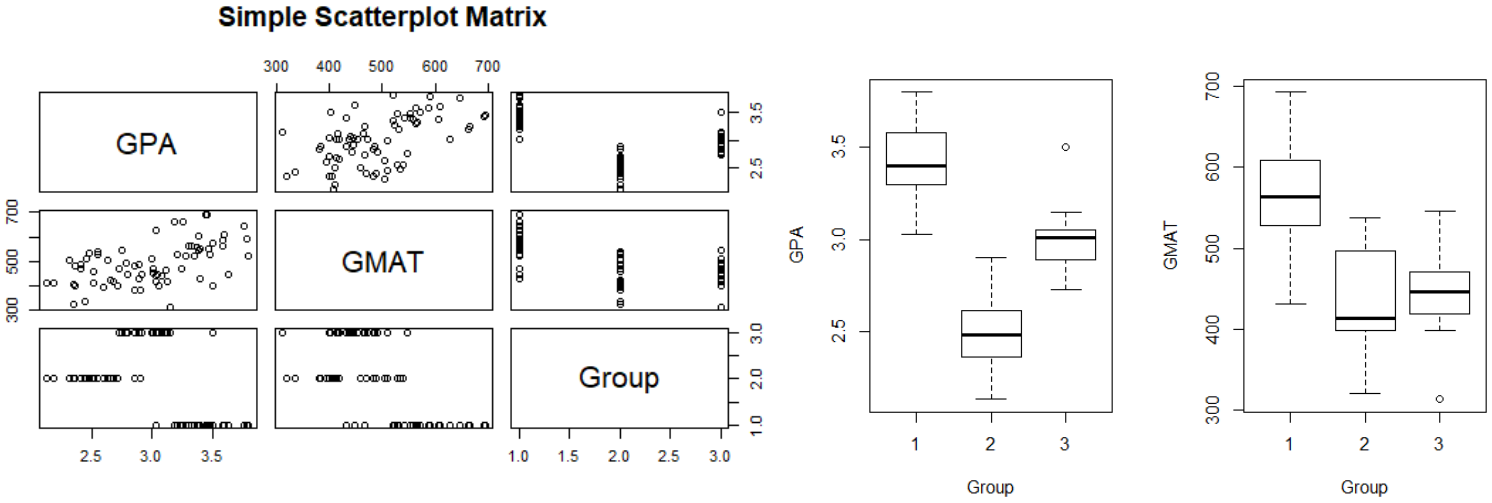
**SECTION 1** **Problem 2**

a) Following is an outline of the training data set.

```
'data.frame':70 obs. of  3 variables:
 $ GPA  : num  3.46 3.03 3.19 3.63 3.59 3.3 3.4 3.5 3.78 3.44 ...
 $ GMAT : int  693 626 663 447 588 563 553 572 591 692 ...
 $ Group: int  1 1 1 1 1 1 1 1 1 1 ...
```

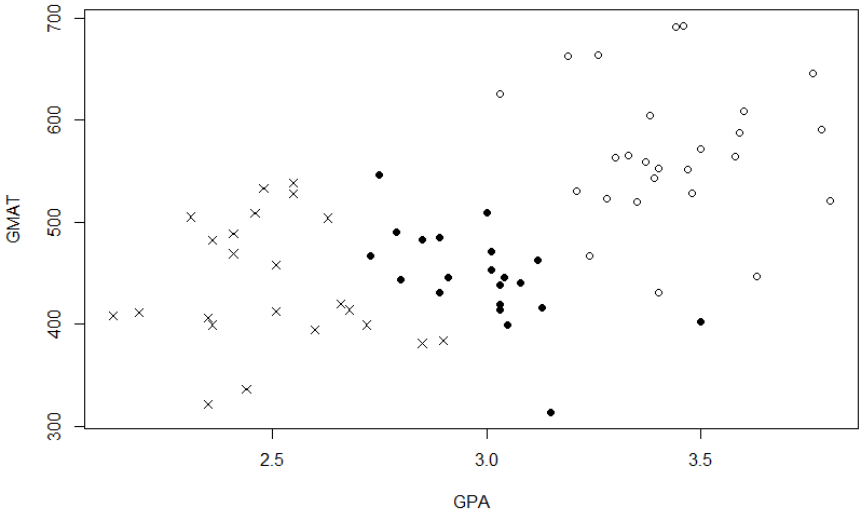
There are 70 observation with 3 variables; GPA, GMAT been numerical and Group is categorical. Below shows a summery of the numerical variables.

```
      GPA             GMAT
 Min.   :2.130   Min.   :313.0
 1st Qu.:2.638   1st Qu.:419.2
 Median :3.030   Median :482.5
 Mean   :2.992   Mean   :489.9
 3rd Qu.:3.365   3rd Qu.:545.2
 Max.   :3.800   Max.   :693.0
 NA's   :170     NA's   :170
```



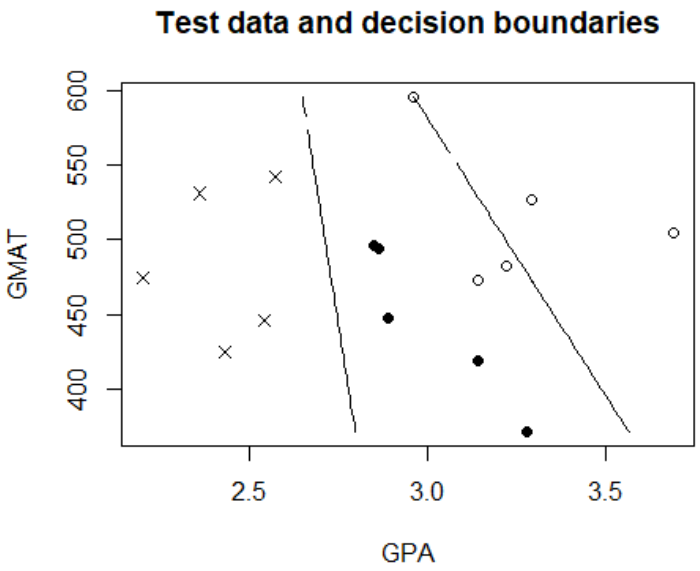

From the scatterplot, there seems to be a relation between GPA and GMAT since the data points on the plots shows a pattern.

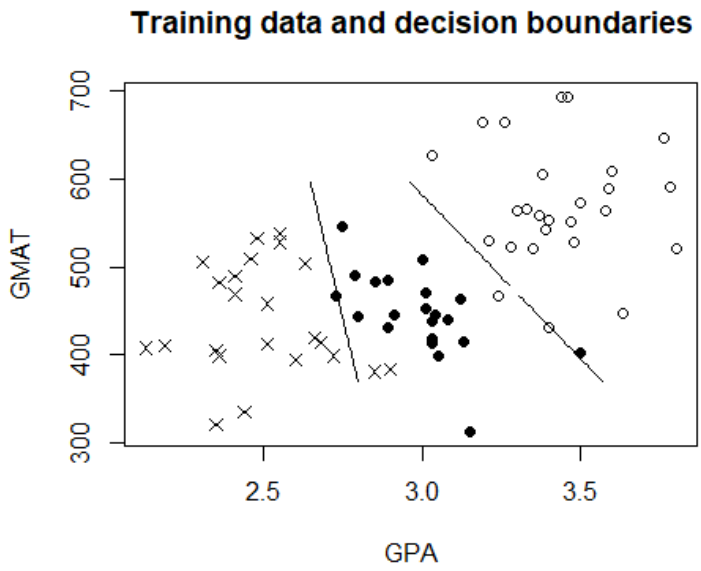
On the other hand, the box plots clearly shows how each group distributed differently in GPA and GMAT. Thus, these groups can be easily classify. As following graph shows, visually we can see a clear separation between these groups.




The correlation between the two variables was found to be **0.5282156,** which confirms that there is a moderately strong linear relationship between the two variables.

b) Decision boundaries, from fitting LDA using training data, was plotted with the test data and training data separately as follows.



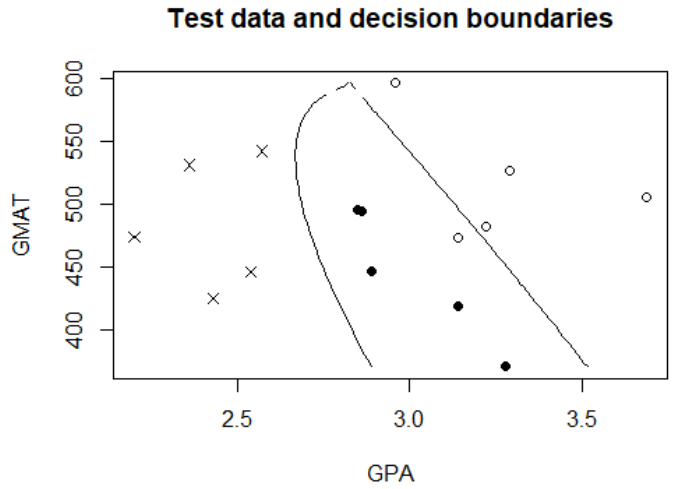



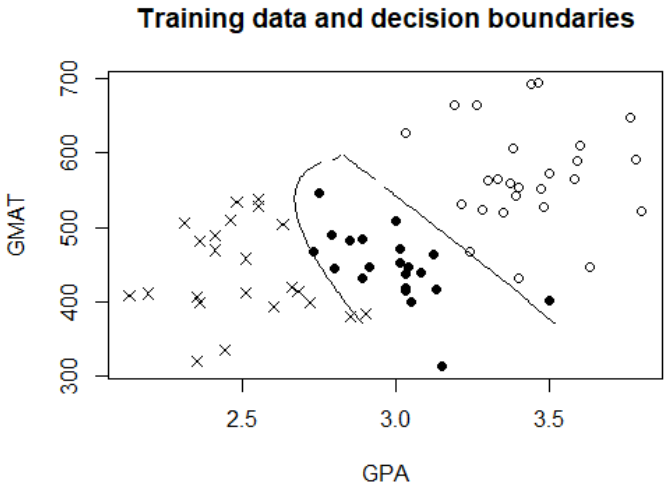

Yes, the decision boundary seems sensible. Confusion matrix for test and training data shows bellow. Misclassification rate for test data and training data also calculated from these matrices.

```
> table(LDA.pred$class, test.y)
   test.y
    1 2 3
  1 2 0 0
  2 0 5 0
  3 3 0 5
> table(LDA.pred_train$class, train.y)
   train.y
     1  2  3
  1 24  0  1
  2  0 21  1
  3  2  2 19

> mc.rate_test = 3/15
> mc.rate_train = 6/70

> mc.rate_test
[1] 0.2
> mc.rate_train
[1] 0.08571429
```

We observe that, as expected, training error rate is higher than test error rate. This might be because we used only 15 observations as test data.

c) Decision boundaries, from fitting QDA using training data, was plotted with the test data and training data separately as follows.




Again, the decision boundary seems sensible. Confusion matrix for test and training data shows bellow. Misclassification rate for test data and training data also calculated from these matrices.

```
> table(QDA.pred$class, test.y)
   test.y
    1 2 3
  1 4 0 0
  2 0 5 0
  3 1 0 5
> table(QDA.pred_train$class, train.y)
   train.y
     1  2  3
  1 26  0  1
  2  0 22  0
  3  0  1 20
> mc.rate_test = 1/15
> mc.rate_train = 2/70
> mc.rate_test
[1] 0.06666667
> mc.rate_train
[1] 0.02857143
```

We see that both training and test error rates are lower. Also, It is seen that the training and test error rates of QDA is better than LDA.

**SECTION 2** **Problem 2**

```
# Importing data file
admission.data <- read.csv(file.choose(), header = T)
attach(admission.data)


admit.data <- subset(admission.data, Group == 1)
reject.data <- subset(admission.data, Group == 2)
borderline.data <- subset(admission.data, Group == 3)

dim(admit.data)
dim(reject.data)
dim(borderline.data)

# Separating into train and test data
train.data <- rbind(admit.data[6:31,],reject.data[6:28,],borderline.data[6:26,])
train.X <- cbind(GPA=train.data$GPA, GMAT=train.data$GMAT)
train.y <- train.data$Group

test.data <- rbind(admit.data[1:5,],reject.data[1:5,],borderline.data[1:5,])
test.x <- cbind(GPA=test.data$GPA, GMAT=test.data$GMAT)
test.y <- test.data$Group
test.x


# a)
# Visualizing the structure of the dataset
str(train.data)
summary(train.data[,-3])

# Scatterplot of the variables
pairs(~. , data=train.data, main="Simple Scatterplot Matrix")

plot(train.X, xlab = "GPA", ylab = "GMAT",
     pch = ifelse(train.y == 1, 1, ifelse(train.y == 2, 4, 16)))

# Correlation between GPA and GMAT
cor(train.X[, 1], train.X[, 2])

par(mfrow = c(1, 2))
boxplot(train.X[, 1] ~ train.y, ylab = "GPA", xlab = "Group")
boxplot(train.X[, 2] ~ train.y, ylab = "GMAT", xlab = "Group")
par(mfrow = c(1, 1))


# b)

library(MASS) # To call lda and qda fucntions
# Fitting lda for the training data
LDA.fit <- lda(Group ~ GPA + GMAT, data = train.data)

# Predictions for test and training observations
LDA.pred <- predict(LDA.fit, test.data)
LDA.pred_train=predict(LDA.fit, train.data)
```

```
# Decision boundary (using the "blind" contour approach; test data)
# Set up a dense grid and compute posterior prob on the grid
n.grid <- 50
x1.grid <- seq(f = min(test.x[, 1]), t = max(test.x[, 1]), l = n.grid)
x2.grid <- seq(f = min(test.x[, 2]), t = max(test.x[, 2]), l = n.grid)
grid <- expand.grid(x1.grid, x2.grid)
colnames(grid) <- colnames(test.x)

# Predictions for grid
pred.grid <- predict(LDA.fit, grid)

prob1 <- matrix(c(pred.grid$posterior[,1] - pmax(pred.grid$posterior[,2],
                  pred.grid$posterior[,3])), nrow = n.grid, ncol = n.grid)
prob2 <- matrix(c(pred.grid$posterior[,2] - pmax(pred.grid$posterior[,1],
                  pred.grid$posterior[,3])), nrow = n.grid, ncol = n.grid)

# Plotting test data and decision boundaries
par(mfrow = c(1, 2))
plot(test.x, xlab = "GPA", ylab = "GMAT", main = "Test data and decision boundaries",
     pch = ifelse(test.y == 1, 1, ifelse(test.y == 2, 4, 16)))
contour(x1.grid, x2.grid, prob1, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)
contour(x1.grid, x2.grid, prob2, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)

plot(train.X, xlab = "GPA", ylab = "GMAT", main = "Training data and decision boundaries",
     pch = ifelse(train.y == 1, 1, ifelse(train.y == 2, 4, 16)))
contour(x1.grid, x2.grid, prob1, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)
contour(x1.grid, x2.grid, prob2, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)
par(mfrow = c(1, 1))

# Confusion matrix for test and training data
table(LDA.pred$class, test.y)
table(LDA.pred_train$class, train.y)

# Misclassification rates
mc.rate_test = 3/15
mc.rate_train = 6/70


# c)
# Fitting qda for the training data
QDA.fit <- qda(Group ~ GPA + GMAT, data = train.data)

# Predictions for test and training observations
QDA.pred <- predict(QDA.fit, test.data)
QDA.pred_train=predict(QDA.fit, train.data)

# Predictions for grid
pred.grid <- predict(QDA.fit, grid)

prob1 <- matrix(c(pred.grid$posterior[,1] - pmax(pred.grid$posterior[,2],
                  pred.grid$posterior[,3])), nrow = n.grid, ncol = n.grid)
prob2 <- matrix(c(pred.grid$posterior[,2] - pmax(pred.grid$posterior[,1],
```

```
                pred.grid$posterior[,3])), nrow = n.grid, ncol = n.grid)


# Plotting test data and decision boundaries
par(mfrow = c(1, 2))
plot(test.x, xlab = "GPA", ylab = "GMAT", main = "Test data and decision boundaries",
     pch = ifelse(test.y == 1, 1, ifelse(test.y == 2, 4, 16)))
contour(x1.grid, x2.grid, prob1, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)
contour(x1.grid, x2.grid, prob2, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)

plot(train.X, xlab = "GPA", ylab = "GMAT", main = "Training data and decision boundaries",
     pch = ifelse(train.y == 1, 1, ifelse(train.y == 2, 4, 16)))
contour(x1.grid, x2.grid, prob1, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)
contour(x1.grid, x2.grid, prob2, levels = 0, labels = "", xlab = "", ylab = "",
        main = "", add = T)
par(mfrow = c(1, 1))

# Confusion matrix for test and training data
table(QDA.pred$class, test.y)
table(QDA.pred_train$class, train.y)

# Misclassification rates
mc.rate_test = 1/15
mc.rate_train = 2/70
```