINEAR DISCRIMINANT ANALYSIS

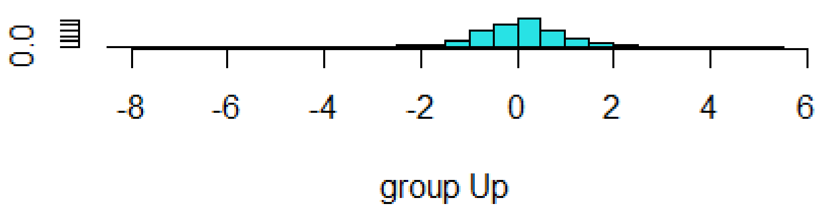
```
lda.fit <- lda(Direction ~ Lag2, data = Train)
lda.fit

## Call:
## lda(Direction ~ Lag2, data = Train)
##
## 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
```

- The Prior probabilities of groups indicate the proportion of observations in each group in the training data. In this case, about 55% of the observations have an Up direction, while 45% have a Down direction.
- The Group means indicate the average value of Lag2 for each group. The Up group has a higher average Lag2 value (0.26) compared to the Down group (-0.04).
- The coefficient of linear discriminant output provides the value 0.44 indicating that Lag2 is positively associated with predicting the Up direction.

```
plot(lda.fit)
```






```
lda.pred <- predict(lda.fit, Test)
lda.class <- lda.pred$class

# Confusion matrix
table(lda.class, Test$Direction)

##
## lda.class Down Up
##      Down    9  5
##      Up     34 56
```

```r
# Fraction of correct prediction
CP <- (9 + 56) / (104)
print(paste0("Fraction of correct predictions: ", CP))

## [1] "Fraction of correct predictions: 0.625"
```

QUADRATIC DISCRIMINANT ANALYSIS

```r
qda.fit <- qda(Direction ~ Lag2, data = Train)
qda.fit

## Call:
## qda(Direction ~ Lag2, data = Train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581

qda.pred <- predict(qda.fit, Test)
qda.class <- qda.pred$class

# Confusion matrix
table(qda.class, Test$Direction)

##
## qda.class Down Up
##      Down    0  0
##      Up     43 61

# Fraction of correct prediction
cp <- (0 + 61) / (104)
print(paste0("Fraction of correct predictions: ", cp))

## [1] "Fraction of correct predictions: 0.586538461538462"
```

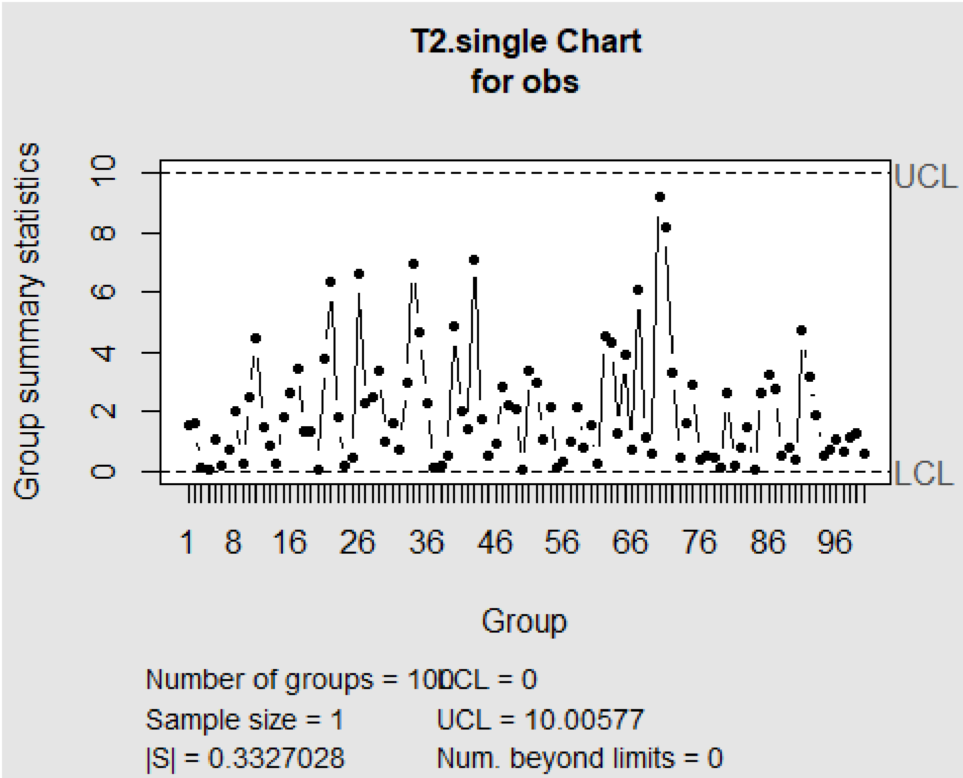
- From the result above, the Logistic model and Linear Discriminant Analysis model outperform the Quadratic Discriminant Analysis model in terms of their accuracy. The Logistic model and Linear Discriminant Analysis model both have similar accuracies of 62.5% which is greater than the Quadratic Discriminant Analysis model which has an accuracy of 58.65%

QUESTION 4: Construct the Hotelling T 2 charts for future observations using the a simulated data Simulation Set-up

a) Use the set.seed(”6559)

b) Generate 100 observations from bivariate normal distribution with μ = (2, 5) and covariance matrix - var(x1)=1; var(x2)=.5, cov(x1,x2)=0.3.
c) Estimate the classical estimators of mean and covariances
d) Generate 25 future observations, using bivariate normal distribution with μ = (2, 5) and covariance matrix - var(x1)=1; var(x2)=.5, cov(x1,x2)=0.3.
e) Draw three T 2 control chart for future observation using classical estimator and robust estimators of mean and covariance matrix. Draw your conclusions. g) Generate another 25 future observations, using bivariate normal distribution with μ = (2.4, 6) and covariance matrix - var(x1)=1; var(x2)=.5, cov(x1,x2)=0.3. and repeat (e).
f) Offer your comments. Compare your results with univariate charts for individual observations.

```
set.seed(6559)

sigma <- matrix(c(1, 0.3, 0.3, 0.5), nrow = 2, ncol = 2)

# generate 100 observations
obs <- mvrnorm(n = 100, mu = c(2,5), Sigma = sigma)
head(obs)

##           [,1]     [,2]
## [1,] 3.2282466 5.244019
## [2,] 0.8246453 4.847112
## [3,] 2.2906361 5.225665
## [4,] 2.0452035 5.138790
## [5,] 1.5147996 5.355329
## [6,] 1.8772734 4.746011

q1 <- mqcc(obs, type = "T2.single", confidence.level = (1 - 0.0027)^2)
```




```
summary(q1)

##
## Call:
## mqcc(data = obs, type = "T2.single", confidence.level = (1 -
0.0027)^2)
##
## T2.single chart for obs
##
## Summary of group statistics:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.020035 0.543787 1.420222 1.980000 2.753418 9.178539
##
## Number of variables:  2
## Number of groups:  100
## Group sample size:  1
##
## Center:
##       V1       V2
## 2.028114 5.027899
##
## Covariance matrix:
##           V1        V2
## V1 1.0293141 0.3524414
## V2 0.3524414 0.4439050
## |S|:  0.3327028
##
```

```
## Control limits:
##   LCL      UCL
##     0 10.00577
```

```
# classical estimators of mean and covariances
class_mean <- q1$center
class_mean
```

```
##       V1       V2
## 2.028114 5.027899
```

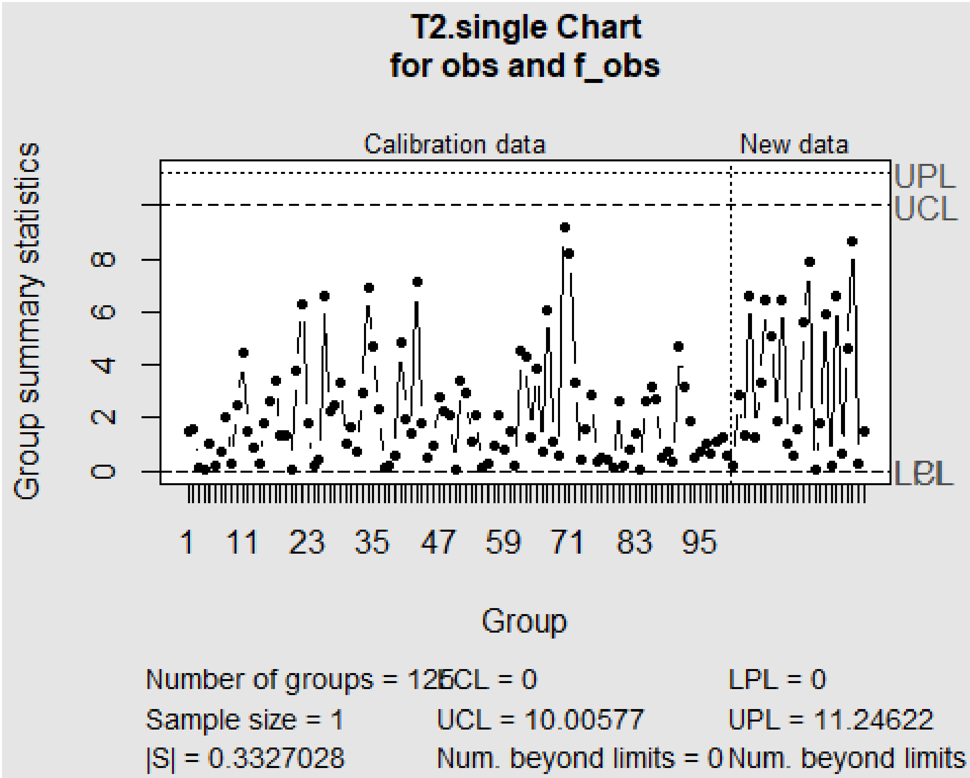
```
class_cov <- q1$cov
class_cov
```

```
##           V1        V2
## V1 1.0293141 0.3524414
## V2 0.3524414 0.4439050
```

```
# first 25 future observations
f_obs <- mvrnorm(n = 25, mu = c(2,5), Sigma = sigma)
f_obs
```

```
##              [,1]     [,2]
##  [1,]  2.49459807 5.105514
##  [2,]  3.37172120 4.895565
##  [3,]  1.52056474 5.441217
##  [4,]  0.44763081 5.651829
##  [5,]  1.29044758 4.273317
##  [6,]  2.49444071 4.186979
##  [7,] -0.07751139 5.140148
##  [8,]  0.87364506 5.741826
##  [9,]  1.93671789 4.215154
## [10,] -0.55367721 4.149838
## [11,]  2.95646967 5.074981
## [12,]  1.35781685 5.005896
## [13,]  3.23063347 5.183684
## [14,]  2.17826263 6.421456
## [15,] -0.30044607 5.146776
## [16,]  2.18925186 4.968234
## [17,]  2.10165085 5.815150
## [18,] -0.16423246 4.918217
## [19,]  1.66315792 5.079373
## [20,]  2.22281729 6.550927
## [21,]  1.53838520 5.229286
## [22,]  3.22041104 6.454068
## [23,] -0.95563927 4.136426
## [24,]  1.54430347 4.994265
## [25,]  3.15049309 5.694374
```

```
qmf = mqcc(obs, type = "T2.single", center = class_mean, cov = class_cov,
pred.limits = TRUE, newdata = f_obs, confidence.level = (1 - 0.0027)^2)
```



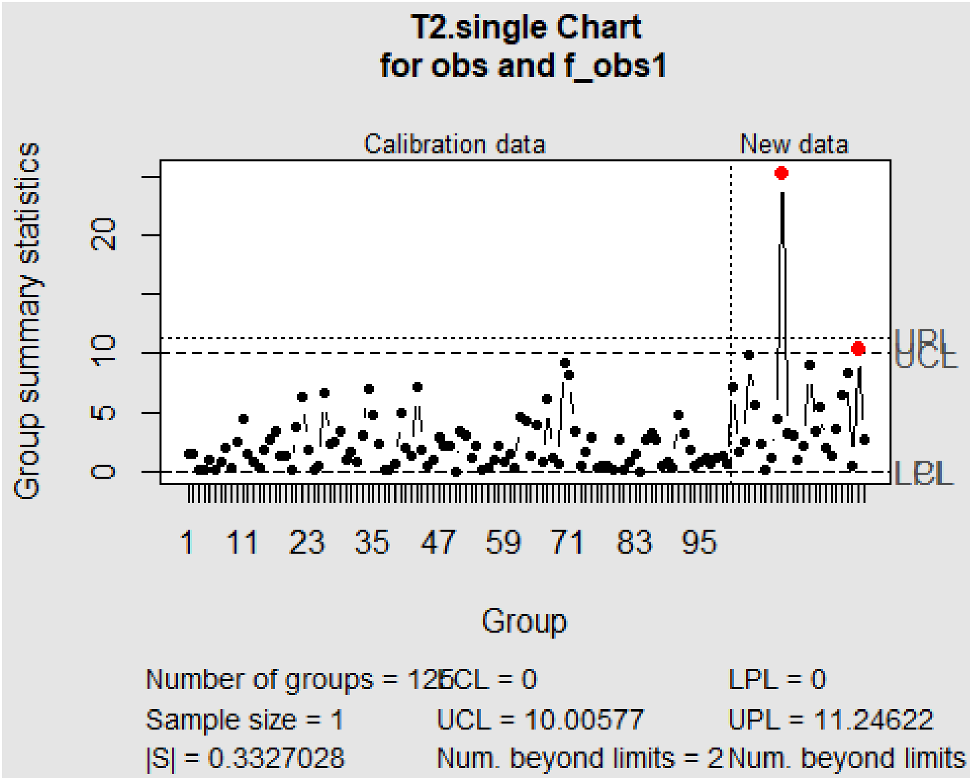

```
summary(qmf)

##
## Call:
## mqcc(data = obs, type = "T2.single", center = class_mean, cov = class_cov,
pred.limits = TRUE, newdata = f_obs, confidence.level = (1 -
0.0027)^2)
##
## T2.single chart for obs
##
## Summary of group statistics:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.020035 0.543787 1.420222 1.980000 2.753418 9.178539
##
## Number of variables:  2
## Number of groups:  100
## Group sample size:  1
##
## Center:
##        V1        V2
## 2.028114 5.027899
##
## Covariance matrix:
##            V1        V2
## V1 1.0293141 0.3524414
## V2 0.3524414 0.4439050
## |S|:  0.3327028
```

```
##
## Summary of group statistics in f_obs:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.066027 1.064161 1.897387 3.303619 5.940627 8.701665
##
## Number of groups:  25
## Group sample size:  1
##
## Control limits:
##  LCL      UCL
##    0 10.00577
##
## Prediction limits:
##  LPL      UPL
##    0 11.24622
```

```
# second 25 future observations
f_obs1 <- mvrnorm(n = 25, mu = c(2.4,6), Sigma = sigma)
f_obs1
```

```
##           [,1]     [,2]
##  [1,] 3.307050 6.815931
##  [2,] 1.732756 5.626937
##  [3,] 3.552650 5.821028
##  [4,] 4.932011 6.756723
##  [5,] 4.183412 6.359303
##  [6,] 3.586077 5.670151
##  [7,] 1.770237 5.107920
##  [8,] 2.871253 5.676942
##  [9,] 3.240412 6.423278
## [10,] 2.108495 7.914922
## [11,] 2.431065 6.145469
## [12,] 2.053433 6.024066
## [13,] 1.562588 5.351807
## [14,] 2.209174 5.931489
## [15,] 4.684600 6.767460
## [16,] 2.470257 6.193990
## [17,] 4.123836 6.372095
## [18,] 2.671862 5.949128
## [19,] 2.380706 5.768234
## [20,] 2.356018 6.183119
## [21,] 3.622273 6.701503
## [22,] 4.068034 6.896960
## [23,] 2.240925 5.497583
## [24,] 3.994280 7.168850
## [25,] 2.255869 6.030680
```

```
qmf1 = mqcc(obs, type = "T2.single", center = class_mean, cov = class_cov,
pred.limits = TRUE, newdata = f_obs1, confidence.level = (1 - 0.0027)^2)
```




```
summary(qmf1)

##
## Call:
## mqcc(data = obs, type = "T2.single", center = class_mean, cov = class_cov,
pred.limits = TRUE, newdata = f_obs1, confidence.level = (1 -
0.0027)^2)
##
## T2.single chart for obs
##
## Summary of group statistics:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.020035 0.543787 1.420222 1.980000 2.753418 9.178539
##
## Number of variables:  2
## Number of groups:  100
## Group sample size:  1
##
## Center:
##       V1       V2
## 2.028114 5.027899
##
## Covariance matrix:
##           V1        V2
## V1 1.0293141 0.3524414
## V2 0.3524414 0.4439050
## |S|:  0.3327028
```

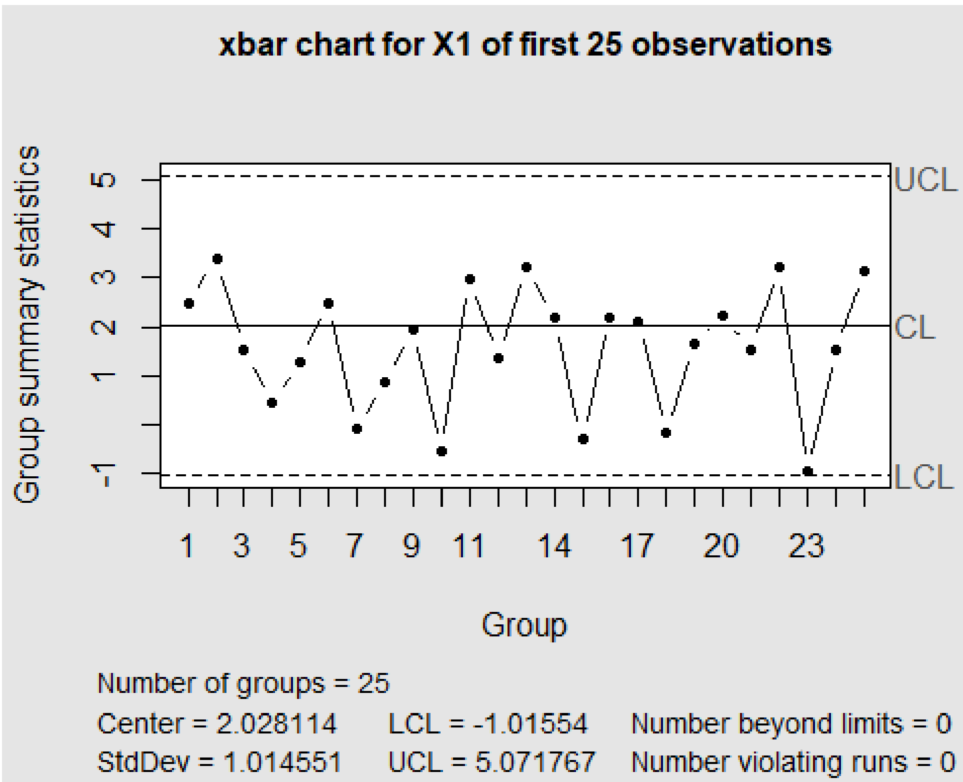
```
##
## Summary of group statistics in f_obs1:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.152257  1.922065  3.126586  4.892074  6.403772 25.303433
##
## Number of groups:  25
## Group sample size:  1
##
## Control limits:
##  LCL      UCL
##    0 10.00577
##
## Prediction limits:
##  LPL      UPL
##    0 11.24622
```

- The Upper prediction limit (UPL) is used to detect when a process is starting to produce data points that are beyond what was predicted, which could indicate a shift in the process.
- Points that are above the UPL (Upper Prediction Limit) are called "outliers" or "out-of-control points". These are data points that are outside of the expected range and may indicate that the process being monitored is out of control and needs to be investigated.
- For first future observations, we observe that no points exceeds the control limits, so for original data, we would conclude that the process is statistically in control.
- Meanwhile, for the second future observations, observe that two points exceed the control limits, so for the original data, we would conclude that the process is not statistically in control.Engineers must take a special look at these points in order to identify and assign causes attributed to changes in the system that led the process to be out of control.

UNIVARIAE CHARTS FOR INDIVIDUAL OBSERVATIONS

```
x1.obs <- obs[,1]
x2.obs <- obs[,2]

# variable means for individual variables of the 100 observations
x1.obs_mean <- mean(x1.obs)
x2.obs_mean <- mean(x2.obs)

# variable standard deviation for individual variables of the 100
observations
x1.std <- sd(x1.obs)
x2.std <- sd(x2.obs)

# setting control limits for X1 variable
UCL1 <- x1.obs_mean + (3 * x1.std)
LCL1<- x1.obs_mean - (3 * x1.std)
```

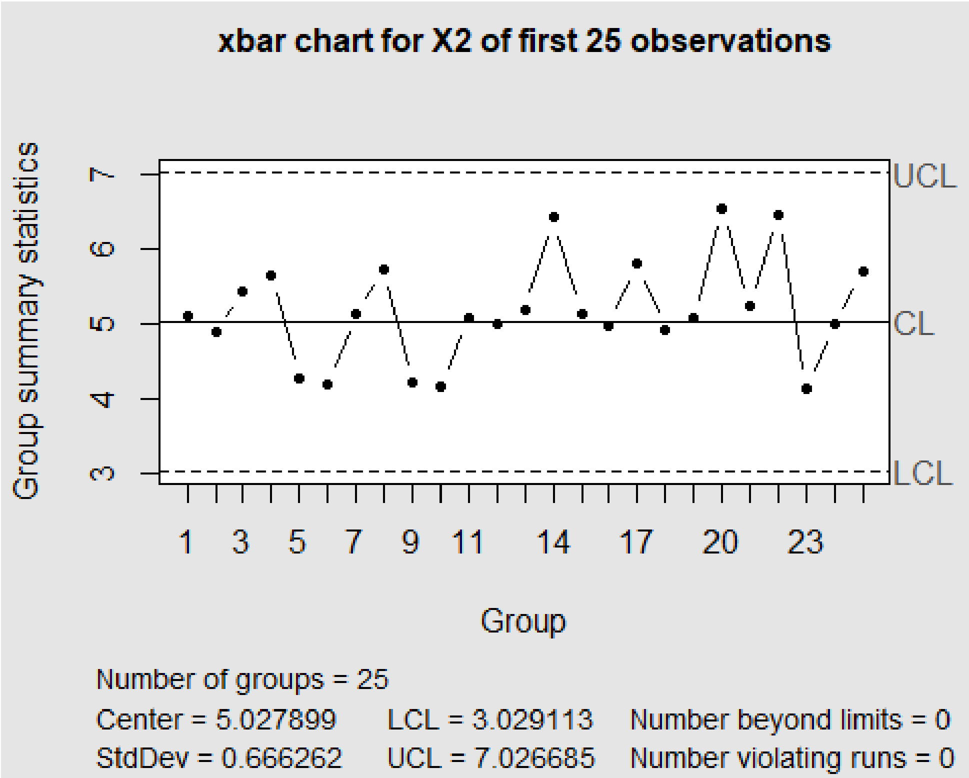
```r
# setting control limits for X2 variable
UCL2 <- x2.obs_mean + (3 * x2.std)
LCL2<- x2.obs_mean - (3 * x2.std)

# univariate charts for first future observation
x1.f <- f_obs[,1]
x2.f <- f_obs[,2]

# Create the control chart
qcc(x1.f, type="xbar", center = x1.obs_mean , std.dev = x1.std, limits =
c(LCL1, UCL1), nsigmas = 3, title = 'xbar chart for X1 of first 25
observations')
```




```
## List of 11
##  $ call      : language qcc(data = x1.f, type = "xbar", center =
x1.obs_mean, std.dev = x1.std,      limits = c(LCL1, UCL1), nsigmas = 3,|
__truncated__
##  $ type      : chr "xbar"
##  $ data.name : chr "x1.f"
##  $ data      : num [1:25, 1] 2.495 3.372 1.521 0.448 1.29 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 2.495 3.372 1.521 0.448 1.29 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 1 1 1 1 1 1 1 1 1 1 ...
##  $ center    : num 2.03
##  $ std.dev   : num 1.01
##  $ nsigmas   : num 3
```

```
##  $ limits    : num [1, 1:2] -1.02 5.07
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

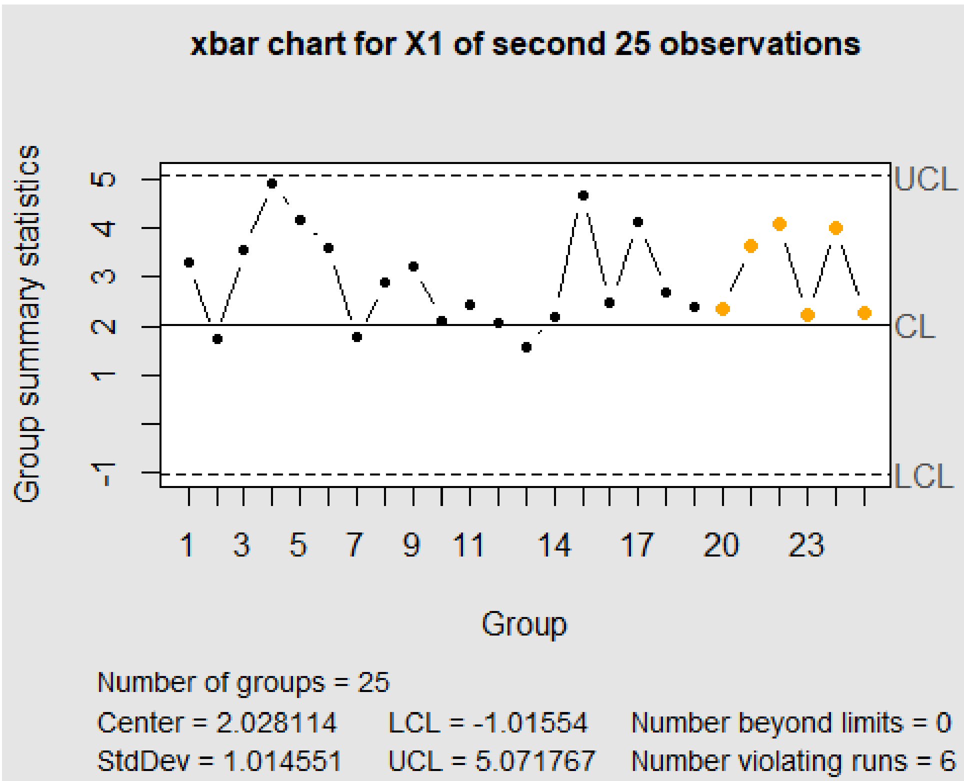
```
qcc(x2.f, type="xbar", center = x2.obs_mean , std.dev = x2.std, limits =
c(LCL2, UCL2), nsigmas = 3, title = 'xbar chart for X2 of first 25
observations')
```




```
## List of 11
##  $ call      : language qcc(data = x2.f, type = "xbar", center =
x2.obs_mean, std.dev = x2.std,      limits = c(LCL2, UCL2), nsigmas = 3,|
__truncated__
##  $ type      : chr "xbar"
##  $ data.name : chr "x2.f"
##  $ data      : num [1:25, 1] 5.11 4.9 5.44 5.65 4.27 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 5.11 4.9 5.44 5.65 4.27 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 1 1 1 1 1 1 1 1 1 1 ...
##  $ center    : num 5.03
##  $ std.dev   : num 0.666
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 3.03 7.03
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

- Univariate chart for the individual observations of the first future observations indicate that there are no violating runs or points beyond limits.

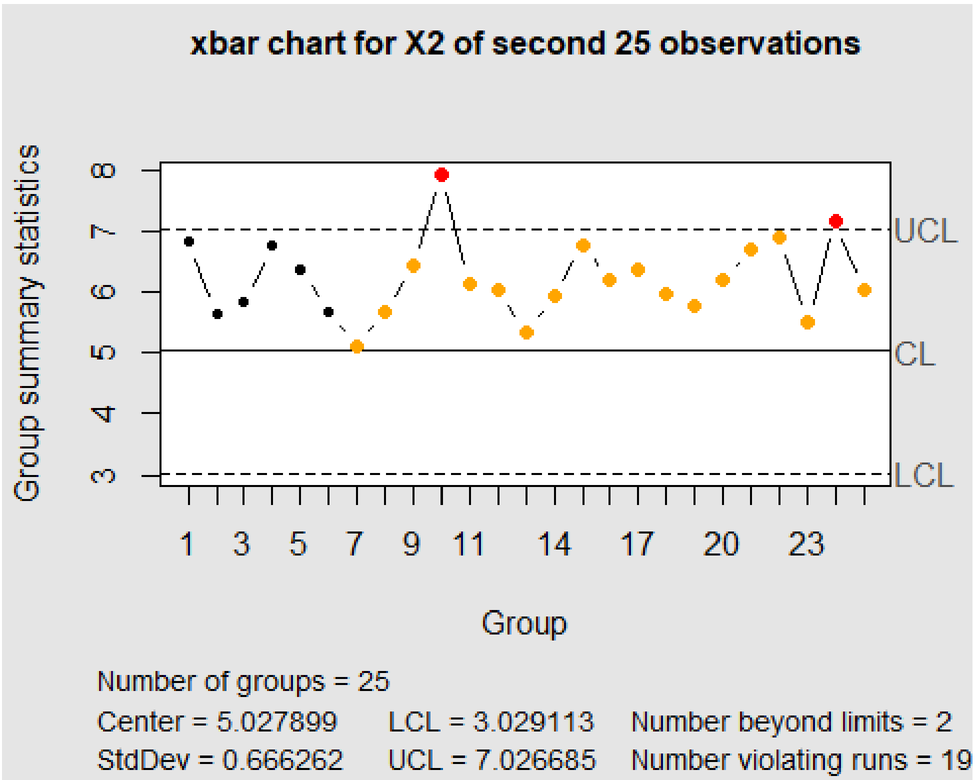
```
# univariate charts for second future observation
x1.f1 <- f_obs1[,1]
x2.f1 <- f_obs1[,2]

# Create the control chart
qcc(x1.f1, type="xbar", center = x1.obs_mean , std.dev = x1.std, limits =
c(LCL1, UCL1), nsigmas = 3, title = 'xbar chart for X1 of second 25
observations')
```




```
## List of 11
##  $ call      : language qcc(data = x1.f1, type = "xbar", center =
x1.obs_mean, std.dev = x1.std,     limits = c(LCL1, UCL1), nsigmas = 3|
__truncated__
##  $ type      : chr "xbar"
##  $ data.name : chr "x1.f1"
##  $ data      : num [1:25, 1] 3.31 1.73 3.55 4.93 4.18 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 3.31 1.73 3.55 4.93 4.18 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 1 1 1 1 1 1 1 1 1 1 ...
##  $ center    : num 2.03
##  $ std.dev   : num 1.01
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] -1.02 5.07
##   ..- attr(*, "dimnames")=List of 2
```

```
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
qcc(x2.f1, type="xbar", center = x2.obs_mean , std.dev = x2.std, limits =
c(LCL2, UCL2), nsigmas = 3, title = 'xbar chart for X2 of second 25
observations')
```




```
## List of 11
##  $ call      : language qcc(data = x2.f1, type = "xbar", center =
x2.obs_mean, std.dev = x2.std,      limits = c(LCL2, UCL2), nsigmas = 3|
__truncated__
##  $ type      : chr "xbar"
##  $ data.name : chr "x2.f1"
##  $ data      : num [1:25, 1] 6.82 5.63 5.82 6.76 6.36 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 6.82 5.63 5.82 6.76 6.36 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 1 1 1 1 1 1 1 1 1 1 ...
##  $ center    : num 5.03
##  $ std.dev   : num 0.666
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 3.03 7.03
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

- Univariate chart for the individual observations of the second future observations indicate that there are a couple of violating runs and points beyond limits. Violating runs refer to a sequence of points that are not randomly distributed within the control limits, indicating a pattern of non-random variation in the process. Points outside the control limit indicate that the process output is outside the expected range of variation and that there may be a significant source of variation in the process that needs to be investigated.

QUESTION 5: Refer the class note on discriminant analysis and definition notations of w, B&S. Show that the w maximizing.

$$\frac{w^T B_w w}{w^T S_w w}$$

satisfies

$$S_w^{-1} B_w w = \lambda w$$

(Hence, w is the eigenvector and λ is eigenvalue of S−1B.) Hint: Argue that we can maximize wT Bw subject to wT Sw = a , where a is a constant. Then introduce a Lagrange multiplier for the constraint and differentiate with respect to elements of w

Discriminant analysis involves finding a linear combination of variables that best separates two or more groups. In this context, Fisher's Linear Discriminant Analysis (LDA) is a common technique that involves finding a linear combination of the input features to maximize the between-class distance while minimizing the within-class distance.

Let's consider a dataset with n observations and p input features, where each observation belongs to one of k classes. The within-class scatter matrix S_w and the between-class scatter matrix B_w are defined as:

$$S_w = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)\,(x - \mu_i)^T$$

and

$$B_w = \sum_{i=1}^{k} n_i\,(\mu_i - \mu)(\mu_i - \mu)^T$$

where C_i is the set of observations in class i, μ_i is the mean vector of the observations in class i, μ is the overall mean vector, and n_i is the number of observations in class i.

Now we want to find the weight vector w that maximizes the ratio of between-class scatter to within-class scatter, which can be written as:

$$\frac{w^T B_w w}{w^T S_w w}$$

To maximize this ratio subject to the constraint $w^T S_w w = 1$, we can introduce a Lagrange multiplier λ and from the Lagrangian function L:

$$L(w, \lambda) = w^T B_w w - \lambda(w^T S_w w - 1)$$

Taking the derivative of L with respect to w and setting it to zero, we get:

$$\nabla_w L(w, \lambda) = 2B_w w - 2\lambda S_w w = 0$$

Multiplying both sides by S_w^{-1}, we get:

$$S_w^{-1} B_w w = \lambda w$$

This equation shows that w is an eigenvector of $S_w^{-1} B_w$ with eigenvalue λ. Therefore, to maximize $\frac{w^T B_w w}{w^T S_w w}$, we need to find the eigenvector w that corresponds to the largest eigenvalue of $S_w^{-1} B_w$.