Homework 2

Tanner Huck

4/18/2017

Problem 1

Intro) In this problem we will simulate data with a qualitative response $y = 1, 2$ and perform a k-nearest neighbors. We will have $p = 2$ and $n = 200$, 200 data points and 2 different independent variables.

a. First generating data with a non-linear Bayes decision boundary. We will sample a vector x_1 from $Norm(2, 2^2)$, x_2 from $Unif(-10, 10)$ and ϵ of random noise from $Norm(0, 1)$. Then create a decision boundary as $f(x_1, x_2) = x_2 - (1/4)*(x_1 - 2)^2 + 1 + $ with the classification criteria $y_i = 1$ if $f(x_1, x_2) > 0$ and 2 otherwise.

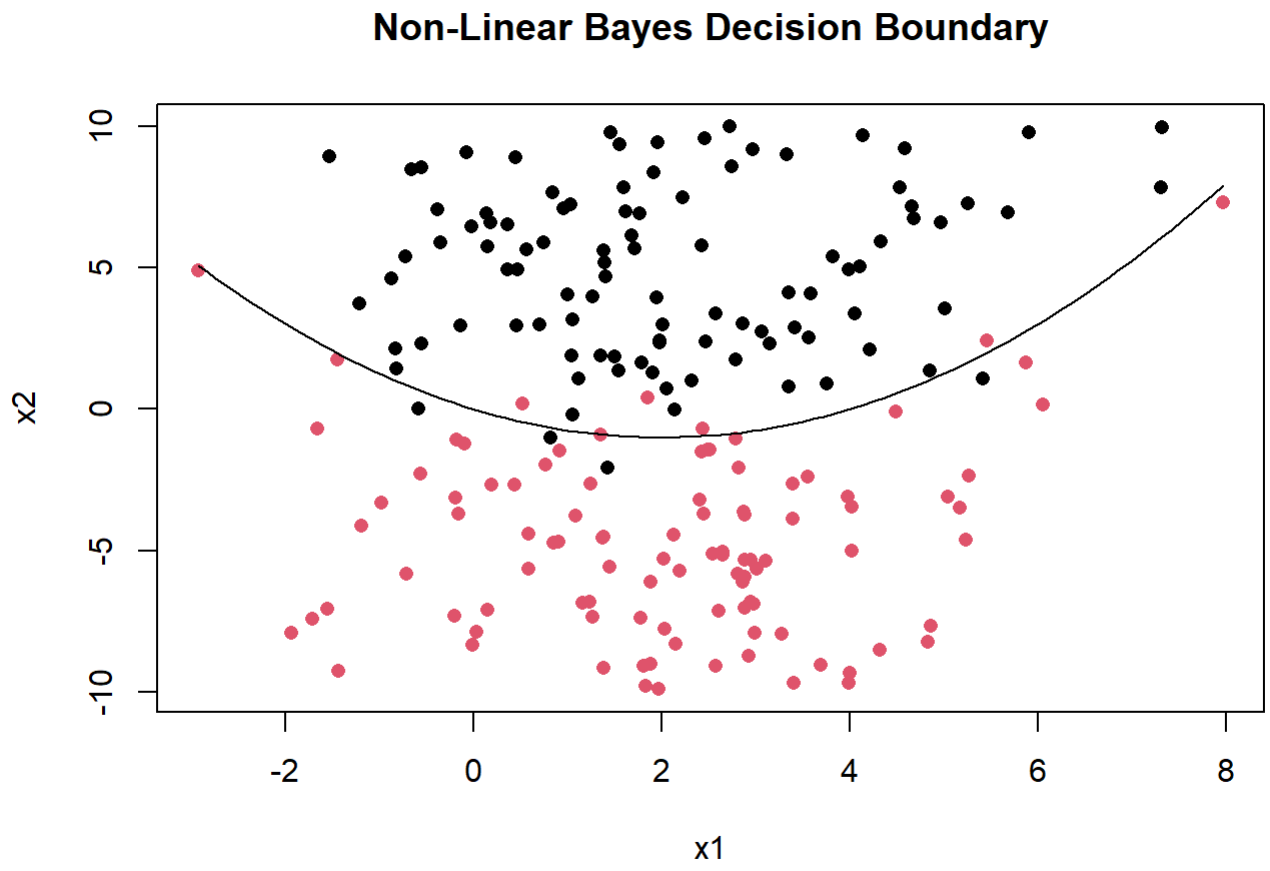
```
## y
##   1   2
## 100 100
```

After generating our data, decision boundary, and classification criteria, we can see that 100 of our data points have $y = 1$ and 100 have $y = 2$.

If we wanted to generate data with a linear decision boundary instead, we would modify our decision boundary equation to only contain linear terms, meaning we could change the $(1/4) * (x_1 - 2)^2$ term to just $(1/4) * (x_1 - 2)$.

If we wanted more than 2 classes, we could change our classification criteria, to $y_i = 1$ if $f(x_1, x_2) > 0$ and $y_i = 2$ if $f(x_1, x_2) \geq 1$, $y_i = 3$ otherwise. Now there are 3 possible classes.

b. Now plotting our data points,



We can see that the decision boundary line is fairly accurate in splitting our data. A couple points are on the wrong side of the line, but generally the boundary is good.

c. Making a knn function that takes training data, labels for training data, a new data point, and number of neighbors and perform k-nearest neighbors and return the label for the new data point.

```
# Knn function
my_knn <- function(x_train, y_train, x_new, k) {
  dists <- sqrt(rowSums(t(t(x_train) - x_new)^2))

  neighbors <- order(dists)[1:k]

  return(as.integer(names(sort(table(y_train[neighbors]), decreasing = TRUE))[1]))
}
```

```
## [1] 1
```

d. Performing a k-nearest neighbors on the 200 training data points with different k levels to find the k which produces the least training error. Then plotting the data with the best k and true bayes decision boundary. Finally, creating a two by two confusion matrix.

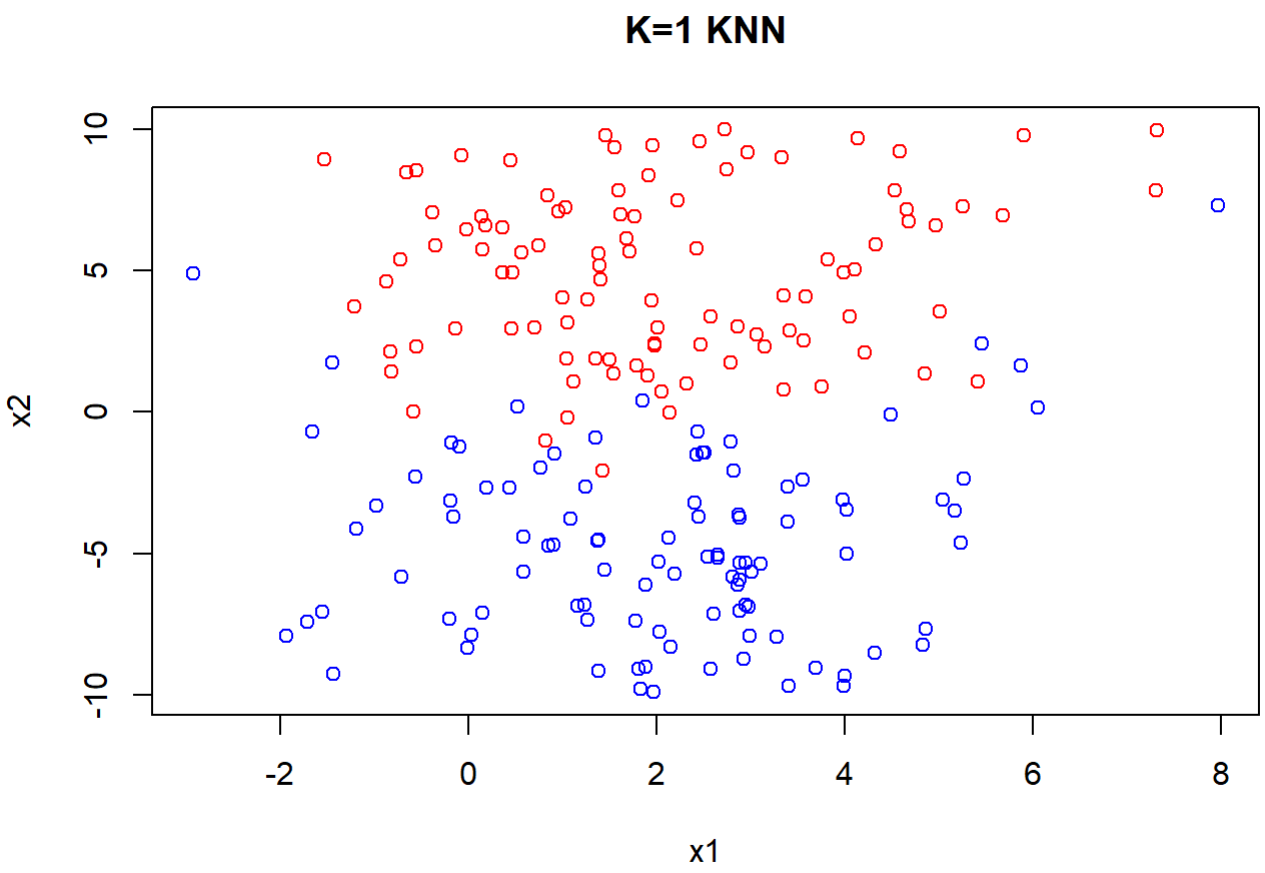
First calculating the training errors.

```
##  [1] 0.000 0.070 0.050 0.060 0.060 0.065 0.065 0.065 0.050 0.055 0.060 0.055
## [13] 0.055 0.060 0.050 0.055 0.055 0.060 0.055 0.060 0.055 0.055 0.055 0.055
## [25] 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.050 0.045 0.055 0.045 0.055
## [37] 0.050 0.055 0.055 0.055 0.055 0.055 0.055 0.055 0.055 0.055 0.055 0.055
## [49] 0.055 0.055
```

```
## [1] 1
```

From this code we can see that the k level with the smallest classifications rate is $k = 1$. This also makes sense intuitively because at $k = 1$ you are basically predicting every point with itself.

```
##     y
## knn2   1   2
##    1 100   0
##    2   0 100
```



Here we can see the graph using $k = 1$ KNN. In comparison with thee graph in part b, they are the same. Looking at the confusion matrix, it is also the same as the table from part a.

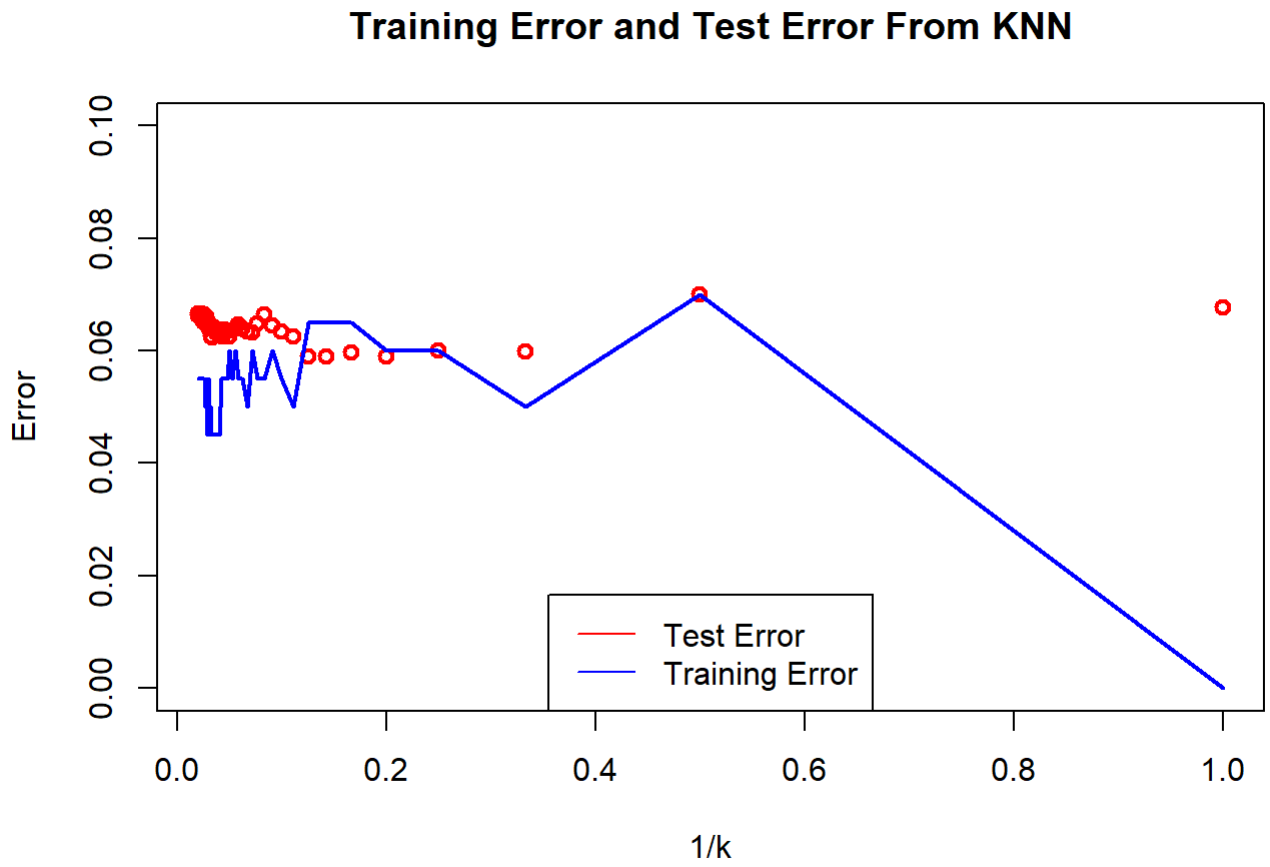
e. Continuing with the same data generating process as in a, now we want 5000 new points. Then we will perform k-nearest neighbors on the new 5000 points for different k-levels and calculate the test error for each k.

```
##  [1] 0.0676 0.0700 0.0598 0.0600 0.0590 0.0596 0.0590 0.0590 0.0626 0.0634
## [11] 0.0644 0.0664 0.0648 0.0632 0.0634 0.0640 0.0646 0.0640 0.0636 0.0626
## [21] 0.0636 0.0638 0.0626 0.0636 0.0632 0.0630 0.0632 0.0642 0.0642 0.0624
## [31] 0.0626 0.0640 0.0640 0.0650 0.0654 0.0660 0.0650 0.0650 0.0664 0.0660
## [41] 0.0664 0.0656 0.0660 0.0666 0.0664 0.0660 0.0666 0.0666 0.0662 0.0666
```

```
## [1] 5
```

After finding the test error for each different k-level, we see that $k = 5$ provides the smallest error of about 0.06.

f. Finally, making a plot of our errors found in part e.



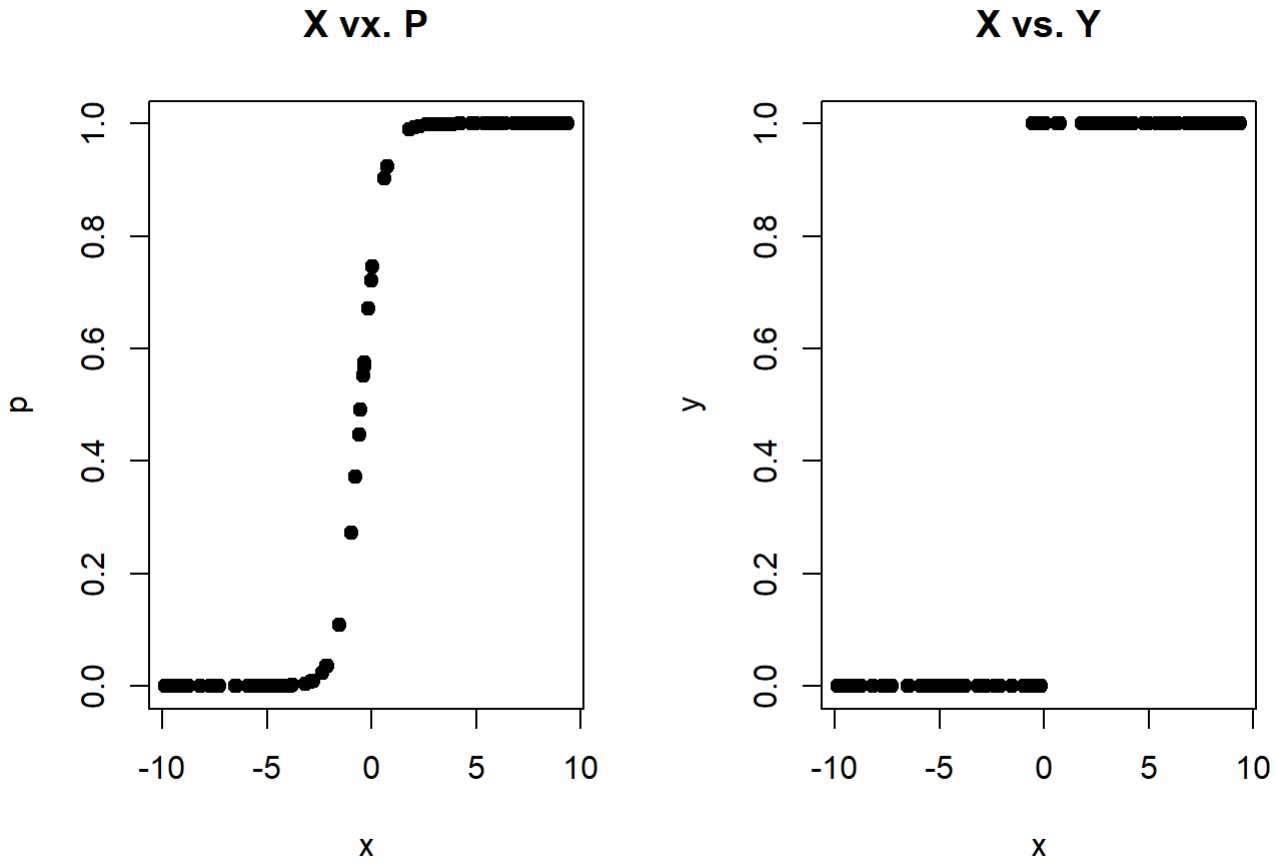
From our graph we do not see anything to surprising. We can see that the training error will decrease as we get closer to 1, and the test error hits a plateau. To choose the best k, I would choose a $1/k$ of around 0.15, which corresponds to about $k = 7$. This level seems to minimize both the training and test error well.

Problem 2

Intro) This question will use a simple probabilistic model to generate binary data. Then we will fit a logistic regression and attempt to recover the true parameters.

a. First generating the responses, using a logit model. We will start by sampling a vector x of length $n = 100$ from a $Unif \sim (10, 10)$. Then, take $[\beta_0, \beta_1] = [1, 2]$ and compute p, the vector of success probability

using the probability of success given by $P(Y = 1|X = x) = \frac{1}{1+e^{-(\beta_0+\beta_1)}}$. Note that $y|p_i \sim Bernoulli(p|i)$. Hence, we can simulate data with a Bernoulli trial.



b. Next, fitting a logistic model using the glm function and reporting the coefficients.

```
##
## Call:
## glm(formula = y ~ x, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.501   0.000   0.000   0.000   1.693
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.446      1.270   1.138    0.255
## x              5.032      3.074   1.637    0.102
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.058  on 99  degrees of freedom
## Residual deviance:  10.256  on 98  degrees of freedom
## AIC: 14.256
##
## Number of Fisher Scoring iterations: 12
```

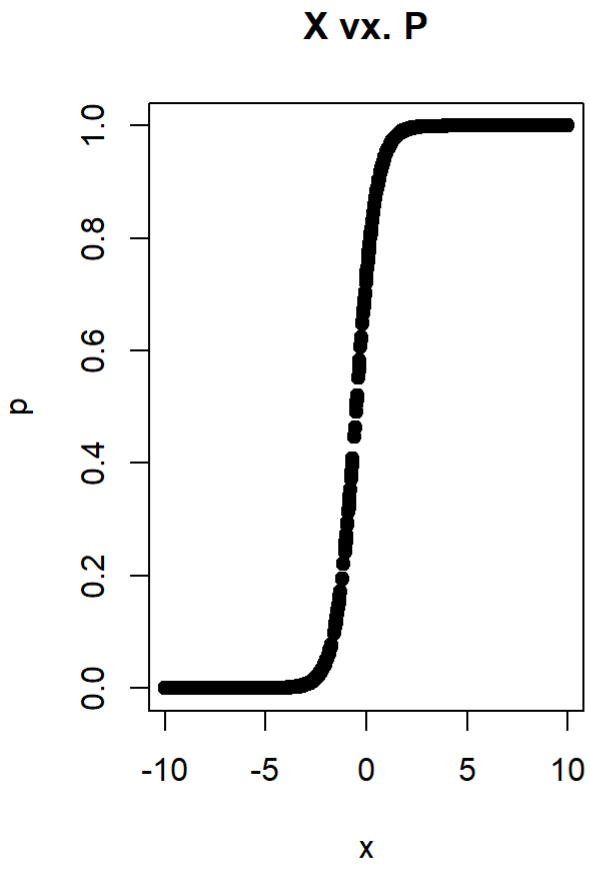
From our model we can see values for $\hat{\beta}_0$ and $\hat{\beta}_1$ of about 1.5 and 5.0. The estimate for $\hat{\beta}_0$ is fairly close to its true value of 1, but the estimate of $\hat{\beta}_1$ is not very close to its true value of 2.

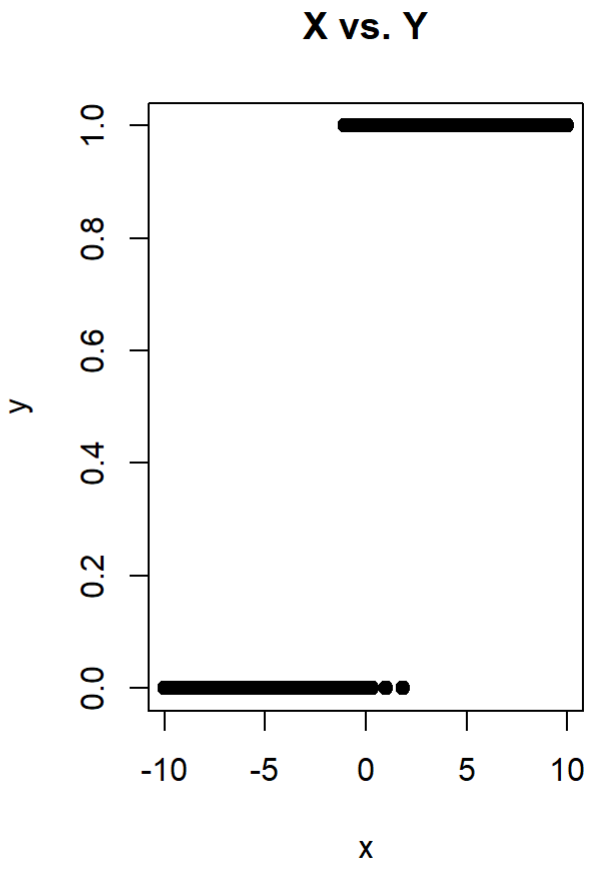
c. Now computing the predicted probabilities, $\hat{p}$ and generating a confusion matrix.

```
##         y
## outcomes  0  1
##        0 41  2
##        1  1 56
```

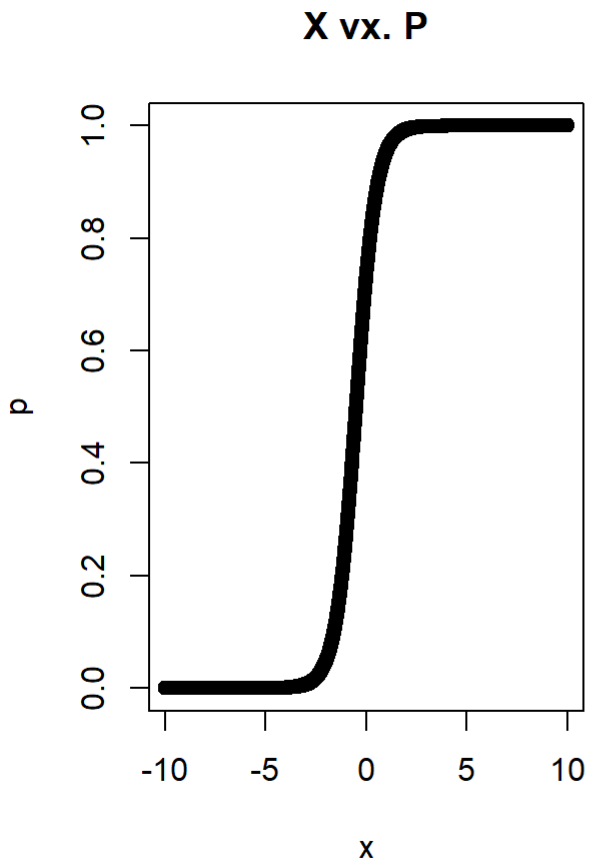
Here we can see the 2x2 confusion matrix. We can see that there are only 3 false negatives and positives, which is pretty good.

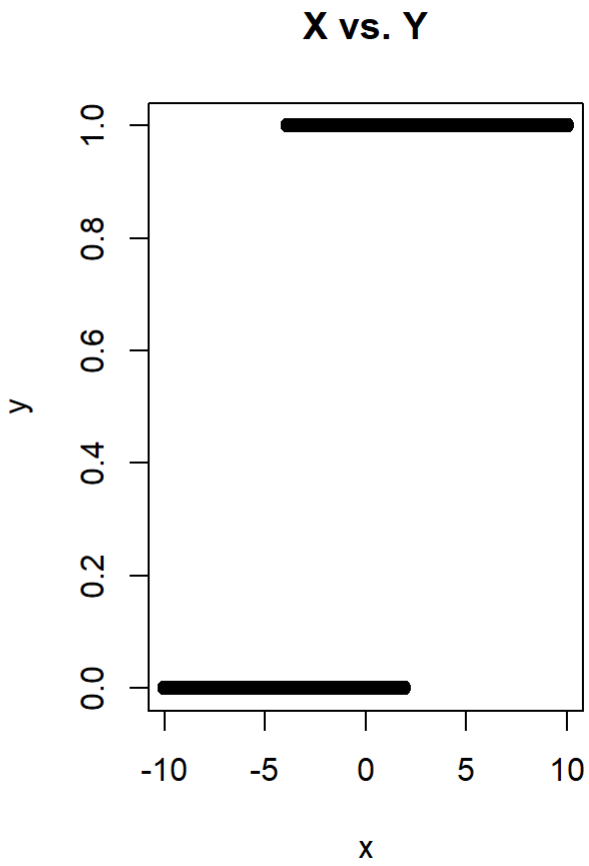
d. Repeating parts a and b with $n = 10^3$ and $n = 10^4$, then comparing the coefficients.





```
##
## Call:
## glm(formula = y_1 ~ x_1, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5111  -0.0011   0.0000   0.0020   2.0140
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.0690     0.2727   3.919 8.88e-05 ***
## x_1           2.7786     0.3766   7.378 1.61e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1377.06  on 999  degrees of freedom
## Residual deviance:  113.01  on 998  degrees of freedom
## AIC: 117.01
##
## Number of Fisher Scoring iterations: 10
```





```
##
## Call:
## glm(formula = y_2 ~ x_2, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0688  -0.0123   0.0001   0.0144   3.5590
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.06423    0.07097   15.00   <2e-16 ***
## x_2          1.91088    0.06475   29.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13819.6  on 9999  degrees of freedom
## Residual deviance:  1721.9  on 9998  degrees of freedom
## AIC: 1725.9
##
## Number of Fisher Scoring iterations: 9
```

Hence we obtain values for $\hat{\beta}_0$ and $\hat{\beta}_1$ of about 1.1 and 2.8 when $n = 10^3$ and about 1.1 and 1.9 when $n = 10^4$. this tells us that the coefficients are getting closer to their true values of 1 and 2 and n increases.

e. Finally, fitting a linear model to the $n = 100$ case instead of a logistic model.

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48884 -0.15744 -0.03067  0.18557  0.53862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.499151   0.024965   19.99   <2e-16 ***
## x           0.072896   0.004162   17.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2453 on 98 degrees of freedom
## Multiple R-squared:  0.7578, Adjusted R-squared:  0.7554
## F-statistic: 306.7 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
##         y
## outcomes  0  1
##        0 42  3
##        1  0 55
```

From our code we can see that the new confusion matrix for the linear model yields very similar results to the logistic model. For this linear model we are making the decision based on binary classification, so it makes sense to see similar answers to the logistic regression.

Problem 3

Intro) In this question we will explore linear and quadratic discriminant analysis techniques on new simulated data.

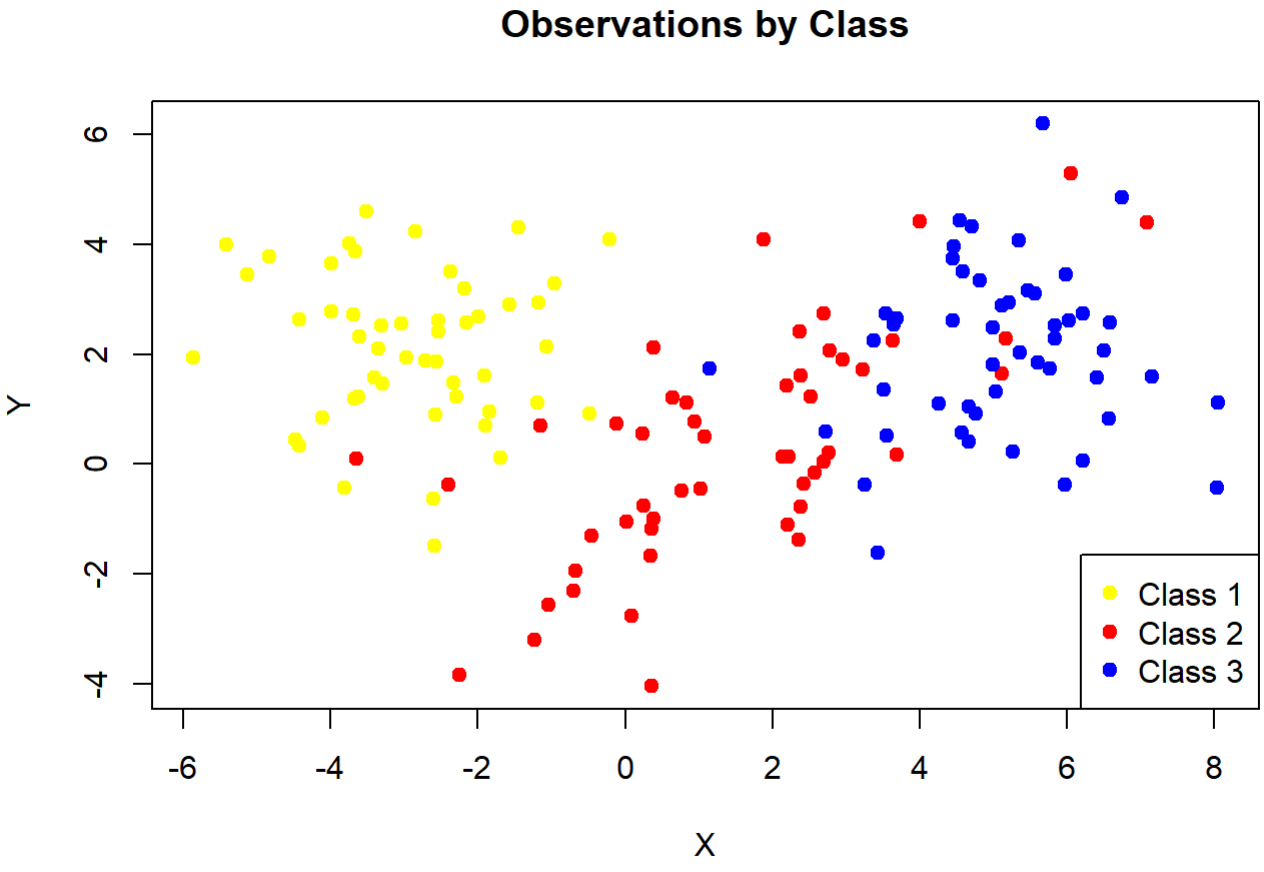
a. First, generating 50 data points for each class $k = 1, 2, 3$ that follow a $N(\mu_k, \sum_k)$ where $\mu_1 = [-3, 2]$, $\mu_2 = [1, 0]$, $\mu_1 = [5, 2]$, and $\sum_1 = \sum_3 =$

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

and $\sum_2 =$

$$\begin{bmatrix} 4 & 3 \\ 3 & 5 \end{bmatrix}$$

. Then plotting all the points on the same axes.



b. Then fitting a linear discriminant analysis model (LDA) and creating a confusion matrix.

```
##
##      1  2  3
##   1 49  3  0
##   2  1 39  4
##   3  0  8 46
```

```
## [1] 0.1066667
```

Here we can see the 3 by 3 confusion matrix for our LDA model and we can see that our training error is about 0.11.

c. Then fitting a Quadratic Discriminant analysis model (QDA) and creating a confusion matrix.

```
##
##      1  2  3
##   1 48  3  0
##   2  2 40  4
##   3  0  7 46
```

```
## [1] 0.1066667
```

Here is the 3 by 3 confusion matrix for our QDA model and we can see that our training error is about 0.11, which is the same as the LDA model.

d. Now generating 500 test observations for each class using the same procedure and reporting the error using LDA and QDA.

```
## [1] 0.1026667
```

```
## [1] 0.09533333
```

In simulating a larger data set, we can see the new training error rate for LDA is about 0.1 and is about 0.1 for the QDA model. We can see that the error rates are fairly similar, but we would prefer the QDA model because it is slightly lower.

e. The key difference between LDA and QDA is their assumption about the variance between the predictors. LDA takes the assumption that the covariance matrix is the same for all classes, whereas QDA assumes different covariance matrices for each predictor. Thus LDA is simpler and can be more restrictive in comparison to QDA. We also know that LDA estimates a single variance matrix and coefficients for each predictor variable. QDA on the other hand estimates a separate covariance matrix for each class as well as the coefficients. Meaning the QDA has more parameters to estimate than LDA. In conclusion, I would say that LDA is a more generalizable model compared to QDA. LDA tries to estimate less coefficients and matrices and takes less assumptions. Additionally, it may help reduce over fitting in the training error.

f. Based on the generation process in this question and the results that we found, LDA and QDA were very similar. We saw that the error rate for QDA was slightly lower than the LDA, but this is not a big enough difference to justify QDA over LDA. Since the two different model were so similar in predicting, I will stand

with the conclusion made in part e in which the LDA model is preferred. This is because even though it preforms just slightly worse, it is simpler and for the other reasons stated in part e.

Question 4

Intro) In this question we will work out a Bayes classifier for a binary classification problem. WE have data (X, Y) where X can be any real number and Y is from $1, 2$. We assume the that observations X from class $Y = 1$ are drawn from $Norm(\mu, \sigma^2)$ and class $Y = 2$ are drawn from a $Unif(0, \theta)$. We also assume that the prior probability of a random observation belonging to class 1 is $P(Y = 1) = \pi^1$.

a. From our givens and PDF of a normal and uniform, we know that

$$P(X = x|Y = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

$$P(X = x|Y = 2) = \frac{1}{\theta}$$

For $0 \leq x < \theta$, and 0 otherwise.

Then finding we want to write $P(X = x|Y = 1)$ and $P(X = x|Y = 2)$.

$$\begin{aligned} P(X = x|Y = 1) &= (usingBayesRule) \\ &= \frac{P(X = x|Y = 1) \cdot P(Y = 1)}{P(X = x)} \\ &\text{based on our prior probability and the law of total probability} \\ &= \frac{P(X = x|Y = 1) \cdot \pi^1}{P(X = x)} \\ &= \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \cdot \pi}{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \cdot \pi + \frac{1}{\theta} \cdot (1 - \pi)} \end{aligned}$$

And without loss of generality,

$$\begin{aligned} P(X = x|Y = 2) &= (usingBayesRule) \\ &= \frac{P(X = x|Y = 2) \cdot P(Y = 2)}{P(X = x)} \\ &\text{based on our prior probability} \\ &= \frac{P(X = x|Y = 2) \cdot (1 - \pi^1)}{P(X = x)} \\ &= \frac{\frac{1}{\theta} \cdot (1 - \pi^1)}{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \cdot \pi + \frac{1}{\theta} \cdot (1 - \pi)} \end{aligned}$$

b. Now finding an expression for the Bayes decision boundary. We can do this by setting $P(Y = 1|X = x) = P(Y = 2|X = x)$ and simplifying.

$$\frac{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot \pi}{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot \pi + \frac{1}{\theta} \cdot (1-\pi)} = \frac{\frac{1}{\theta} \cdot (1-\pi)}{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot \pi + \frac{1}{\theta} \cdot (1-\pi)}$$

$$\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot \pi = \frac{1}{\theta} \cdot (1-\pi)$$

$$-\frac{(x-\mu)^2}{2\sigma^2} = ln(\frac{\sqrt{2\pi\sigma^2}}{\pi\theta} \cdot (1-\pi))$$

$$-\frac{(x-\mu)^2}{2\sigma^2} = ln(\frac{\sqrt{2\pi\sigma^2}}{\pi\theta} \cdot (1-\pi))$$

$$(x-\mu) = \pm\sqrt{-2\sigma^2 ln(\frac{\sqrt{2\pi\sigma^2}}{\pi\theta} \cdot (1-\pi))}$$

$$x = \mu \pm \sqrt{-2\sigma^2 ln(\frac{\sqrt{2\pi\sigma^2}}{\pi\theta} \cdot (1-\pi))}$$

c. Now suppose that we know $\mu = 0$, $\sigma^2 = 1$, $\theta = 1$, and $\pi = 0.5$, then using the Bayes decision boundary we found in part b we can find the range of x-values that will get assigned to class 1.

First substituting in our values into the Bayes decision boundary found in part b,

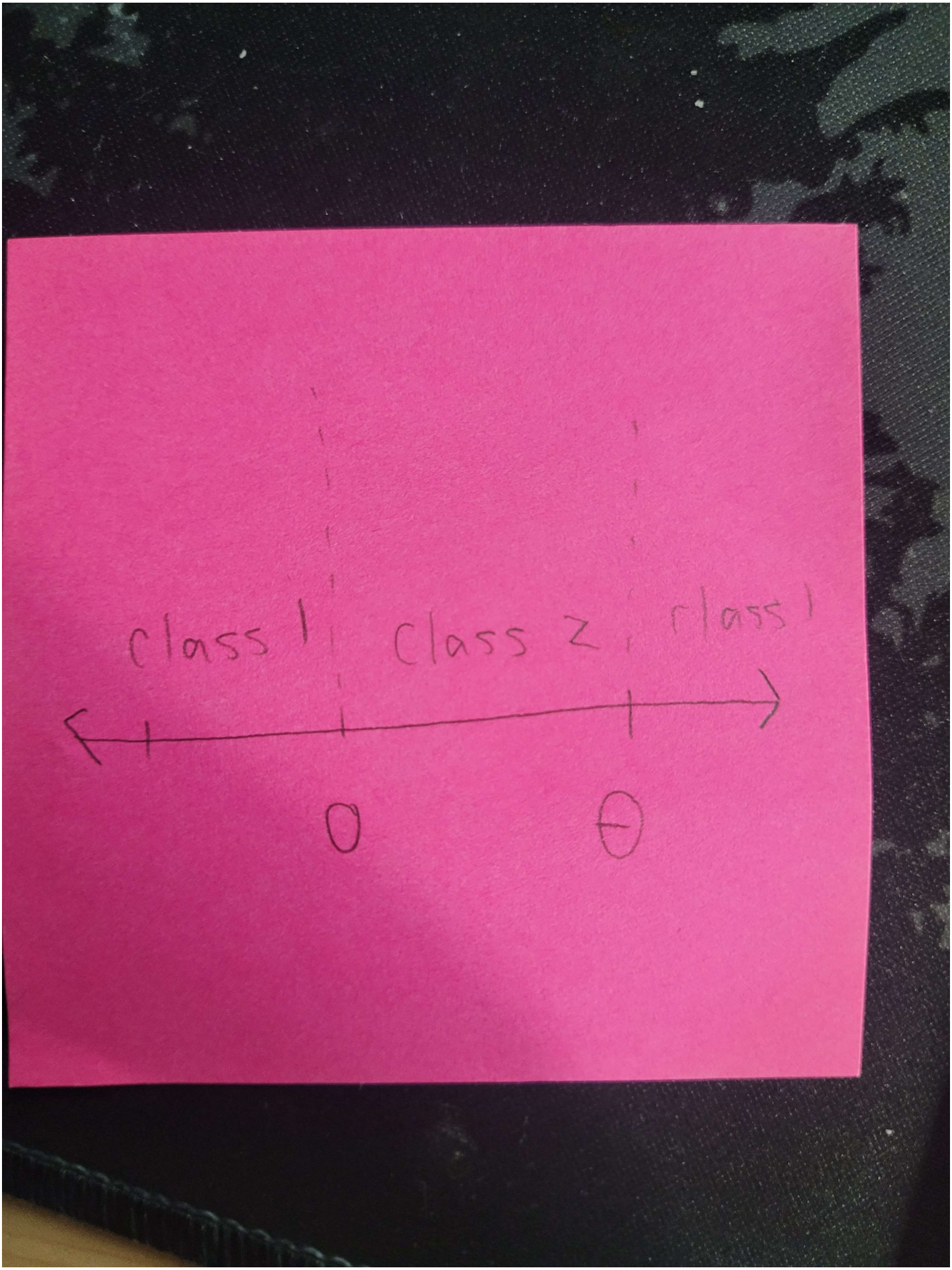
$$x = \mu \pm \sqrt{-2\sigma^2 ln(\frac{\sqrt{2\pi\sigma^2}}{\pi\theta} \cdot (1-\pi))}$$

$$x = (0) \pm \sqrt{-2(1)^2 ln(\frac{\sqrt{2(0.5)(1)^2}}{(0.5)(1)} \cdot (1-(0.5)))}$$

$$x = \pm\sqrt{-2ln(\frac{1}{0.5} \cdot (0.5))}$$

$$x = 0$$

This tells us that the values for x that will be assigned to class 1 are from 0 to θ. Hence we can draw the real line describing x and it's classification as,


class 1
Class 2
class 1
0
θ

d. Now finding an expression to estimate $\pi = P(Y = 1)$ from the data.

$$\frac{\sum_{i=1}^{n} I(Y_i = 1)}{n}$$

where the numerator is the sum of the indicator functions that would count the number of observations in class 1, and the denominator is the number of observations in the data.

e. Then assuming that $X_1, \ldots, X_n \sim Norm(\mu, \sigma^2)$ are independent and identically distributed. We can estimate the mean μ and variance σ^2 as,

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$\hat{\sigma^2} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

f. Then assuming that $X_1, \ldots, X_n \sim Unif(0, \theta)$ are independent and identically distributed, we can estimate θ as,

$$\hat{\theta} = X_i$$

Where X_i is the maximum value in the sample. We know this because in thee Uniform distribution, the max value is a consistent estimator of θ.

g. Finally, suppose that the predicted classification intervals are,

$$\hat{y} = 1, x \in (\infty, 10) \cup [35, 50] \cup (100, \infty)$$

$$\hat{y} = 2, x \in [10, 35] \cup (50, 100]$$

And the true classification intervals are

$$\hat{y} = 1, x \in (\infty, 10) \cup (20, 50] \cup (120, \infty)$$

$$\hat{y} = 2, \in [10, 20] \cup (50, 120]$$

Our goal is to identify intervals that will be classifieds as class 1 and 2 by the estimated classifier.

From these intervals, we can see that the estimated classifier will classify any value of x within $(100, 120]$ will be misclassified as class 1 by the estimated classifier when it reality it should be class 2. Additionally, any x values within the interval $(20, 35]$ will be misclassified as class 2 by the estimated classifier when they should be class 1.

Problem 5

Intro) Assume the logistic model $log\frac{p}{1-p} = mx$ and $Y|X = x \sim Bernoulli(p)$.

a. First we want to show that the probability density function of $Y|X = x$ takes the form $f(y|x) = exp(\frac{y\theta - b\theta}{\alpha} + c(y, \alpha))$.

First solving for p in our logistic model,

$$\begin{aligned}\log\frac{p}{1-p} &= mx \\ \frac{p}{1-p} &= e^{mx} \\ p &= (1-p)e^{mx} \\ p + pe^{mx} &= e^{mx} \\ p &= \frac{e^{mx}}{1+e^{mx}}\end{aligned}$$

Then using the fact that the PDF of a Bernoulli random variable is $P(Y=y|X=x) =^x (1-p)^{1-x}$ and substituting in our found p,

$$P(Y=y|X=x) = \frac{e^{mx}}{1+e^{mx}}^{y}(1-\frac{e^{mx}}{1+e^{mx}})^{1-y}$$

Thus taking ln of both sides,

$$\begin{aligned}ln(P(Y=y|X=x)) &= yln(\frac{e^{mx}}{1+e^{mx}}) + (1-y)ln(1-\frac{e^{mx}}{1+e^{mx}}) \\ &= ymx - yln(1+e^{mx}) - ln(1+e^{mx}) + yln(1+e^{mx}) \\ &= ymx - ln(1+e^{mx})\end{aligned}$$

Then taking the exponential of both sides,

$$\begin{aligned}exp(ln(P(Y=y|X=x)) &= exp(ymx - ln(1+e^{mx})) \\ (P(Y=y|X=x) &= exp(ymx - ln(1+e^{mx}))\end{aligned}$$

Comparing this to the form of $f(y|x) = exp(\frac{y\theta - b\theta}{\alpha} + c(y,\alpha))$, we know that our corresponding values are

$$\alpha = 1, \theta = mx, y = y, b(\theta) = ln(1+e^{mx}), c(y,\alpha) = 0$$

b. Now we want to show that $E[Y|X=x] = b'(\theta)$, or equivalently, $\theta = (b')^{-1}(E[Y|X=x])$. From what we know about X and expectation of Bernoulli random variables,

$$E[Y|X=x] = p = \frac{e^{mx}}{1+e^{mx}}$$

Additionally,

$$b'(\theta) = \frac{d}{d\theta}ln(1+e^{mx}) = \frac{1}{1+e^{mx}} \times e^{mx}$$

Hence $E[Y|X=x] = b'(\theta)$.

c. Finally, we want to show that $Var(Y|X=x) = ab''(\theta)$. From what we know about X and varience of Bernoulli random variables,

$$\begin{aligned} E[Y|X=x] = p(1-p) &= \frac{e^{mx}}{1+e^{mx}} \times (1 - \frac{e^{mx}}{1+e^{mx}}) \\ &= \frac{e^{mx}}{1+e^{mx}} \times \frac{1+e^{mx}-e^{mx}}{1+e^{mx}} \\ &= \frac{e^{mx}}{(1+e^{mx})^2} \\ &\text{since } \theta = mx \\ &= \frac{e^{\theta}}{(1+e^{\theta})^2} \end{aligned}$$

Additionally,

$$\begin{aligned} \alpha b''(\theta) &= 1 \times \frac{d^2}{d^2\theta} ln(1+e^{mx}) \\ &= \frac{d}{d\theta} \frac{1}{1+e^{mx}} \times e^{mx} \\ &= \frac{d}{d\theta} \frac{1}{1+e^{\theta}} \times e^{\theta} \\ &= \frac{e^{\theta}}{(1+e^{\theta})^2} \end{aligned}$$

Hence $Var(Y|X=x) = ab''(\theta)$.

Appendix

- Problem 3 part a
- Problem 3 part b
- Problem 3 part c
- Problem 3 part d
- Problem 3 part e