Question One

a) If we choose p(x) = 0.5 it moves the decision boundary to min testing error.

$\hat{\beta}_0 = -16 \quad \hat{\beta}_1 = 1.4 \quad \hat{\beta}_2 = 0.3 \quad x_1 = 5\ hours \quad x_2 = 36\ classes$

$$P(GPA \geq 7) = \frac{\exp(\hat{\beta}_0+\hat{\beta}_1 x_1+\hat{\beta}_2 x_2)}{1+\exp(\hat{\beta}_0+\hat{\beta}_1 x_1+\hat{\beta}_2 x_2)} = \frac{\exp(-16+1.4*5+0.3*36)}{1+\exp(-16+1.4*5+0.3*36)} = 0.00074$$

b) $-16 + 1.4x + 0.3 * 18$

$= 1.4x - 10.6$

If we set this equal to k

$$\frac{e^k}{1+e^k} = 0.5$$

$$= e^k = 0.5 + 0.5e^k$$

$$=0.5e^k = 0.5$$

$$= e^k = 1$$

$$\ln(e^k) = \ln(1)$$

$$k = 0$$

$$1.4x - 10.6 = 0$$

$$= x = \frac{10.6}{1.4} = 7.57$$

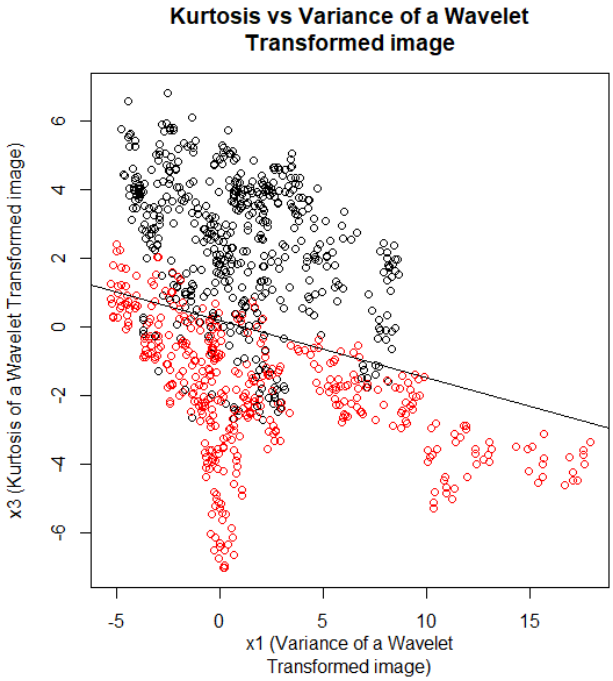
Question Two

a) In the logistic model, the p values for the coefficients x1 and x3 are both low and show that they are statistically significant. The Intercept has a p-value close to 0.05 which means it would not be significant if using a confidence level of 0.99, however we can just centre the data to solve this.

```
Call:
glm(formula = y ~ x1 + x3, family = binomial, data = dataq1train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.83187  -0.28343  -0.06417   0.50032   1.99366

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.22041    0.11206   1.967   0.0492 *
x1          -1.31489    0.08822 -14.905  < 2e-16 ***
x3          -0.21738    0.02880  -7.548 4.42e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1322.01  on 959  degrees of freedom
Residual deviance:  572.07  on 957  degrees of freedom
AIC: 578.07

Number of Fisher Scoring iterations: 6
```

b)

(i)



(ii) In the confusion matrix below, we see that the Test error of 56/412 = 0.1359 is better than using the null classifier of 176/412 = 0.4272.
The precision of the model is 152/184 = 0.826
Specificity (true negative rate) is 204/236 = 0.86
Sensitivity (true positive rate) is 152/176 = 0.86
For the example, this means that we are correctly predicting 86% of the forged banknotes and 86% of the genuine banknotes.

```
glm.pred0.5   0   1
          0 204  24
          1  32 152
```

(iii) Using θ = 0.3, we have:
We see that the Test error of 58/412 = 0.1408 is better than using the null classifier of 176/412 = 0.4272.
The precision of the model is 171/224 = 0.763
Specificity (true negative rate) is 183/236 = 0.78
Sensitivity (true positive rate) is 171/176 = 0.97
For the example, this means that we are correctly predicting 97% of the forged banknotes and 78% of the genuine banknotes.

```
glm.pred0.3   0   1
          0 183   5
          1  53 171
```

Using $\theta = 0.6$, we have:
In the confusion matrix below, we see that the Test error of 61/412 = 0.1481 is better than using the null classifier of 176/412 = 0.4272.
The precision of the model is 141/167 = 0.763
Specificity (true negative rate) is 210/236 = 0.89
Sensitivity (true positive rate) is 141/176 = 0.80
For the example, this means that we are correctly predicting 80% of the forged banknotes and 89% of the genuine banknotes.

```
glm.pred0.6   0   1
          0 210  35
          1  26 141
```

This shows that when we increase the value of θ, we see an increase in the specificity but a drop in the sensitivity and the reverse when we decrease θ. We would use θ=0.3 in a situation where correctly predicting the forged banknotes is more important than predicting the genuine notes. A situation where this may occur is in a casino, where we want to correctly predict the false banknotes that are coming through. If we are sorting a large amount of money from a single customer, we would not want to have to test every possible note as this would take a long period of time. We would instead use a subset of these notes and we would want to decrease θ to the lowest possible value (in our case we did to 0.3) within our budget so we can be the most accurate as possible in predicting the false banknotes.

Question Three

a) LDA Model

```
     0   1
0 203  22
1  33 154
```

b) QDA Model

```
qda.class   0   1
        0 208  18
        1  28 158
```

c)

Comparing the three models, it appears that QDA is the best model as it has the highest specificity, sensitivity and precision while also having the lowest testing error.

	Specificity	Sensitivity	Precision	Testing Error
Logistic Regression model	0.864	0.864	0.826	0.136
LDA	0.860	0.875	0.823	0.133
QDA	0.881	0.898	0.849	0.112

Question Four

$f_0(x) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{2*4}(x-0)^2}$
$\pi_0(x) = 0.69$

$f_1(x) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{2*4}(x-2)^2)}$

$\pi_0(x) = 0.4$
$\pi_1(x) = 0.6$

Bayes decision boundary:
$f_0(x) * \pi_0(x) = f_1(x) * \pi_1(x)$

$0.4 * \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{2*4}(x-0)^2} = 0.6* \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{2*4}(x-2)^2)}$

$= 0.4 * \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}x^2} = 0.6* \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}(x-2)^2)}$

$= 0.4 * e^{-\frac{1}{8}x^2} = 0.6* e^{-\frac{1}{8}(x-2)^2)}$

$= \ln(0.4) - \frac{1}{8}x^2 = \ln(0.6)* -\frac{1}{8}(x-2)^2)$

$= \ln(0.4) - \ln(0.6) = \frac{1}{8}x^2 - \frac{1}{8}(x-2)^2)$

$= 8\ln(0.4) - 8\ln(0.6) = x^2 - (x^2 - 4x + 4)$

$= 8\ln(0.4) - 8\ln(0.6) = 4x - 4$

$= 2\ln(0.4) - 2\ln(0.6) = x - 4$

$= 2\ln(0.4) - 2\ln(0.6) = x$

$= x = 0.1891$

pnorm(x, mean0, sd) = 0.5376639

pnorm(x, mean1, sd) = 0.1826135

Bayes Error Rate = 0.1826135 + (1-0.5376639)

= 0.6449496

R-Code

```
library(ISLR)
#####################QUESTION ONE####################
#Is a bunch of math


#####################QUESTION TWO####################

#a
dataq1train = read.csv('BankTrain.csv')
dataq1test = read.csv('BankTest.csv')

glmq1a <- glm(y~x1+x3,data= dataq1train, family = binomial )

summary(glmq1a)


#bi

glmq1a.probs=predict(glmq1a,dataq1train,type="response")

glm.pred=rep("0",960)

glm.pred[glmq1a.probs>.5]="1"


datameow = dataq1train[1:960,c(1,3,5)]

datameow = cbind(datameow,glm.pred)

plot(datameow$x3,
     datameow$x1,
     col=datameow$y+1,
     main = "Kurtosis vs Variance of a Wavelet
Transformed image",
     xlab = "x1 (Variance of a Wavelet
Transformed image)",
     ylab = "x3 (Kurtosis of a Wavelet Transformed image)")


abline(-(0.22041/-1.31489),-(-0.21738/-1.31489))



#bii

glmq1test.probs=predict(glmq1a,dataq1test,type="response")
glm.pred0.5=rep("0",412)

glm.pred0.5[glmq1test.probs>.5]="1"


table(glm.pred0.5, dataq1test$y)


#biii


glm.pred0.3=rep("0",412)

glm.pred0.3[glmq1test.probs>.3]="1"


table(glm.pred0.3, dataq1test$y)


glm.pred0.6=rep("0",412)
```

```
glm.pred0.6[glmq1test.probs>.6]="1"


table(glm.pred0.6, dataq1test$y)


#####################QUESTION THREE#####################

################### A
library(MASS)

lda.fit=lda(y~x1+x3,data=dataq1train)
plot(lda.fit)

lda.fit

lda.pred=predict(lda.fit, dataq1test)



table(lda.pred$class, dataq1test$y)

################### B
qda.fit=qda(y~x1+x3,data=dataq1train)
qda.fit
qda.class=predict(qda.fit,dataq1test)$class
table(qda.class,dataq1test$y)


#####################QUESTION FOUR#####################

x=0.1891

mean0 = 0

mean1 = 2

sd = 2

pnorm(x, mean0, sd)

pnorm(x, mean1, sd)
```