

Linga Sai Yuvesh Venketa Kotiala

16242113

ISL HW-4

1. (a) Use LDA to build a classifier for the following data. To get full credit you must show all work and all discriminant values.

1. >

(a) grouping them to classes 1 and 0.

| X_1 | Y_1 | Y |
|---------|-------|-----|
| 4 | 1 | 0 |
| -11 | -4 | 0 |
| 8 | 1 | 0 |
| -6 | -3 | 0 |
| 0 | -7 | 0 |
| -3 | 5 | 0 |
| 0 | 3 | 0 |
| class 0 | | |
| 6 | 3 | 1 |
| -3 | 7 | 1 |
| 4 | 12 | 1 |
| 10 | 11 | 1 |
| 5 | 5 | 1 |
| -7 | 8 | 1 |
| 13 | 8 | 1 |
| 5 | 8 | 1 |
| class 1 | | |

solution

$$X_1 = \begin{bmatrix} 4 & 1 \\ -11 & -4 \\ 8 & 1 \\ -6 & -3 \\ 0 & -7 \\ -3 & 5 \\ 0 & 3 \end{bmatrix}$$

$$\text{mean } \mu_1 = [-1.1428 \quad -0.5714]$$

②

$$\underline{X_2} = \begin{bmatrix} 6 & 3 \\ -3 & 7 \\ 4 & 12 \\ 10 & 11 \\ 5 & 5 \\ -7 & 8 \\ 13 & 8 \\ 5 & 8 \end{bmatrix}$$

$$\text{mean } \mu_2 = [4.125 \quad 7.75]$$

g = number of groups in y in our case it is 2 [class 0 & class 1]

$$\mu = \text{it is mean of whole set} = [1.4911 \quad 3.5893] \cong [1.5 \quad 3.6]$$

$$X_1^0 = \begin{bmatrix} \frac{40-15}{10} & \frac{10-15}{10} \\ \frac{-10-15}{10} & \frac{-40-15}{10} \\ \frac{80-15}{10} & \frac{10-15}{10} \\ \frac{-60-15}{10} & \frac{-30-15}{10} \\ \frac{-15}{10} & \frac{-70-15}{10} \\ \frac{-30-15}{10} & \frac{80-15}{10} \\ \frac{-15}{10} & \frac{30-15}{10} \end{bmatrix} = \begin{bmatrix} 2.5 & -0.5 \\ -12.5 & -5.5 \\ 6.5 & -0.5 \\ -7.5 & -4.5 \\ -1.5 & -8.5 \\ -4.5 & 3.5 \\ -1.5 & 1.5 \end{bmatrix}$$

$$X_2^0 = \begin{bmatrix} \frac{60-36}{10} & \frac{30-36}{10} \\ \frac{-30-36}{10} & \frac{70-36}{10} \\ \frac{40-36}{10} & \frac{120-36}{10} \\ \frac{100-36}{10} & \frac{110-36}{10} \\ \frac{50-36}{10} & \frac{50-36}{10} \\ \frac{-70-36}{10} & \frac{80-36}{10} \\ \frac{130-36}{10} & \frac{80-36}{10} \\ \frac{50-36}{10} & \frac{80-36}{10} \end{bmatrix} = \begin{bmatrix} 2.4 & -0.6 \\ -6.6 & 3.4 \\ 0.4 & 9.4 \\ 6.4 & 7.4 \\ 1.4 & 1.4 \\ -10.6 & 4.4 \\ 9.4 & 4.4 \\ 1.4 & 4.4 \end{bmatrix}$$

$$C_i = \frac{(X_i^0)^T (X_i^0)}{n_i} \quad \text{This we calculated in the R-programing.}$$

Mean of the Matrix [M-mean]

$$D = M - M\text{-mean}$$

$$C = (n-1)^{-1} \times D^T \times D$$

$$C_1 = \begin{bmatrix} 39.47 & 9.07 \\ 9.07 & 17.95 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 41.8392 & 1.7321 \\ 1.7321 & 9.8392 \end{bmatrix}$$

C1

```
> #create vectors -- these will be our columns
> a <- c(2.5,-12.5,6.5,-7.5,-1.5,-4.5,-1.5)
> b <- c(-0.5,-5.5,-0.5,-4.5,-8.5,3.5,1.5)
>
>
> #create matrix from vectors
> M <- cbind(a,b)
>
> k <- ncol(M) #number of variables
> n <- nrow(M) #number of subjects
>
> #create means for each column
> M_mean <- matrix(data=1, nrow=n) %%% cbind(mean(a),mean(b))
>
> #creates a difference matrix
> D <- M - M_mean
>
> #creates the covariance matrix
> C <- ((7-1)^-1)* t(D) %%% D
> C
      a      b
a 39.476190  9.071429
b  9.071429 17.952381
>
> ## b using the covariance function
> cov(M)
      a      b
a 39.476190  9.071429
b  9.071429 17.952381
> |
```

C2

```
> #create vectors -- these will be our columns
> a <- c(2.4,-6.6,0.4,6.4,1.4,-10.6,9.4,1.4)
> b <- c(-0.6,3.4,9.4,7.4,1.4,4.4,4.4,4.4)
>
>
> #create matrix from vectors
> M <- cbind(a,b)
>
> k <- ncol(M) #number of variables
> n <- nrow(M) #number of subjects
>
> #create means for each column
> M_mean <- matrix(data=1, nrow=n) %%% cbind(mean(a),mean(b))
>
> #creates a difference matrix
> D <- M - M_mean
>
> #creates the covariance matrix
> C <- ((8-1)^-1)* t(D) %%% D
> C
      a      b
a 41.839286  1.732143
b  1.732143  9.839286
>
> ## b using the covariance function
> cov(M)
      a      b
a 41.839286  1.732143
b  1.732143  9.839286
> |
```

$$C(x, s) = \frac{1}{n} \sum_{i=1}^g n_i \cdot C_i(x, s) \quad (4)$$

$$= \begin{bmatrix} \frac{7}{15} (39.47) + \frac{8}{15} (41.8392) & \frac{7}{15} (9.07) + \frac{8}{15} (1.732) \\ \frac{7}{15} (9.07) + \frac{8}{15} (1.732) & \frac{7}{15} (17.95) + \frac{8}{15} (9.8372) \end{bmatrix}$$

$$C = \begin{bmatrix} 40.73 & 5.16 \\ 5.16 & 8.37 \end{bmatrix}$$

$$P = \begin{bmatrix} 7/15 \\ 8/15 \end{bmatrix} = \begin{bmatrix} 0.467 \\ 0.533 \end{bmatrix}$$

we can get inverse function by $\text{ginv}()$ function which is present in 'MASS' library.

$$C^{-1} = \frac{1}{40.73 \times 8.37 - 5.16 \times 5.16} \begin{bmatrix} 8.37 & -5.16 \\ -5.16 & 40.73 \end{bmatrix} = \begin{bmatrix} 0.0265 & -0.0157 \\ -0.0157 & 0.1238 \end{bmatrix}$$

Discriminant function:-

$$f_i = \mu_i C^{-1} x_k^T - \frac{1}{2} \mu_i C^{-1} \mu_i^T + \ln(p_i)$$

let us compare the sample $[4 \ 1] = x_k$, which is of class 0.

$$f_1 = [-1.1428 \ -0.5714] \begin{bmatrix} 0.0265 & -0.0157 \\ -0.0157 & 0.1238 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} - \left(\frac{1}{2}\right) [-1.1428 \ -0.5714] \begin{bmatrix} -1.1428 & -0.5714 \\ -0.5714 & -0.5714 \end{bmatrix} \begin{bmatrix} -1.1428 \\ -0.5714 \end{bmatrix} + \ln(p_1)$$

$$\begin{bmatrix} 0.0265 & -0.0157 \\ -0.0157 & 0.1238 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} - \left(\frac{1}{2}\right) \begin{bmatrix} -1.1428 \\ -0.5714 \end{bmatrix} + \ln(p_1)$$

$$f_1 = -1.1112$$

$$f_2 = 0.0055$$

[4,1] class0

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(4,1),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2)
>
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     4     1
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.111218
```

$F_1 = -1.1112$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(4,1),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> u1 <- matrix(data=c(-1.142, -0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     4     1
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 0.005528841
```

$F_2 = 0.0055$

[-11, -4] class0

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-11,-4),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]  -11   -4
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -2.219538
```

$F_1 = -2.2195$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-11,-4),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]  -11   -4
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -5.974221
```

$F_2 = -5.9742$

[8,1] class0

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(8,1),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     8     1
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.769586
```

$F_1 = -1.769586$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(8,1),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     8     1
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -0.6171212
```

$F_2 = -0.6171$

[-6, -3] class0

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-6,-3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    -6    -3
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.227458
```

$F_1 = -1.227458$

```
> #create vectors -- these will be our column:
> a <- matrix(data=c(-6,-3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    -6    -3
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -3.989996
```

$F_2 = -3.989996$

[0, -7] class0

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(0,-7),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    0   -7
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -3.425571
```

$F_1 = -3.425571$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(0,-7),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    0   -7
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -9.924521
```

$F_2 = -9.924521$

[-3,5] class0

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-3,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -3    5
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -2.864564
```

F₁= -2.864564

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-3,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -3    5
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.979666
```

F₂= 1.979666

[0,3] class0

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(0,3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    0    3
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.47767
```

$F_1 = -1.47767$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(0,3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    0    3
> x <- a %%% M %%% t(d)
>
> y <- 0.5 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.498354
```

$F_2 = 1.498354$

[6,3] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(6,3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     6     3
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.799822
```

$F_1 = -1.799822$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(6,3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     6     3
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.229779
```

$F_2 = 1.229779$

[-3,7] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-3,7),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -3    7
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -4.549983
```

$F_1 = -4.549983$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-3,7),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -3    7
> x <- a %*% M %*% t(d)
>
> y <- 0.5 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,]  2.189241
```

$F_2 = 2.189241$

[4,12] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(4,12),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    4   12
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -9.853027
```

$F_1 = -9.853027$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(4,12),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    4   12
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.686191
```

$F_2 = 1.686191$

[10,11] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(10,11),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   10   11
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -8.643869
```

$F_1 = -8.643869$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(10,11),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   10   11
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 2.001429
```

$F_2 = 2.001429$

[5,5] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(5,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     5     5
> x <- a %**% M %**% t(d)
>
> y <- 1/2 * (a %**% M %**% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -2.6189
```

$F_1 = -2.6189$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(5,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     5     5
> x <- a %**% M %**% t(d)
>
> y <- 1/2 * (a %**% M %**% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 2.296766
```

$F_2 = 2.296766$

[-7,8] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-7,8),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    -7     8
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -6.525625
```

$F_1 = -6.525625$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-7,8),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    -7     8
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.125379
```

$F_2 = 1.125379$

[13,8] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(13,8),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    13     8
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -6.029466
```

$F_1 = -6.029466$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(13,8),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    13     8
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.800129
```

$F_2 = 1.800129$

[5,8] class1

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(5,8),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     5     8
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -4.95593
```

$F_1 = -4.95593$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(5,8),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     5     8
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,]  2.802229
```

$F_2 = 2.802229$

(b)

[1,2]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(1,2),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157 0.1238
> a
      [,1] [,2]
[1,]     1     2
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.118502
```

$F_1 = -1.118502$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(1,2),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157 0.1238
> a
      [,1] [,2]
[1,]     1     2
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 0.9189538
```

$F_2 = 0.9189538$

[1,4]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(1,4),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     1     4
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.935522
```

$F_1 = -1.935522$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(1,4),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     1     4
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.996929
```

$F_2 = 1.996929$

[-6,6]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-6,6),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -6    6
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -4.221847
```

F₁= -4.221847

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-6,6),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -6    6
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.543091
```

F₂= 1.543091

[-4, -1]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-4,-1),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -4   -1
> x <- a %**% M %**% t(d)
>
> y <- 1/2 * (a %**% M %**% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -0.8352621
```

$F_1 = -0.8352621$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-4,-1),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -4   -1
> x <- a %**% M %**% t(d)
>
> y <- 1/2 * (a %**% M %**% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -1.684946
```

$F_2 = -1.684946$

[3,5]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(3,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     3     5
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -2.521316
```

$F_1 = -2.521316$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(3,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     3     5
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,]  2.376491
```

$F_2 = 2.376491$

[0,5]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(0,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     0     5
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -2.57369
```

$F_1 = -2.57369$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(0,5),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     0     5
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,]  2.297329
```

$F_2 = 2.297329$

[6,3]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(6,3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     6     3
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.799822
```

$F_1 = -1.799822$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(6,3),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]     6     3
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] 1.229779
```

$F_2 = 1.229779$

[-1,-2]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-1,-2),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    -1    -2
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -0.8646782
`|`
```

$F_1 = -0.8646782$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-1,-2),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    -1    -2
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -2.635071
`|`
```

$F_2 = -2.635071$

[15, -4]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(15,-4),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    15    -4
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -9.306446
> |
```

$F_1 = -9.306446$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(15,-4),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]    15    -4
> x <- a %%% M %%% t(d)
>
> y <- 1/2 * (a %%% M %%% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -5.783931
```

$F_2 = -5.783931$

[-8,-2]

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-8,-2),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> #d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -8   -2
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(7/15)
      [,1]
[1,] -1.330584
```

$F_1 = -1.330584$

```
> #create vectors -- these will be our columns
> a <- matrix(data=c(-8,-2),1,2)
> b <- c(0.0265,-0.0157)
> c <- c(-0.0157,0.1238)
> #d <- matrix(data=c(-1.142,-0.5714),1,2) # u1
> d <- matrix(data=c(4.125,7.75),1,2) #u2
> #create matrix from vectors
> M <- cbind(b,c)
> M
      b      c
[1,] 0.0265 -0.0157
[2,] -0.0157  0.1238
> a
      [,1] [,2]
[1,]   -8   -2
> x <- a %*% M %*% t(d)
>
> y <- 1/2 * (a %*% M %*% t(a) )
>
> x-y+log(8/15)
      [,1]
[1,] -3.163484
```

$F_2 = -3.163484$

Score Calculation [used similarly from excel sheet]

$$M_1 - M_2 = \begin{bmatrix} -5.2678 \\ -8.3214 \end{bmatrix}$$

$$M_1 + M_2 = \begin{bmatrix} 2.9822 & 7.1786 \end{bmatrix}$$

$$W_0 = -\ln \left(\frac{0.467(p_1)}{0.533(p_2)} \right) - \frac{1}{2} \begin{bmatrix} 2.9822 & 7.1786 \end{bmatrix} \begin{bmatrix} -5.2678 \\ -8.3214 \end{bmatrix}$$

$$= 0.1322 - \frac{1}{2} [-15.7096 - 59.736]$$

$$= 37.8550.$$

$$W = C^{-1} \times [M_1 - M_2]$$

$$= \begin{bmatrix} 0.0265 & -0.0857 \\ -0.0157 & 0.1238 \end{bmatrix} \begin{bmatrix} -5.2678 \\ -8.3214 \end{bmatrix} = \begin{bmatrix} -0.009 \\ -0.9475 \end{bmatrix}$$

$$\text{Score} = x_k^T \times W^T + W_0$$

$$= \begin{bmatrix} 4 \\ 1 \end{bmatrix} \begin{bmatrix} -0.009 & -0.9475 \end{bmatrix} + 37.8550$$

$$= 36.8715.$$


```

> a <- matrix(data = c(4,1),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 36.8715
> a <- matrix(data = c(-11,-4),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 41.744
> a <- matrix(data = c(6,3),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 34.9585
> a <- matrix(data = c(8,1),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 36.8355
> a <- matrix(data = c(-6,-3),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 40.7515
> a <- matrix(data = c(0,-7),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 44.4875

```

```

> a <- matrix(data = c(-3,5),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 33.1445
> a <- matrix(data = c(0,3),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 35.0125
> a <- matrix(data = c(6,3),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 34.9585
> a <- matrix(data = c(-3,7),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 31.2495
> a <- matrix(data = c(4,12),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 26.449
> a <- matrix(data = c(10,11),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 27.3425

```

```

> a <- matrix(data = c(5,5),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 33.0725
> a <- matrix(data = c(-7,8),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 30.338
> a <- matrix(data = c(13,8),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 30.158
> a <- matrix(data = c(5,8),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 30.23

```

| X ₁ | X ₂ | Y | F ₁ | F ₂ | Score |
|----------------|----------------|---|----------------|----------------|---------|
| 4 | 1 | 0 | -1.1112 | 0.0055 | 36.8715 |
| -11 | -4 | 0 | -2.2195 | -5.9742 | 41.744 |
| 8 | 1 | 0 | -1.769586 | -0.6171 | 36.8355 |
| -6 | -3 | 0 | -1.227458 | -3.989996 | 40.7515 |
| 0 | -7 | 0 | -3.425571 | -9.924521 | 44.4875 |
| -3 | 5 | 0 | -2.864564 | 1.979666 | 35.1445 |
| 0 | 3 | 0 | -1.47767 | 1.498354 | 35.0125 |
| | | | | | |
| 6 | 3 | 1 | -1.799822 | 1.229779 | 34.9585 |
| -3 | 7 | 1 | -4.549983 | 2.189241 | 31.2495 |
| 4 | 12 | 1 | -9.853027 | 1.686191 | 26.449 |
| 10 | 11 | 1 | -8.643869 | 2.001429 | 27.3425 |
| 5 | 5 | 1 | -2.6189 | 2.296766 | 33.0725 |
| -7 | 8 | 1 | -6.525625 | 1.125379 | 30.338 |
| 13 | 8 | 1 | -6.029466 | 1.800129 | 30.158 |
| 5 | 8 | 1 | -4.95593 | 2.802229 | 30.23 |

So by this we can assume that all the values of class 0 have the score above 35 (class0 if(score>35)) and we see that if the scores are below 35 then it is class1

1.(b)

Ans.

```
> a <- matrix(data = c(1,2),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 35.951
> a <- matrix(data = c(1,4),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 34.056
> a <- matrix(data = c(-6,6),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 32.224
> a <- matrix(data = c(-4,-1),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 38.8385
> a <- matrix(data = c(3,5),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 33.0905
> a <- matrix(data = c(0,5),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 33.1175
```

```

> a <- matrix(data = c(6,3),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 34.9585
> a <- matrix(data = c(-1,-2),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 39.759
> a <- matrix(data = c(15,-4),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 41.51
> a <- matrix(data = c(-8,-2),2,1)
> b <- matrix(data = c(-0.009,-0.9475),1,2)
> c <- (t(a) %*% t(b))
> c+ 37.8550
      [,1]
[1,] 39.822

```

| X ₁ | X ₂ | F ₁ | F ₂ | score | Expected Y |
|----------------|----------------|----------------|----------------|---------|------------|
| 1 | 2 | -1.118502 | 0.9189538 | 35.951 | 0 |
| 1 | 4 | -1.935522 | 1.996929 | 34.056 | 1 |
| -6 | 6 | -4.221847 | 1.543091 | 32.224 | 1 |
| -4 | -1 | -0.8352621 | -1.684946 | 38.8385 | 0 |
| 3 | 5 | -2.521316 | 2.376491 | 33.0905 | 1 |
| 0 | 5 | -2.57369 | 2.297329 | 33.1175 | 1 |
| 6 | 3 | -1.799822 | 1.229779 | 34.9585 | 1 |
| -1 | -2 | -0.8646782 | -2.635071 | 39.759 | 0 |
| 15 | -4 | -9.306446 | -5.783931 | 41.51 | 0 |
| -8 | -2 | -1.330584 | -3.163484 | 39.822 | 0 |

2. Using basic statistical properties of the variance, as well as single variable calculus, prove that given by equation (5.6) in the text book does indeed minimize $\text{Var}(\alpha X + (1-\alpha)Y)$.
Ans.

2.7

We know that

$$\text{Var}(\alpha X + (1-\alpha)Y) = \alpha^2 \sigma_x^2 + (1-\alpha)^2 \sigma_y^2 + 2\alpha(1-\alpha)\sigma_{xy}$$

We now take the first derivative of $\text{Var}(\alpha X + (1-\alpha)Y)$ relative to α and we get.

$$\frac{\partial}{\partial \alpha} \text{Var}(\alpha X + (1-\alpha)Y) = 2\alpha \sigma_x^2 - 2\sigma_y^2 + 2\sigma_{xy} - 2\sigma_{xy} = 2\alpha \sigma_x^2 - 2\sigma_y^2 + 2\sigma_{xy}$$

We now seek critical points by equalling the last expression to 0.

$$2\alpha \sigma_x^2 - 2\sigma_y^2 + 2\sigma_{xy} = 0$$

Which implies that.

$$\alpha = \frac{\sigma_y^2 - \sigma_{xy}}{\sigma_x^2 + \sigma_y^2 - 2\sigma_{xy}}$$

It remains to check that this point is in fact a minimum.

This is equivalent to prove that the second derivative is positive.

$$\frac{\partial^2}{\partial \alpha^2} \text{Var}(\alpha X + (1-\alpha)Y) = 2\sigma_x^2 = 2\text{Var}(X) \geq 0$$

3. We will now derive the probability that a given observation is part of a bootstrap sample. Suppose that we obtain a bootstrap sample from a set of n observations.

(a) What is the probability that the first bootstrap observation is not the j th observation from the original sample? Justify your answer.

Ans.

There are n observations in the sample since bootstrap sampling draws items with replacement, we are sampling the data from the same pool with some probability every time. Those are $(n-1)$ items in the n observations that are not in j hence the probability is

$(n-1)/n$ which can be written as :

$$1 - 1/n.$$

(b) What is the probability that the second bootstrap observation is not the j th observation from the original sample?

Ans.

Since we draw with replacement, it is same as the one bootstrapping sample:

$$1 - 1/n.$$

(c) Argue that the probability that the j th observation is not in the bootstrap sample is

$$(1 - 1/n)^n.$$

Ans.

As bootstrapping sample with replacement, we have that the probability that the j th observation is not in the bootstrap sample is the product of the probabilities that each bootstrap observation is not the j th observation from the original sample

$$(1 - 1/n) \cdots (1 - 1/n) = (1 - 1/n)^n$$

as these probabilities are independent.

(d) When $n=5$, what is the probability that the j th observation is in the bootstrap sample?

Ans.

We have

$$P(j\text{th obs in bootstrap sample}) = 1 - (1 - 1/5)^5 = 0.672.$$

(e) When $n=100$, what is the probability that the j th observation is in the bootstrap sample ?

Ans

We have

$$P(\text{jth obs in bootstrap sample}) = 1 - (1 - 1/100)^{100} = 0.634.$$

(f) When $n=10000$, what is the probability that the j th observation is in the bootstrap sample ?

Ans.

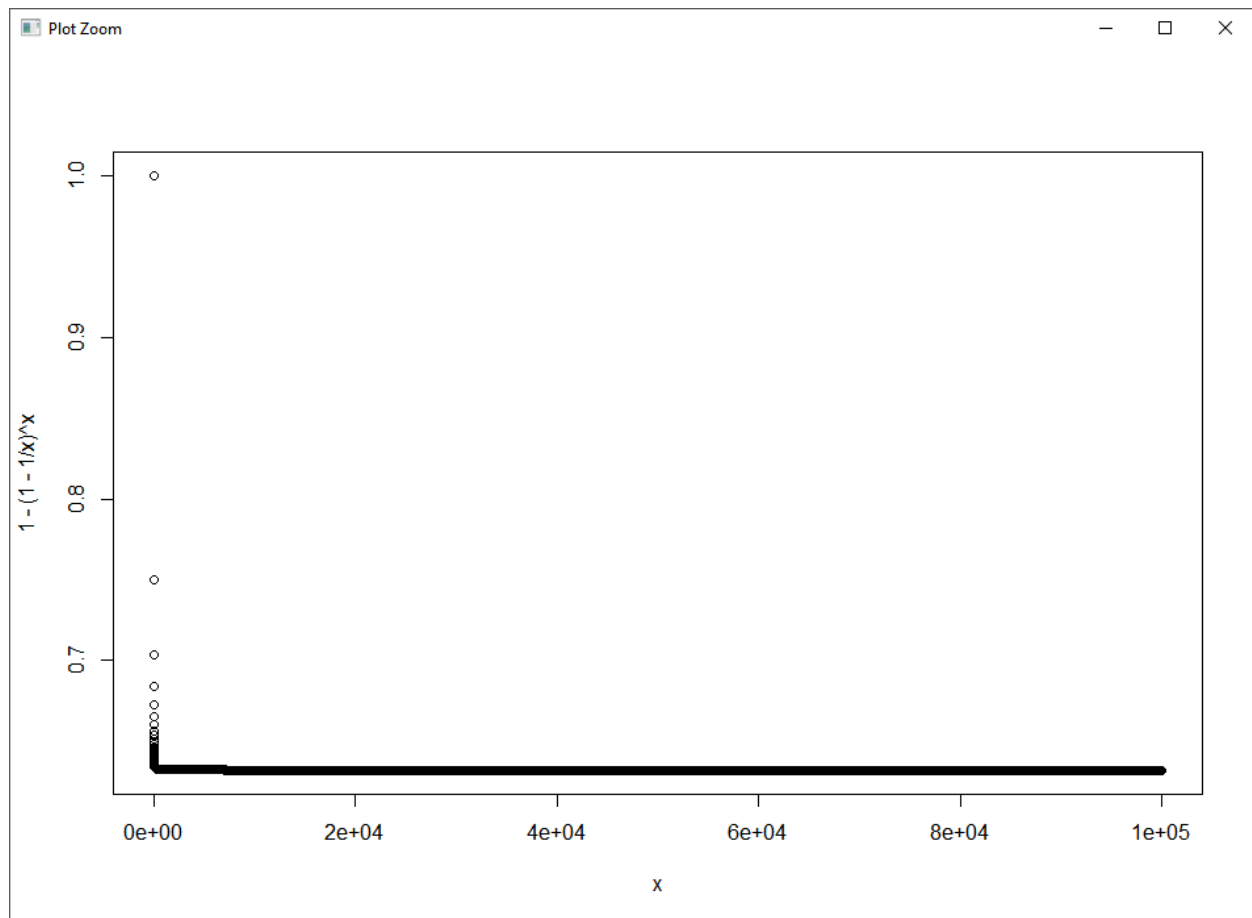
We have

$$P(\text{jth obs in bootstrap sample}) = 1 - (1 - 1/10000)^{10000} = 0.632.$$

(g) Create a plot that displays, for each integer value of n from 1 to 100000, the probability that the j th observation is in the bootstrap sample. Comment on what you observe.

Ans.

```
> x <- 1:100000  
> plot(x, 1 - (1 - 1/x)^x)
```

We may see that the plot quickly reaches an asymptote at about $0.6320.632$.

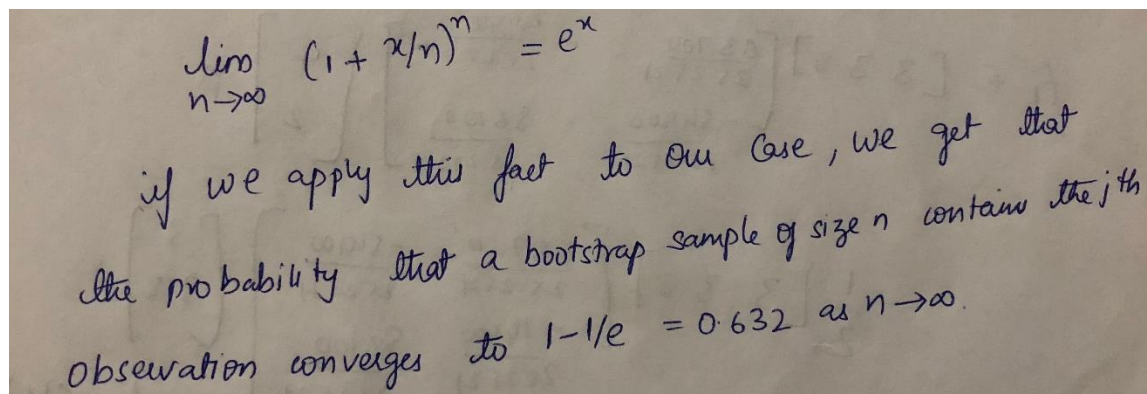
(h) We will now investigate numerically the probability that a bootstrap sample of size $n=100$ contains the j th observation. Here $j=4$. We repeatedly create bootstrap samples, and each time we record whether or not the fourth observation is contained in the bootstrap sample.

Ans.

```
> store <- rep(NA, 10000)
> for (i in 1:10000) {
+   store[i] <- sum(sample(1:100, rep = TRUE) == 4) > 0
+ }
> mean(store)
[1] 0.6472
```

Comment on the results obtained.

A known fact from calculus tells us that



4. We now review k-fold cross-validation.

(a) Explain how k-fold cross-validation is implemented.

Ans.

The k-fold cross validation is implemented by taking the n observations and randomly splitting it into k non-overlapping groups of length of (approximately) n/k . These groups act as a validation set, and the remainder (of length $n - n/k$) acts as a training set. The test error is then estimated by averaging the k resulting MSE estimates.

(b) What are the advantages and disadvantages of k-fold cross validation relative to:

i. The validation set approach?

Ans.

The validation set approach has two main drawbacks compared to k-fold cross-validation. First, the validation estimate of the test error rate can be highly variable (depending on precisely which observations are included in the training set and which observations are included in the validation set). Second, only a subset of the observations is used to fit the model. Since statistical methods tend to perform worse when trained on fewer observations, this suggests that the validation set error rate may tend to overestimate the test error rate for the model fit on the entire data set.

ii. LOOCV?

Ans.

The LOOCV cross-validation approach is a special case of k-fold cross-validation in which $k=n$. This approach has two drawbacks compared to k-fold cross-validation. First, it requires fitting the potentially computationally expensive model n times compared to k-fold cross-validation which requires the model to be fitted only k times. Second, the LOOCV cross-validation approach may give approximately unbiased estimates of the test error, since each training set contains $n-1$ observations; however, this approach has higher variance than k-fold cross-validation (since we are averaging the outputs of n fitted models trained on an almost identical set of observations, these outputs are highly correlated, and the mean of highly correlated quantities has higher variance than less correlated ones). So, there is a bias-

variance trade-off associated with the choice of k in k -fold cross-validation; typically using $k=5$ or $k=10$ yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

5. Suppose that we use some statistical learning method to make a prediction for the response Y for a value of the predictor X . Carefully describe how we might estimate the standard deviation of our prediction.

Ans.

We may estimate the standard deviation of our prediction by using the bootstrap method. In this case, rather than obtaining new independent data sets from the population and fitting our model on those data sets, we instead obtain repeated random samples from the original data set. In this case, we perform sampling with replacement BB times and then find the corresponding estimates and the standard deviation of those B .