Project 02

Indrajith Wasala Mudiyanselage

Question 01

a) The pairwise scatter plot visually reveals that the suggested response variable (Quality) has a moderately positive linear relationship with the variables Aroma, Body, and flavor and it does not show any linear relationship between Clarity and Oakiness. Also, note that variable Flavor has a positive linear relationship between Aroma and Body. The correlation matrix confirms the above information.Region 3 tends to have high Quality values while Region 2 gets low Quality values.

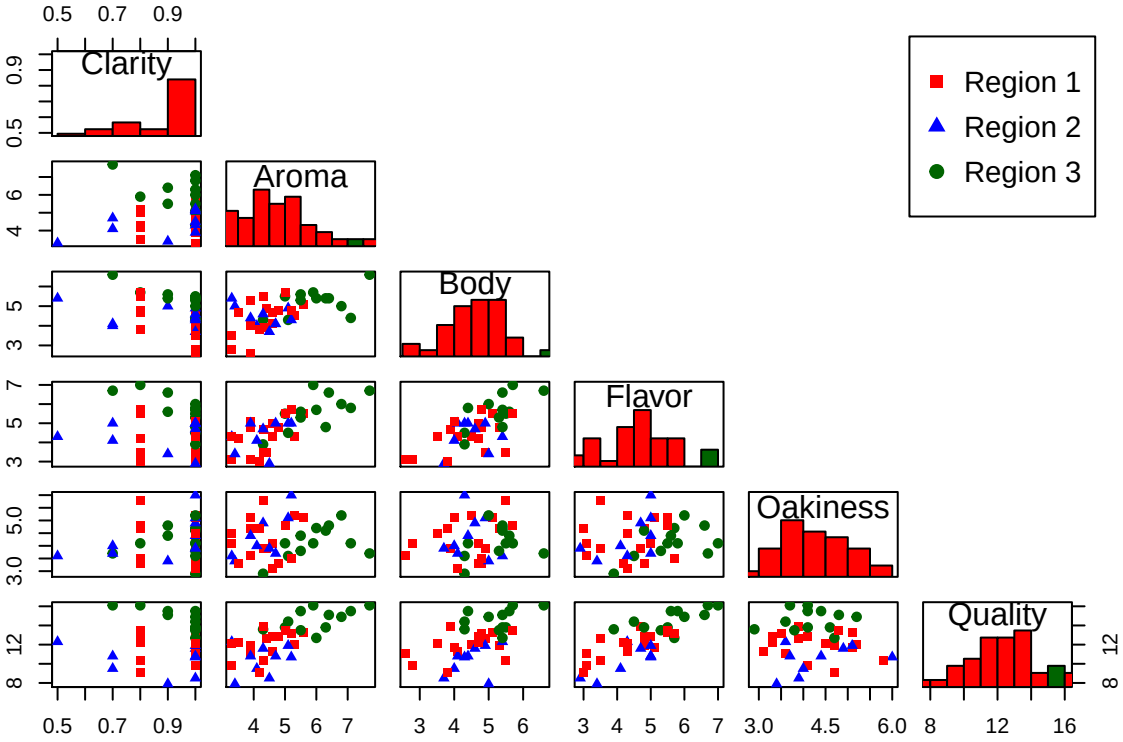

Figure 1: Pairwise scatterplots

	Clarity	Aroma	Body	Flavor	Oakiness	Quality
Clarity	1.00	0.06	-0.31	-0.09	0.18	0.03
Aroma	0.06	1.00	0.55	0.74	0.20	0.71
Body	-0.31	0.55	1.00	0.65	0.15	0.55
Flavor	-0.09	0.74	0.65	1.00	0.18	0.79
Oakiness	0.18	0.20	0.15	0.18	1.00	-0.05
Quality	0.03	0.71	0.55	0.79	-0.05	1.00

Table 1: Correlation Matrix

b) The above explanatory analysis suggests that Quality have linear relationships between most of the given predictor variables.Also, note that the response variable (Quality) is not far out of normal distribution.Therefore Check Linear regression model assumption.

Normality of error term: According to the Shapiro-Wilk test, the error terms (residuals) are normally distributed (p-value = 0.8993 . 0.05). According to the Q-Q plot of figure 2, the points are roughly forming a straight line; this also indicates residuals are approximately normal.

Constant Variance (Heteroscedasticity): According to the Breusch-Pagan test, the variance of residual is the same (p-value =0.8993 > 0.05). Residuals Vs. Fitted plot in figure 2 confirms that variances of the error terms are equal as residuals roughly form a horizontal band around the 0 line.

Independence of error term : The Durbin-Watson test suggests that the error terms are independent (No autocorrelation). But note that the p-value is small (p-value =0.088>0.05). In the Scale-location plot (figure 2), the red trend line is approximately horizontal. So that we can assume the independent error term assumption is not violated.

Therefore Quality is appropriate as a response variable and a transformation is not needed.

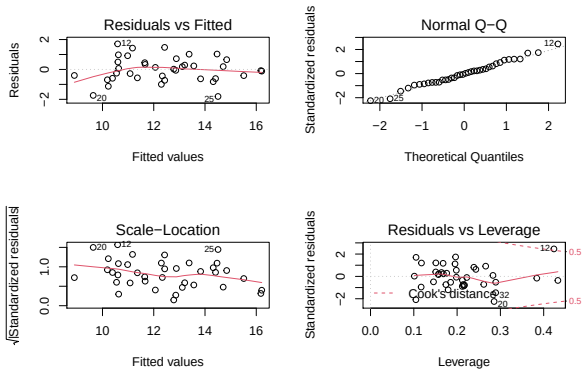

Figure 2: Residual plots

c) Variables Clarity, Oakiness are not significant when they fit as simple linear regression with the response (p values < 0.05). Other variables have a significant association between the predictor and the response. Exploratory analysis plots and residual plots also confirm the above conclusion.

	Estimate	R^2	Adjusted R^2	F-Statistic	p-value
Clarity	0.4692	0.0008089	-0.02695	0.02914	0.8654
Aroma	1.3365	0.5003	0.4864	36.04	6.871e-07
Body	1.3618	0.3011	0.2817	15.51	0.0003612
Flavor	1.5719	0.6242	0.6137	59.79	3.683e-09
Oakiness	-0.1304	0.002213	-0.0255	0.07984	0.7791
Region2 | Region3	-1.5320 | 2.6069	0.6113	0.5891	27.52	6.587e-08

Table 2: Model summay for each predictor

d) Only for the predictor Flavor, we can reject the null hypothesis $H_0 : \beta_j = 0$ at a 5% significant level as the p value (0.00006) is less than 0.05. All other p-vales are greater than 0.05.

e) Select predictor variables Flavor, Oakiness, and Region by using the stepwise AIC algorithm. But the predictor Oakiness is still insignificant in the model as the p-value (0.1281) is greater than 0.05. Therefore remove Oakiness from the model. Then Flavor and Region were fitted with interactions. There the interaction terms were not significant as the p-values were greater than 0.05. Remove interaction and select the model with Flavor and Region as the reasonably good multiple linear regression.

f) Final model:

$$Quality = 7.0943 + 1.1155(Flavor) - 1.5335(Region2) + 1.2234(Region3)$$

g)

	Means value	Lower limit	Upper limit
Prediction I.	12.41371	10.53775	14.28967
Confidence I.	12.41371	11.95152	12.8759

Table 3: 95% Prediction interval and confidence interval

Question 02

a) The exploratory analysis plots visually reveal that the predictor GPA has clear differences in the three groups (higher GPA tends to come from Group 1 while lower GPA tends to come from Group 2). But, predictor GMAT has similar values for groups 2 and 3 (distinguish between Group 2 and 3 is difficult). Therefore predictor GPA is more helpful in predicting response.

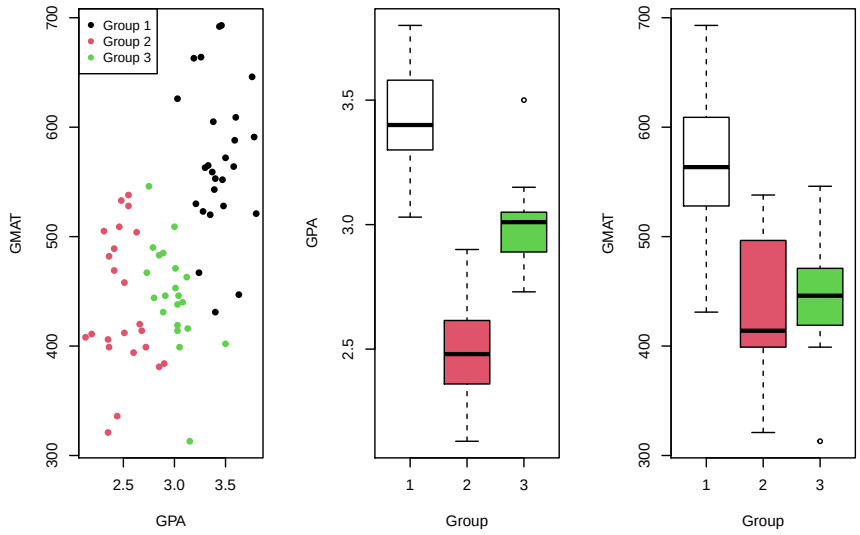

Figure 3: Exploratory analysis plots

b) LDA was performed using the traing data. The decision boundary seems sensible as it correctly classifies most of the training data. According to the confusion matrices, the overall misclassification rates of both training and test data are very low (misclassification rate of training data =0.086 and misclassification rate of test data= 0.2).

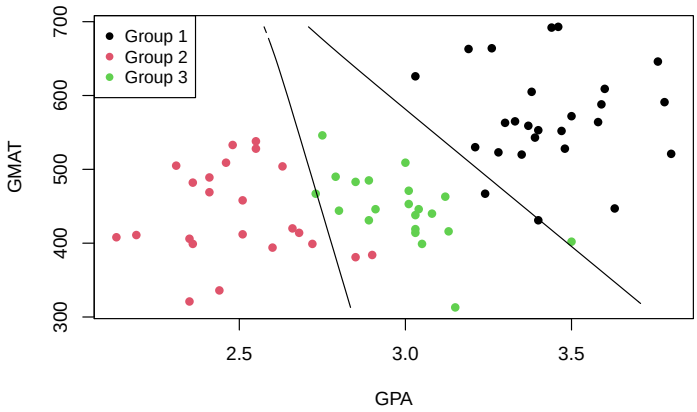

Figure 4: Decision boundary (LDA) using training data

	Predicted group		
True group	1	2	3
1	24	0	2
2	0	21	2
3	1	1	19

Table 4: Confusion matrix for training data

	Predicted group		
True group	1	2	3
1	2	0	3
2	0	5	0
3	0	0	5

Table 5: Confusion matrix for test data

c) QDA was performed using the traing data. The decision boundary seems sensible as it correctly classifies most of the training data. According to the confusion matrices, the overall misclassification rates of both training and test data are very low (misclassification rate of training data =0.029 and misclassification rate of test data= 0.067).

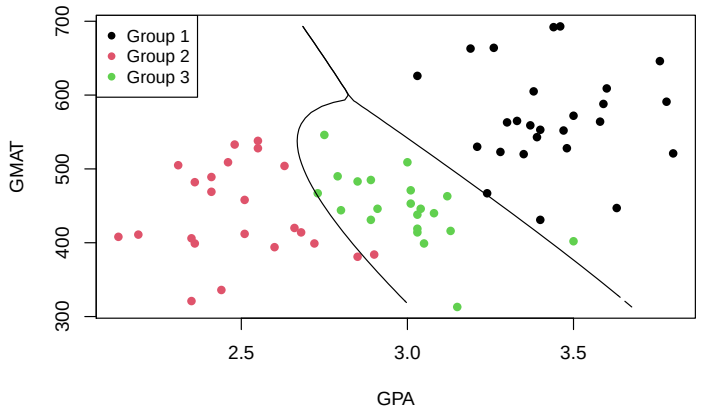

Figure 5: Decision boundary (QDA) using training data

	Predicted group		
True group	1	2	3
1	26	0	0
2	0	22	1
3	1	0	20

Table 6: Confusion matrix for training data

	Predicted group		
True group	1	2	3
1	4	0	1
2	0	5	0
3	0	0	5

Table 7: Confusion matrix for test data

d) Overall misclassification rates for QDA are better than LDA. Therefore I would recommend QDA as a classifier for this dataset.

Qusetion 3

a) Most of the predictors are useful in predicting the response (Outcome) except Blood Pressure. Because the distributions of Blood Pressure are quite similar for the two levels of the outcome variable.

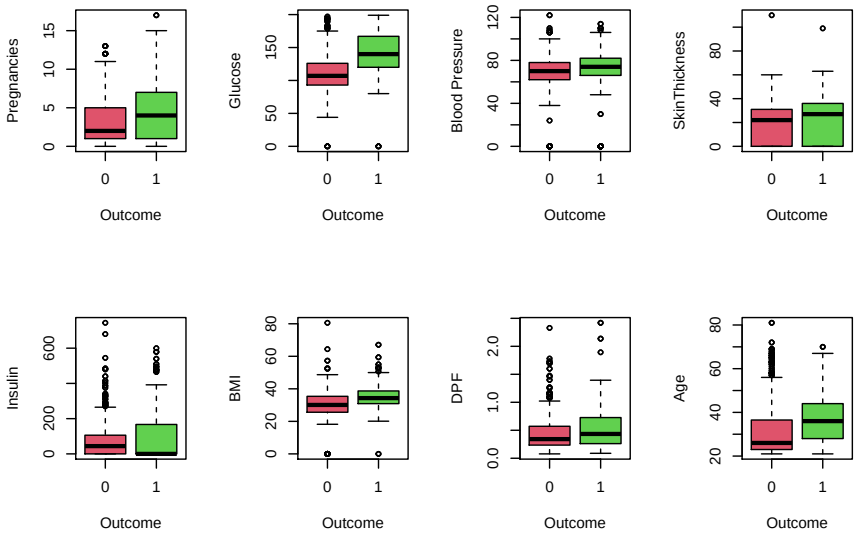

Figure 6: Exploratory analysis plots

b) LDA was fitted for the complete dataset. Here the overall misclassification rate (0.22) is low and the sensitivity is lower than specificity.But in this case sensitivity should be lager than specificity. Therefore 0.5 is not that much good cutoff for the posterior probability.

	Predicted group	
True group	0	1
0	1174	142
1	298	386

Table 8: Confusion matrix for training data

	Value
Sensivity	0.5643
Specificity	0.8921
Overall misclassification	0.22

Table 9: Measures for classification

c) QDA was fitted for the complete dataset. Here the overall misclassification rate is low (0.2355) and the sensitivity is lower than specificity.But in this case sensitivity should be lager than specificity.Therefore 0.5 is not that much good cutoff for the posterior probability. Note that ROC curves for both LDA and QDA are quite similar to each other.

	Predicted group	
True group	0	1
0	1135	181
1	290	394

Table 10: Confusion matrix for training data

	Value
Sensivity	0.5760
Specificity	0.8625
Overall misclassification	0.2355

Table 11: Measures for classification

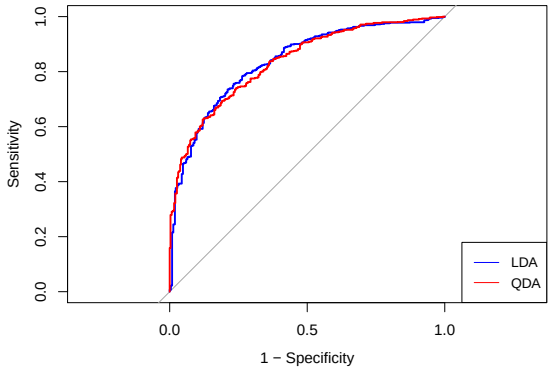

Figure 7: ROC curves for both LDA and QDA

d) I would recommend the QDA approach since the sensitivity is a bit better than LDA even though the misclassification rates suggest otherwise. But, here both approaches provide relatively similar outcomes. In this case, sensitivity should be larger than specificity. To increase the sensitivity we should decrease the posterior cutoff (from 0.5). Select posterior cutoff as 0.2. Now the sensitivity is higher (0.82) and also the specificity is also a reasonably good value.

	Predicted group	
True group	0	1
0	888	428
1	121	563

Table 12: Confusion matrix for training data

	Value
Sensivity	0.8231
Specificity	0.6748
Overall misclassification	0.2745

Table 13: Measures for classification

R Codes

```
knitr::opts_chunk$set(echo = TRUE)

## ----setup, include=FALSE-----------------------------------------------------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)


## ---- include=FALSE----------------------------------------------------------------------------------------------------
#########################################################################################################################

### Question 01

wine<-read.table("wine.txt",header = T)
wine$Region<-as.factor(wine$Region)
str(wine)


## ----echo=FALSE,fig.align="center",fig.cap="Pairwise scatterplots"------------------------------------------------------
## a)

# Pairwise scatterplots

panel.hist <- function(x, ...)
{
usr <- par("usr"); on.exit(par(usr))
par(usr = c(usr[1:2], 0, 1.5) )
h <- hist(x, plot = FALSE)
breaks <- h$breaks; nB <- length(breaks)
y <- h$counts; y <- y/max(y)
rect(breaks[-nB], 0, breaks[-1], y, ...)
}

my_cols <- c("red", "blue","green")
pairs(wine[,1:6], pch =c( 15,17,19)[wine$Region], col=c("red", "blue","dark green")[wine$Region],cex = 0.9,upper
par(xpd=TRUE)
legend("topright", legend=c("Region 1", "Region 2","Region 3"), col=c("red", "blue","dark green"),pch = c(15,17,



## ----results="asis", echo=FALSE-----------------------------------------------------------------------------------------
# Correlation Matrix
library(xtable)
options(xtable.comment=FALSE)
xtable(cor(wine[,1:6]),caption = "Correlation Matrix",placement = "H")


## ----echo=FALSE, fig.align="center",fig.cap="Residual plots", out.width = "40%"-----------------------------------------
## b)

#Residual plots
model1<-lm(Quality~.,wine)
par(mfrow=c(2,2))
plot(model1)

## ----include=FALSE------------------------------------------------------------------------------------------------------
#Test for normality of error term
shapiro.test(model1$residuals)

## ----include=FALSE------------------------------------------------------------------------------------------------------
#Test for Constant Variance (Homoscedasticity)
#Breusch-Pagan test:(depend on normality of errors)
library(lmtest)
```

```r
bptest(model1)

## ----include=FALSE-----------------------------------------------------------------------------------------------------------
# Durbin-Watson test to check whether residuals are uncorrelated
library(car)
durbinWatsonTest(model1)


## ----include=FALSE-----------------------------------------------------------------------------------------------------------
## c)

#Fit simple linear regression for each predictor,
model_C<-lm(Quality~Clarity,wine)
summary(model_C)
model_A<-lm(Quality~Aroma,wine)
summary(model_A)
model_B<-lm(Quality~Body,wine)
summary(model_B)
model_F<-lm(Quality~Flavor,wine)
summary(model_F)
model_O<-lm(Quality~Oakiness,wine)
summary(model_O)
model_R<-lm(Quality~Region,wine)
summary(model_R)


## ----include=FALSE-----------------------------------------------------------------------------------------------------------
par(mfrow=c(3,4))
plot(model_C,main = "Clarity")
plot(model_A,main = "Aroma")
plot(model_B,main = "Body")


## ----include=FALSE-----------------------------------------------------------------------------------------------------------
par(mfrow=c(3,4))
plot(model_F,main = "Flavor")
plot(model_O,main = "Oakiness")
plot(model_R,main = "Region")


## ----include=FALSE, results="asis"-------------------------------------------------------------------------------------------
## d)

model_full<-lm(Quality~.,wine)

library(jtools)
print(summ(model_full, model.info = FALSE, digits = 5))


## ----include=FALSE-----------------------------------------------------------------------------------------------------------
## e)

# Using stepAIC function. Find the best reduce model by removing unimporatant variables.
library(MASS)
low <- lm(Quality~Flavor + Region ,data=wine)
Red.model <- stepAIC(model_full, scope = list(lower = low, upper = model_full),direction = "both",trace=FALSE)
summary(Red.model)


## ----include=FALSE-----------------------------------------------------------------------------------------------------------
# fit the model for the seclected variables with interaction
model_inter_new<- lm(Quality~Flavor + Region + Flavor * Region , data=wine)
summary(model_inter_new)
```

```r
## ----include=FALSE-----------------------------------------------------------
model_reduced<-lm(Quality~Flavor + Region, data=wine)
summary(model_reduced)


## ----include=FALSE-----------------------------------------------------------
# Model assumptions
#Residual plots
par(mfrow=c(2,2))
plot(model_reduced)

## ----include=FALSE-----------------------------------------------------------
#Test for normality of error term
shapiro.test(model_reduced$residuals)

## ----include=FALSE-----------------------------------------------------------
#Test for Constant Variance (Homoscedasticity)
#Breusch-Pagan test:(depend on normality of errors)
library(lmtest)
bptest(model_reduced)

## ----include=FALSE-----------------------------------------------------------
# Durbin-Watson test to check whether residuals are uncorrelated
library(car)
durbinWatsonTest(model_reduced)


## ----include=FALSE-----------------------------------------------------------
## g)

pred_data<-data.frame(Flavor=mean(wine$Flavor),Region="1")
predict(model_reduced,pred_data)

predict(model_reduced, newdata = pred_data, interval = "prediction",level=0.95)
predict(model_reduced, newdata = pred_data, interval = "confidence",level=0.95)


## ---- include=FALSE---------------------------------------------------------
################################################################################################################

### Question 02

admission <- read.csv("admission.csv", header = T)
admission$Group<- as.factor(admission$Group)
str(admission)
test.admission<- rbind(head(admission[admission$Group==1,],5),head(admission[admission$Group==2,],5),head(admiss
train.admission<-rbind(admission[admission$Group==1,][-c(1:5),],admission[admission$Group==2,][-c(1:5),],admissi


## ----echo=FALSE,fig.align="center",fig.cap="Exploratory analysis plots", out.width = "60%"----------------------
## a)

# Plot data
par(mfrow = c(1, 3))
plot(train.admission[,1:2], col = train.admission$Group,pch=16)
legend("topleft", legend=c("Group 1", "Group 2","Group 3"), col=1:3 ,pch =16 , cex=0.9)
boxplot(train.admission[, "GPA"] ~ train.admission$Group , xlab= "Group" ,ylab = "GPA",col=c(0,2,3))
boxplot(train.admission[, "GMAT"] ~ train.admission$Group , xlab= "Group" ,ylab = "GMAT",col=c(0,2,3))



## ----include=FALSE-----------------------------------------------------------
## b)
```

```r
library(MASS)

lda.fit1 <- lda(Group ~ GPA + GMAT, data =train.admission )
lda.fit1


## ----echo=FALSE,fig.align="center",fig.cap="Decision boundary (LDA) using training data", out.width = "50%"---
# 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(train.admission[, 1]), t = max(train.admission[, 1]), l = n.grid)
x2.grid <- seq(f = min(train.admission[, 2]), t = max(train.admission[, 2]), l = n.grid)
grid <- expand.grid(x1.grid, x2.grid)
colnames(grid) <- colnames(train.admission[,1:2])

pred.grid <- predict(lda.fit1, grid)


prob1 <- matrix(pred.grid$posterior[, 1], nrow = n.grid, ncol = n.grid, byrow = F)
prob2 <- matrix(pred.grid$posterior[, 2], nrow = n.grid, ncol = n.grid, byrow = F)
plot(train.admission[,1:2], col = train.admission$Group,pch=16)
legend("topleft", legend=c("Group 1", "Group 2","Group 3"), col=1:3 ,pch =16 , cex=0.9)

contour(x1.grid, x2.grid, prob1, levels = 0.5, labels = "", xlab = "", ylab = "",  main = "", add = T)
contour(x1.grid, x2.grid, prob2, levels = 0.5, labels = "", xlab = "", ylab = "",  main = "", add = T)

## ----include=FALSE-----------------------------------------------------------------------------------------

lda.pred.train <- predict(lda.fit1, train.admission)
train.con.mat<-table(train.admission$Group,lda.pred.train$class)

lda.pred.test <- predict(lda.fit1, test.admission)
test.con.mat<-table(test.admission$Group,lda.pred.test$class)


## ----include=FALSE, results="asis"-------------------------------------------------------------------------
library(xtable)
options(xtable.comment=FALSE)
row1<-list()
row1$pos<-list(0,0)
row1$command<-c("& \\multicolumn{3}{c}{Predicted group}\\\\\n","True group & 1 & 2 & 3  \\\\\n  ")

# Confusion matrix for training data
t1<-print(xtable(train.con.mat, caption = "Confusion matrix for training data"),add.to.row = row1,include.colnam

row2<-list()
row2$pos<-list(0,0)
row2$command<-c("& \\multicolumn{3}{c}{Predicted group}\\\\\n","True group & 1 & 2 & 3  \\\\\n  ")

# Confusion matrix for test data
t2<-print(xtable(test.con.mat, caption = "Confusion matrix for test data"),add.to.row= row2,include.colnames =FA



## ----include=FALSE-----------------------------------------------------------------------------------------
# misclassification rate of training data
mean(lda.pred.train$class != train.admission$Group)
# misclassification rate of test data
mean(lda.pred.test$class != test.admission$Group)


## ----include=FALSE-----------------------------------------------------------------------------------------
```

```r
## c)

qda.fit1 <- qda(Group ~ GPA + GMAT, data = train.admission)
pred.grid.qda <- predict(qda.fit1, grid)


## ----echo=FALSE,fig.align="center",fig.cap="Decision boundary (QDA) using training data", out.width = "50%"---
p11_qda=pred.grid.qda$posterior[,1] - pmax(pred.grid.qda$posterior[,2],pred.grid.qda$posterior[,3])
p22_qda=pred.grid.qda$posterior[,2] - pmax(pred.grid.qda$posterior[,1],pred.grid.qda$posterior[,3])
prob_qda1 <- matrix(p11_qda, nrow = n.grid, ncol = n.grid, byrow = F)
prob_qda2 <- matrix(p22_qda, nrow = n.grid, ncol = n.grid, byrow = F)
plot(train.admission[,1:2], col = train.admission$Group,pch=16)
legend("topleft", legend=c("Group 1", "Group 2","Group 3"), col=1:3 ,pch =16 , cex=0.9)

contour(x1.grid, x2.grid, prob_qda1, levels = 0, labels = "", xlab = "", ylab = "",main = "", add = T)
contour(x1.grid, x2.grid, prob_qda2, levels = 0, labels = "", xlab = "", ylab = "",main = "", add = T)


## ----include=FALSE-----------------------------------------------------------------------------------------------

qda.pred.train <- predict(qda.fit1, train.admission)
train.con.mat2<-table(train.admission$Group,qda.pred.train$class)

qda.pred.test <- predict(qda.fit1, test.admission)
test.con.mat2<-table(test.admission$Group,qda.pred.test$class)


## ----include=FALSE, results="asis"-------------------------------------------------------------------------------
library(xtable)
options(xtable.comment=FALSE)
row11<-list()
row11$pos<-list(0,0)
row11$command<-c("& \\multicolumn{3}{c}{Predicted group}\\\\\n","True group & 1 & 2 & 3  \\\\\n  ")

# Confusion matrix for training data
t11<-print(xtable(train.con.mat2, caption = "Confusion matrix for training data"),add.to.row = row11,include.col

row22<-list()
row22$pos<-list(0,0)
row22$command<-c("& \\multicolumn{3}{c}{Predicted group}\\\\\n","True group & 1 & 2 & 3  \\\\\n  ")

# Confusion matrix for test data
t22<-print(xtable(test.con.mat2, caption = "Confusion matrix for test data"),add.to.row= row22,include.colnames



## ----include=FALSE-----------------------------------------------------------------------------------------------
# misclassification rate of training data
mean(qda.pred.train$class != train.admission$Group)
# misclassification rate of test data
mean(qda.pred.test$class != test.admission$Group)


## ---- include=FALSE---------------------------------------------------------------------------------------------
##########################################################################################################

### Qusetion 3

diabetes<-read.csv("diabetes.csv",header = T)
diabetes$Outcome<-as.factor(diabetes$Outcome)
str(diabetes)
```

```r
## ----echo=FALSE,fig.align="center",fig.cap="Exploratory analysis plots",  out.width = "60%"----------
## a)

par(mfrow=c(2,4))
boxplot(diabetes[, 1] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Pregnancies",col=c(2,3))
boxplot(diabetes[, 2] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Glucose",col=c(2,3))
boxplot(diabetes[, 3] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Blood Pressure",col=c(2,3))
boxplot(diabetes[, 4] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "SkinThickness",col=c(2,3))
boxplot(diabetes[, 5] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Insulin",col=c(2,3))
boxplot(diabetes[, 6] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "BMI",col=c(2,3))
boxplot(diabetes[, 7] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "DPF",col=c(2,3))
boxplot(diabetes[, 8] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Age",col=c(2,3))


## ----include=FALSE----------------------------------------------------------------------------------
## b)

library(MASS)

lda.fit <- lda(Outcome ~ Pregnancies.. + Glucose.. + BloodPressure.. + SkinThickness..
               + Insulin.. + BMI.. + DiabetesPedigreeFunction.. + Age.. , data =diabetes )
lda.fit


## ----include=FALSE----------------------------------------------------------------------------------

lda.pred <- predict(lda.fit, diabetes)
con.mat<-table(diabetes$Outcome,lda.pred$class)


## ----include=FALSE, results="asis"------------------------------------------------------------------
library(xtable)
options(xtable.comment=FALSE)
row<-list()
row$pos<-list(0,0)
row$command<-c("& \\multicolumn{2}{c}{Predicted group}\\\\\\n","True group & 0 & 1  \\\\\\n  ")

# Confusion matrix for training data
print(xtable(con.mat, caption = "Confusion matrix for data"),add.to.row = row,include.colnames = FALSE,table.pla


## ----include=FALSE----------------------------------------------------------------------------------
## c)

library(MASS)

qda.fit <- qda(Outcome ~ Pregnancies.. + Glucose.. + BloodPressure.. + SkinThickness..
               + Insulin.. + BMI.. + DiabetesPedigreeFunction.. + Age.. , data =diabetes )
qda.fit


## ----include=FALSE----------------------------------------------------------------------------------

qda.pred <- predict(qda.fit, diabetes)
q.con.mat<-table(diabetes$Outcome,qda.pred$class)


## ----include=FALSE, results="asis"------------------------------------------------------------------
library(xtable)
options(xtable.comment=FALSE)
rowq<-list()
rowq$pos<-list(0,0)
rowq$command<-c("& \\multicolumn{2}{c}{Predicted group}\\\\\\n","True group & 1 & 2  \\\\\\n  ")
```

```r
# Confusion matrix for training data
print(xtable(q.con.mat, caption = "Confusion matrix for data"),add.to.row = row,include.colnames = FALSE,table.p


## ----echo=FALSE,fig.align="center",fig.cap="ROC curves for both LDA and QDA",message = FALSE,warning = FALSE,
# ROC curves
library(pROC)
#options(pROC.comment=FALSE)

roc.lda <- roc(diabetes$Outcome, lda.pred$posterior[, 1], levels = c(1, 0))
roc.qda <- roc(diabetes$Outcome, qda.pred$posterior[, 1], levels = c(1, 0))
plot(roc.lda, legacy.axes = T,col="blue")
plot(roc.qda, add = T, col = "red")
legend("bottomright", legend=c("LDA", "QDA"), col=c("blue","red") ,lty = 1 , cex=0.9)


## ----include=FALSE-----------------------------------------------------------------------------------------------
## d)

qda.pre.adj<-ifelse(qda.pred$posterior[,2]> 0.2,1,0)
table(diabetes$Outcome,qda.pre.adj)
```