STAT 32950 Assignment 5

Bin Yu

May 6, 2025

Question 1

(a)

Let the training data be

$$(x_1, y_1) = (0.3,\ 1), \quad (x_2, y_2) = (0.5,\ 1), \quad (x_3, y_3) = (0.7,\ 0).$$

1-NN

For a test point x, we compute the distances to each training point:

$$|x - 0.3|, \quad |x - 0.5|, \quad |x - 0.7|.$$

The decision boundaries occur where two distances are equal:

$$|x - 0.3| = |x - 0.5| \implies x = 0.4, \quad |x - 0.5| = |x - 0.7| \implies x = 0.6.$$

Hence

$$\hat{y}_{\text{1-NN}}(x) = \begin{cases} 1, & x < 0.6, \\ 0, & x > 0.6, \\ \text{tie between } (0.5, 1) \text{ and } (0.7, 0), & x = 0.6. \end{cases}$$

At $x = 0.6$, the two nearest neighbors $(0.5, 1)$ and $(0.7, 0)$ are equidistant. To break this tie, we can use the class-prior frequencies in the training set:

$$\Pr(y = 1) = \tfrac{2}{3}, \quad \Pr(y = 0) = \tfrac{1}{3},$$

since among $\{y_1, y_2, y_3\} = \{1, 1, 0\}$, two are 1 and one is 0. Therefore we assign

$$\hat{y}_{\text{1-NN}}(0.6) = 1.$$

Therefore, the 1-NN classifier can be summarized as

$$\hat{y}_{\text{1-NN}}(x) = \begin{cases} 1, & x \le 0.6, \\ 0, & x > 0.6, \end{cases}$$

since at the boundary $x = 0.6$ the tie is broken in favor of class 1 by the higher prior probability.

3-NN

For a test point $x \in [0, 1]$, compute the distances to each training point:

$$d_1 = |x - 0.3|, \quad d_2 = |x - 0.5|, \quad d_3 = |x - 0.7|.$$

Since $k = 3$, we take all three training points as the nearest neighbors regardless of x. Their labels are

$$\{y_1, y_2, y_3\} = \{1,\ 1,\ 0\}.$$

Applying majority vote:

$$\sum_{i=1}^{3} y_i = 1 + 1 + 0 = 2 > 1,$$

so class 1 has the majority of votes. Therefore, the 3-NN classifier is

$$\hat{y}_{\text{3-NN}}(x) = 1 \quad \text{for all } x \in [0, 1].$$

(b)

We have training data $\{(0.3, 1), (0.5, 1), (0.7, 0)\}$. For each $x \in [0, 1]$ define

$$d_1(x) = |x - 0.3|, \quad d_2(x) = |x - 0.5|, \quad d_3(x) = |x - 0.7|.$$

Let $\mathcal{N}_2(x)$ be the indices of the two smallest distances. Then

$$\hat{y}_{\text{2-NN}}(x) = \frac{1}{2} \sum_{i \in \mathcal{N}_2(x)} y_i.$$

Solve

$$d_1 = d_2 \implies x = 0.4, \quad d_2 = d_3 \implies x = 0.6, \quad d_1 = d_3 \implies x = 0.5.$$

We have

Interval	Distance ordering	$\mathcal{N}_2(x)$	$\hat{y}_{\text{2-NN}}(x)$	Assigned value
$0 \le x < 0.4$	$d_1 < d_2 < d_3$	$\{1, 2\}$	$\frac{1+1}{2} = 1.0$	1
$0.4 < x < 0.5$	$d_2 < d_1 < d_3$	$\{1, 2\}$	1.0	1
$x = 0.5$	$d_2 < d_1 = d_3$	$\{2,\ 1 \text{ or } 3\}$	tie	1
$0.5 < x < 0.6$	$d_2 < d_3 < d_1$	$\{2, 3\}$	$\frac{1+0}{2} = 0.5$	0.5
$0.6 < x \le 1$	$d_3 < d_2 < d_1$	$\{2, 3\}$	0.5	0.5

Table 1: 2-NN: intervals, ordering of distances, nearest neighbors, raw mean and tie-broken assigned value

Hence partition $[0, 1]$ at $x = 0.5$.

At $x = 0.5$ the distances to 0.3 and 0.7 tie, so the two nearest neighbors are at 0.5 (label 1) and one of $\{0.3, 0.7\}$. We break the tie in favor of $y = 1$ since two of the three labels are 1, 1 might be more common in the population, so $\hat{y}_{\text{2-NN}}(0.5) = 1$.

Thus,

$$\hat{y}_{\text{2-NN}}(x) = \begin{cases} \dfrac{1+1}{2} = 1.0, & 0 \le x \le 0.5, \\[2ex] \dfrac{1+0}{2} = 0.5, & 0.5 < x \le 1. \end{cases}$$

R code to plot $\hat{y}_{\textbf{2-NN}}(x)$:

```
x_train <- c(0.3, 0.5, 0.7)
y_train <- c(1,   1,   0)

xgrid <- seq(0, 1, length=1001)
ygrid <- sapply(xgrid, function(x) {
  d <- abs(x - x_train)
  idx <- order(d)[1:2]
  mean(y_train[idx])
})

plot(xgrid, ygrid, type="s", ylim=c(0,1),
     xlab="x", ylab=expression(hat(y)[2-NN](x)),
     main="2-NN Function")
abline(v=0.5, lty=2, col="gray")  # decision boundary
points(0.5, 1, pch=19, col="black")# tie-broken value
```

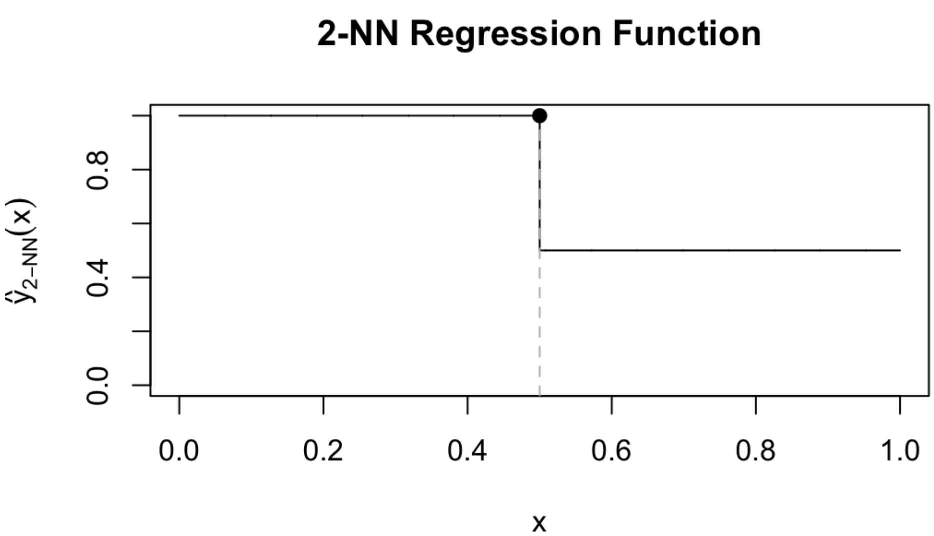

Figure 1: 2-NN Function

Question 2

(a)

In order to make $f_i(x)$ a probability density function, it requires for each i that

$$\int_{-\infty}^{\infty} f_i(x)\,dx \;=\; 1.$$

1. $f_1(x) = c_1\big(1 - |x - 0.5|\big)\, 1_{\{-0.5 \le x \le 1.5\}}(x)$

Set $u = x - 0.5$. Then $x \in [-0.5, 1.5]$ makes $u \in [-1, 1]$, and

$$\int_{-\infty}^{\infty} f_1(x)\, dx = c_1 \int_{-0.5}^{1.5} \big(1 - |x - 0.5|\big)\, dx = c_1 \int_{-1}^{1} \big(1 - |u|\big)\, du.$$

$$\int_{-1}^{1} \big(1 - |u|\big)\, du = 2\int_{0}^{1} (1 - u)\, du = 2\Big[\, u - \tfrac{1}{2}u^2 \Big]_0^1 = 2\big(1 - \tfrac{1}{2}\big) = 1.$$

Hence

$$c_1 \cdot 1 = 1$$

$$c_1 = 1$$

2. $f_2(x) = c_2\big(1 - |x|\big)\, 1_{\{-1 \le x \le 1\}}(x)$

$$\int_{-\infty}^{\infty} f_2(x)\, dx = c_2 \int_{-1}^{1} \big(1 - |x|\big)\, dx = c_2 \cdot 2\int_{0}^{1} (1 - x)\, dx = c_2 \cdot 2\Big[\, x - \tfrac{1}{2}x^2 \Big]_0^1 = c_2 \cdot 1.$$

Thus

$$c_2 = 1$$

3. $f_3(x) = c_3\big(2 - |x - 0.5|\big)\, 1_{\{-1.5 \le x \le 2.5\}}(x)$

Let $u = x - 0.5$. Then $x \in [-1.5, 2.5]$ makes $u \in [-2, 2]$, and

$$\int_{-\infty}^{\infty} f_3(x)\, dx = c_3 \int_{-1.5}^{2.5} \big(2 - |x - 0.5|\big)\, dx = c_3 \int_{-2}^{2} \big(2 - |u|\big)\, du.$$

$$\int_{-2}^{2} \big(2 - |u|\big)\, du = 2\int_{0}^{2} (2 - u)\, du = 2\Big[\, 2u - \tfrac{1}{2}u^2 \Big]_0^2 = 2(\, 4 - 2\,) = 4.$$

Hence

$$c_3 \cdot 4 = 1$$

$$c_3 = \tfrac{1}{4}$$

(b)

We have

$$f_1(x) = (1 - |x - 0.5|)\, 1_{[-0.5,\, 1.5]}(x), \quad f_2(x) = (1 - |x|)\, 1_{[-1,\, 1]}(x),$$

with prior probabilities $p_1 = 0.8$, $p_2 = 0.2$ and equal misclassification costs $c(2|1) = c(1|2)$.

The region minimizing the expected cost of misclassification (ECM) is

$$R_1 = \big\{x : c(2|1)\, p_1\, f_1(x) \;\ge\; c(1|2)\, p_2\, f_2(x)\big\}$$

$$R_2 = \big\{x : c(2|1)\, p_1\, f_1(x) \;<\; c(1|2)\, p_2\, f_2(x)\big\}$$

Since $c(2|1) = c(1|2)$, this simplifies to

$$R_1 = \big\{x : p_1\, f_1(x) \;\ge\; p_2\, f_2(x)\big\} \iff \frac{f_1(x)}{f_2(x)} \;\ge\; \frac{p_2}{p_1} = \frac{0.2}{0.8} = 0.25.$$

On the overlap of the supports, $x \in [-0.5, 1]$, we have

$$\frac{f_1(x)}{f_2(x)} = \frac{1 - |x - 0.5|}{1 - |x|} \; \geq \; 0.25.$$

In different set of x:

1. $x \in [-0.5, 0]$: $|x - 0.5| = 0.5 - x,\ |x| = -x$.

$$\frac{1 - (0.5 - x)}{1 - (-x)} = \frac{0.5 + x}{1 + x} \; \geq \; 0.25$$

$$0.5 + x \; \geq \; 0.25(1 + x)$$

$$x \; \geq \; -\tfrac{1}{3}.$$

   Hence on $[-0.5, 0]$, classify as Population 1 if $x \geq -\frac{1}{3}$.

2. $x \in [0, 0.5]$: $|x - 0.5| = 0.5 - x,\ |x| = x$.

$$\frac{1 - (0.5 - x)}{1 - x} = \frac{0.5 + x}{1 - x} \; \geq \; 0.25$$

$$0.5 + x \; \geq \; 0.25(1 - x)$$

$$x \; \geq \; -0.2,$$

   which holds for all $x \geq 0$. Thus classify as 1 on $[0, 0.5]$.

3. $x \in [0.5, 1]$: $|x - 0.5| = x - 0.5,\ |x| = x$.

$$\frac{1 - (x - 0.5)}{1 - x} = \frac{1.5 - x}{1 - x} \; \geq \; 0.25$$

$$1.5 - x \; \geq \; 0.25(1 - x)$$

$$1.5 - x \; \geq \; 0.25 - 0.25x$$

$$x \; \leq \; \tfrac{5}{3},$$

   which holds for all $x \leq 1$. Hence classify as 1 on $[0.5, 1]$.

Let

$$S = \operatorname{supp}(f_1) \cup \operatorname{supp}(f_2) = [-0.5,\, 1.5] \;\cup\; [-1,\, 1] = [-1,\, 1.5].$$

Within S, the decision boundary is at $x = -\frac{1}{3}$.

Hence the classification regions are

$$R_1 = \left\{\, x \in [-1, 1.5] : x \geq -\tfrac{1}{3} \right\}, \qquad R_2 = \left\{\, x \in [-1, 1.5] : x < -\tfrac{1}{3} \right\}.$$

By construction $R_1 \cup R_2 = S$, as required. For $x \notin S$, both $f_1(x) = f_2(x) = 0$ and no classification is needed.

(c)

R code.

```
c1 <- 1
c3 <- 1/4

xgrid <- seq(-1.5, 2.5, length=1000)
f1    <- c1 * (1 - abs(xgrid - 0.5)) * (xgrid >= -0.5 & xgrid <= 1.5)
f3    <- c3 * (2 - abs(xgrid - 0.5)) * (xgrid >= -1.5 & xgrid <= 2.5)

plot(xgrid, f1, type="l", col="blue", lwd=2,
     ylim=c(0, max(f1,f3)),
     xlab="x", ylab="Density",
     main="Densities f1 (blue) and f3 (red)")
lines(xgrid, f3, col="red", lwd=2)
legend("topright", legend=c("f1","f3"),
       col=c("blue","red"), lwd=2, bty="n")

p1 <- 0.8; p3 <- 0.2
R1_logical <- (p1 * f1 >= p3 * f3)

rug(xgrid[R1_logical], col="darkgreen", lwd=2)
```

Plot:

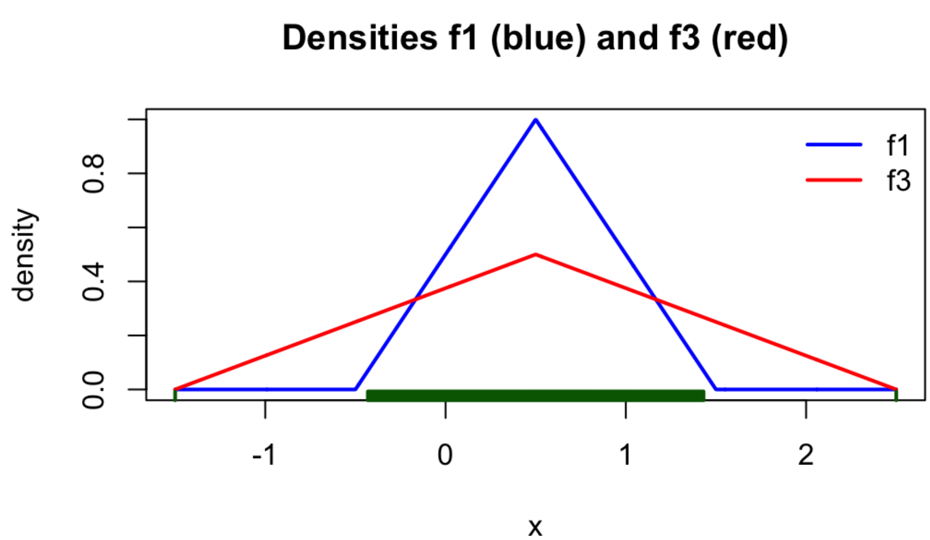

Figure 2: Densities f1 (blue) and f3 (red)

We compare

$$0.8\, f_1(x) \;\geq\; 0.2\, f_3(x) \quad\iff\quad \frac{f_1(x)}{f_3(x)} \;\geq\; \frac{0.2}{0.8} = 0.25.$$

Since

$$f_1(x) = 1 - |x - 0.5|, \quad f_3(x) = \tfrac{1}{4}\big(2 - |x - 0.5|\big),$$

To solve

$$\frac{1 - |x - 0.5|}{(2 - |x - 0.5|)/4} \;\geq 0.25$$

set

$$u = x - 0.5, \quad f_1(x) = 1 - |u|, \quad f_3(x) = \tfrac{1}{4}\,(2 - |u|).$$

The decision rule $p_1 f_1 \geq p_3 f_3$ with $p_1 = 0.8,\ p_3 = 0.2$ becomes

$$\frac{1 - |u|}{\tfrac{1}{4}(2 - |u|)} \ \geq\ 0.25.$$

Since on the support $-2 < u < 2$ the denominator $2 - |u|$ is positive, we may multiply both sides by $2 - |u|$:

$$4\,\big(1 - |u|\big) \ \geq\ 0.25\,\big(2 - |u|\big).$$

Expand both sides:

$$4 - 4|u| \ \geq\ 0.5 - 0.25\,|u|.$$

Therefore

$$|u| \ \leq\ \frac{3.5}{3.75} \ =\ \frac{14}{15}.$$

$u = x - 0.5$, the final boundary is

$$|x - 0.5| \ \leq\ \frac{14}{15},$$

i.e. $x \in \big[0.5 - \frac{14}{15},\, 0.5 + \frac{14}{15}\big] = [-0.4333,\ 1.4333]$.

The support of the two densities is

$$\mathrm{supp}(f_1) = [-0.5,\ 1.5], \qquad \mathrm{supp}(f_3) = [-1.5,\ 2.5].$$

Therefore the overall region on which we need to classify is

$$S = \mathrm{supp}(f_1)\ \cup\ \mathrm{supp}(f_3) = [-1.5,\ 2.5].$$

From the inequality $|x - 0.5| \leq \frac{14}{15}$ we obtain the ECM-optimal decision sets restricted to S:

$$R_1 = \big\{\, x \in [-1.5, 2.5] : \ |x - 0.5| \leq \tfrac{14}{15} \big\} = \big[0.5 - \tfrac{14}{15},\ 0.5 + \tfrac{14}{15}\big] = [-0.4333,\ 1.4333],$$

$$R_3 = \big\{\, x \in [-1.5, 2.5] : \ |x - 0.5| > \tfrac{14}{15} \big\} = (-1.5, -0.4333)\ \cup\ (1.4333, 2.5].$$

Thus on the support of f_1, f_3, one assigns

$$x \ \mapsto\ \begin{cases} \pi_1, & x \in [-0.4333,\ 1.4333], \\ \pi_3, & x \in (-1.5, -0.4333) \cup (1.4333, 2.5]. \end{cases}$$

or:

$$x \ \mapsto\ \begin{cases} \pi_1, & x \in \big[-\tfrac{13}{30},\ \tfrac{43}{30}\big], \\[4pt] \pi_3, & x \in \big[-\tfrac{3}{2},\ -\tfrac{13}{30}\big)\ \cup\ \big(\tfrac{43}{30},\ \tfrac{5}{2}\big]. \end{cases}$$

Question 3

(a)

Using the following R code:

```
group_means <- aggregate(cbind(GPA,GMAT) ~ admit,
                         data = gsbdata,
```

```
                        FUN  = mean)
print(group_means)

# 2. Overall mean
overall_mean <- colMeans(gsbdata[, c("GPA","GMAT")])
print(overall_mean)

# 3. Unweighted average of subgroup means
unweighted_mean <- colMeans(group_means[, c("GPA","GMAT")])
print(unweighted_mean)

# 4. Pooled covariance
ns     <- table(gsbdata$admit)
S_list <- by(gsbdata[, c("GPA","GMAT")],
             gsbdata$admit,
             cov)
S_pool <- Reduce(`+`,
                 Map(function(S, n) (n - 1) * S,
                     S_list, ns)
                ) / (sum(ns) - length(ns))
print(S_pool)
```

Results:

The class-specific sample means are

$$\bar{x}_1 = \begin{pmatrix} 3.403871 \\ 561.2258 \end{pmatrix}, \quad \bar{x}_2 = \begin{pmatrix} 2.482500 \\ 447.0714 \end{pmatrix}, \quad \bar{x}_3 = \begin{pmatrix} 2.992692 \\ 446.2308 \end{pmatrix}.$$

The overall sample mean is

$$\bar{x} = \begin{pmatrix} 2.974588 \\ 488.447059 \end{pmatrix},$$

which differs from the unweighted average of the subgroup means,

$$\frac{1}{3}\sum_{i=1}^{3} \bar{x}_i = \begin{pmatrix} 2.959688 \\ 484.842668 \end{pmatrix}.$$

The pooled covariance matrix is

$$S_{\text{pool}} = \begin{pmatrix} 0.03606795 & -2.01875915 \\ -2.01875915 & 3655.901121 \end{pmatrix}.$$

(b)

R code:

```
# 1. Within-group W
W <- matrix(0,2,2)
for(gid in unique(gsbdata$admit)) {
  Xg  <- as.matrix(subset(gsbdata, admit==gid)[,c("GPA","GMAT")])
  mu  <- as.numeric(group_means[group_means$admit==gid, c("GPA","GMAT")])
  D   <- sweep(Xg, 2, mu, FUN="-")
  W   <- W + t(D) %*% D
}
print(W)
```

```
# 2. Inverse of W
W_inv <- solve(W)
print(W_inv)

# 3. Between-group B (using sample-mean average)
grand_mean_unw <- colMeans(group_means[,c("GPA","GMAT")])
B <- matrix(0,2,2)
for(i in seq_len(nrow(group_means))) {
  mu_i <- as.numeric(group_means[i, c("GPA","GMAT")])
  d    <- mu_i - grand_mean_unw
  B    <- B + tcrossprod(d, d)
}
print(B)

# 4. Eigen-decomposition of W^{-1}B
M          <- W_inv %*% B
eig        <- eigen(M)
lambda_hat <- eig$values
a_vectors  <- eig$vectors
print(lambda_hat)
print(a_vectors)
```

Results:

The within-group matrix is

$$W = \begin{pmatrix} 2.957572 & -165.5383 \\ -165.5383 & 299783.8919 \end{pmatrix},$$

and its inverse is

$$W^{-1} = \begin{pmatrix} 0.3488985 & 0.0001926589 \\ 0.0001926589 & 0.000003442121 \end{pmatrix}.$$

The between-group matrix (using the unweighted mean of the three group means) is

$$B = \begin{pmatrix} 0.4260962 & 50.67771 \\ 50.67771 & 8751.92909 \end{pmatrix}.$$

The two estimated eigenvalues of $W^{-1}B$ are

$$\hat{\lambda}_1 = 0.1912519, \quad \hat{\lambda}_2 = 0.007064677,$$

with corresponding eigenvectors (as columns)

$$a_1 = \begin{pmatrix} 0.999998564 \\ 0.001694796 \end{pmatrix}, \quad a_2 = \begin{pmatrix} -0.999969462 \\ 0.007815072 \end{pmatrix}.$$

(c)

Below is the R code that implements the rule "assign x to class π_k if $a^\top x$ (in R^2) is closest to $a^\top \bar{x}_k$ for $k = 1, 2, 3$", where

$$a = [a_1 \; a_2]$$

is the 2×2 matrix of eigenvectors.

```
# 1. Ensure class means are ordered by admit = 1,2,3
group_means <- group_means[order(group_means$admit), ]

# 2. Form the 2×2 eigenvector matrix
A <- a_vectors    # columns are a1, a2

# 3. Build the 3×2 matrix of class means
mu_mat <- as.matrix(group_means[, c("GPA","GMAT")])

# 4. Project class means into canonical space y = A' x
z_means <- mu_mat %*% A   # 3×2 matrix

# 5. Define the two new observations
x_new <- matrix(c(3.21, 497,
                  3.22, 497),
                ncol = 2, byrow = TRUE)
colnames(x_new) <- c("GPA","GMAT")

# 6. Project new observations y = A' x
z_new <- x_new %*% A      # 2×2 matrix

# 7. Classify by nearest projected-centroid
predicted_class <- apply(z_new, 1, function(zj) {
  d2 <- rowSums((z_means - matrix(zj, nrow=3, ncol=2, byrow=TRUE))^2)
  k  <- which.min(d2)       # index of closest class-mean
  group_means$admit[k]      # return class label (1,2 or 3)
})

# 8. Output results
results <- data.frame(
  GPA   = x_new[,1],
  GMAT  = x_new[,2],
  class = predicted_class
)
print(results)
```

The printed output is:

GPA	GMAT	Predicted class
3.21	497	3
3.22	497	3

Thus both observations $(3.21, 497)$ and $(3.22, 497)$ are assigned to class π_3, which is the class broaderline.

(d)

Below is the R code that projects all observations into the first two Fisher discriminants and then plots them, colored by admission decision:

```
# 1. Build the discriminant matrix A from part (b)
A    <- a_vectors    # 2×2 matrix of eigenvectors a1,a2

# 2. Project all GPA/GMAT observations into LD1{LD2 space
X    <- as.matrix(gsbdata[, c("GPA","GMAT")])  # n×2
Z    <- X %*% A                                 # n×2
```

```
# 3. Define colors and symbols for each admit level
cols <- c("red","blue","green")[gsbdata$admit]
pchs <- c(16,17,15)[gsbdata$admit]

# 4. Plot
plot(Z,
     col   = cols,
     pch   = pchs,
     xlab  = "LD1 (first discriminant)",
     ylab  = "LD2 (second discriminant)",
     main  = "Gsb Admissions in First Two Discriminants")

# 5. Add legend
legend("topright",
       legend = c("Yes","No","Borderline"),
       col    = c("red","blue","green"),
       pch    = c(16,17,15),
       cex    = 0.7,
       bty    = "n")
```

This plot shows the resulting scatter:

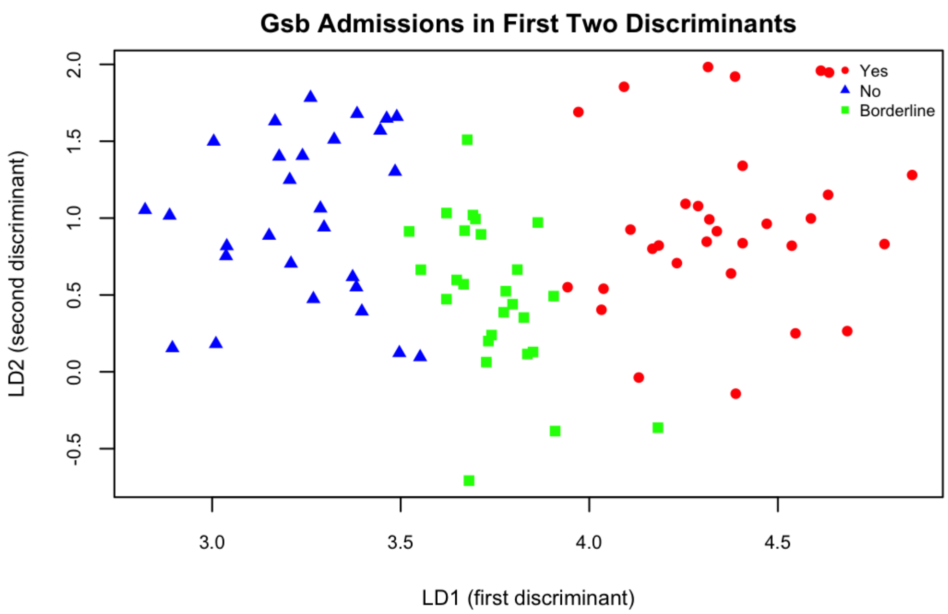

Figure 3: LDA

Comment on (c) and Policy Evaluation In part (c) we classified the two new points $(3.21, 497)$ and $(3.22, 497)$ by assigning each to the class whose projected mean in the LD1–LD2 plane was closest.

On the scatter, we see:

- **Yes** (π_1, red circles) occupy high LD1 values.
- **No** (π_2, blue triangles) occupy low LD1 values.
- **Borderline** (π_3, green squares) lie between the two, with some overlap.

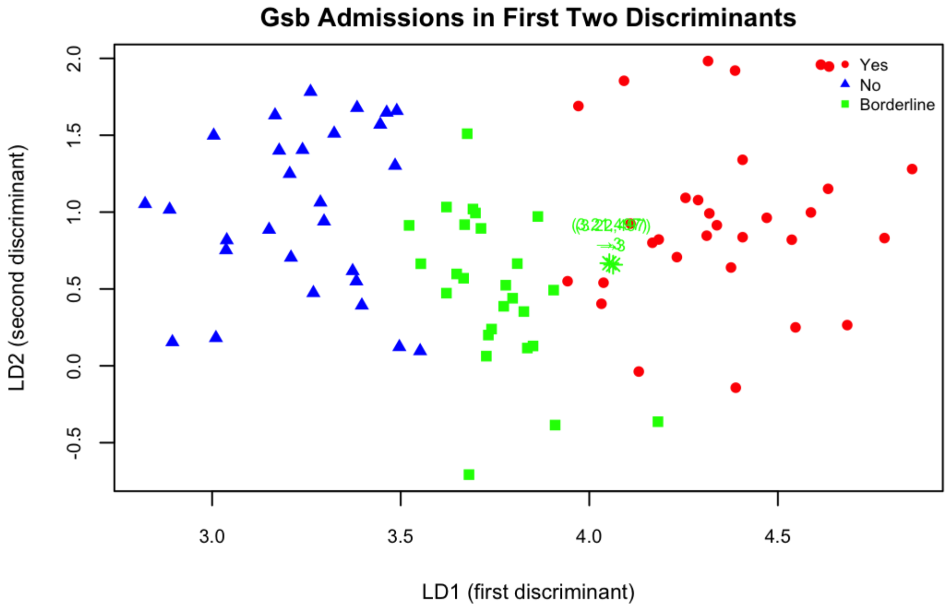

Figure 4: LDA with New Observations

The two new observations $(3.21, 497)$ and $(3.22, 497)$ are plotted as green stars and lie near to "Borderline" cloud (LD14.0, LD20.8). They are clearly distant from both the center of red "Yes" cluster and the center of blue "No" cluster, and fall near the region occupied by the committee's "Borderline" decisions.

This reinforces the result of part (c): these applicants are truly marginal cases, justifying their assignment to π_3.

Overall, the admission policy—using a linear combination of GPA and GMAT to separate admit vs. deny and leaving ambiguous cases in a "Borderline" category, produces a clear separation along the first discriminant (LD1). Only the borderline group overlaps slightly with the admits and rejects, which is the intended purpose of a borderline decision. Hence the policy appears to be good and supported by the data.

Question 4

(a)

Using the following code we got:

```
X1 = cbind(c(3,2,4),c(7,4,7))
X2=cbind(c(6,5,4),c(9,7,8))
mu1 = colMeans(X1)
mu2 = colMeans(X2)
S1=cov(X1)
S2=cov(X2)
```

$$\bar{x}_1 = \begin{pmatrix} 3 \\ 6 \end{pmatrix} \quad \bar{x}_2 = \begin{pmatrix} 5 \\ 8 \end{pmatrix} \quad \bar{x}_1 - \bar{x}_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$$

$$S_1 = \begin{pmatrix} 1 & 1.5 \\ 1.5 & 3 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

$$S_p = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1+n_2-2} = \frac{S_1+S_2}{2} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

The inverse of S_p is

$$S_p^{-1} = \frac{1}{(1 \cdot 2 - 1 \cdot 1)} \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}.$$

Hence the linear discriminant function can be written as

$$y \;=\; (\bar{x}_1 - \bar{x}_2)^\top S_p^{-1}\, x = (-2 - 2) \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} x = \begin{pmatrix} -2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -2\,x_1 + 0 \cdot x_2 = -2\,x_1,$$

where $x = (x_1, x_2)^\top$ is the feature vector for a new observation, and x_1 denotes its first coordinate.

(b)

Under equal priors and equal costs, allocate a new observation x_o to π_1 if

$$\hat{y}_o = (\bar{x}_1 - \bar{x}_2)^\top S_{\text{pool}}^{-1}\, x_o \;\geq\; \frac{1}{2}\, (\bar{x}_1 - \bar{x}_2)^\top S_{\text{pool}}^{-1} (\bar{x}_1 + \bar{x}_2) = \hat{m},$$

and to π_2 otherwise.

The sample means are:

$$\bar{x}_1 - \bar{x}_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \quad S_{\text{pool}}^{-1} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}, \quad \bar{x}_1 + \bar{x}_2 = \begin{pmatrix} 8 \\ 14 \end{pmatrix}.$$

$$S_{\text{pool}}^{-1}(\bar{x}_1 + \bar{x}_2) = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 8 \\ 14 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix},$$

$$\hat{m} = \tfrac{1}{2}\, (\bar{x}_1 - \bar{x}_2)^\top \begin{pmatrix} 2 \\ 6 \end{pmatrix} = \tfrac{1}{2} \big[(-2)(2) + (-2)(6) \big] = -8.$$

For any $x = (x_1, x_2)^\top$,

$$\hat{y}_o = (\bar{x}_1 - \bar{x}_2)^\top S_{\text{pool}}^{-1} x = \begin{pmatrix} -2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -2\,x_1\,.$$

Thus the decision rule is

$$-2\,x_1 \;\geq\; -8 \quad\Longleftrightarrow\quad x_1 \;\leq\; 4.$$

$$x^{(1)} = \begin{pmatrix} 4.1 \\ 5 \end{pmatrix}: \quad \hat{y}_o = -2(4.1) = -8.2 < -8 \;\Longrightarrow\; \text{classified as} \quad \pi_2,$$

$$x^{(2)} = \begin{pmatrix} 3.9 \\ 9 \end{pmatrix}: \quad \hat{y}_o = -2(3.9) = -7.8 \geq -8 \;\Longrightarrow\; \text{classified as} \quad \pi_1.$$

(c)

Under the multivariate normal model with equal covariance Σ, the classification rule is

$$(\mu_1 - \mu_2)^\top \Sigma^{-1} x_0 \;-\; \tfrac{1}{2}\, (\mu_1 - \mu_2)^\top \Sigma^{-1} (\mu_1 + \mu_2) \;\geq\; \ln \frac{c(1 \mid 2)\, p_2}{c(2 \mid 1)\, p_1},$$

allocate to π_1 if the inequality holds, and to π_2 otherwise.

Based on (a) and (b), we already have:

$$\bar{x}_1 = \begin{pmatrix} 3 \\ 6 \end{pmatrix}, \quad \bar{x}_2 = \begin{pmatrix} 5 \\ 8 \end{pmatrix}, \quad \Sigma = S_{\text{pool}} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix},$$

$$\Sigma^{-1} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}, \quad \bar{x}_1 - \bar{x}_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \quad \bar{x}_1 + \bar{x}_2 = \begin{pmatrix} 8 \\ 14 \end{pmatrix}.$$

Costs and priors:

$$c(2 \mid 1) = 3, \quad c(1 \mid 2) = 20, \quad p_1 = 0.1, \quad p_2 = 0.9.$$

We have:

$$(\bar{x}_1 - \bar{x}_2)^\top \Sigma^{-1} = \begin{pmatrix} -2 & -2 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} -2 & 0 \end{pmatrix},$$

so for $x = (x_1, x_2)^\top$,

$$g(x) = (\bar{x}_1 - \bar{x}_2)^\top \Sigma^{-1} x - \tfrac{1}{2}\,(\bar{x}_1 - \bar{x}_2)^\top \Sigma^{-1} (\bar{x}_1 + \bar{x}_2) = -2x_1 - \tfrac{1}{2}\big[(-2, 0) \cdot (8, 14)\big] = -2x_1 + 8.$$

The right-hand side is

$$\ln \frac{c(1 \mid 2)\, p_2}{c(2 \mid 1)\, p_1} = \ln\!\left(\frac{20 \cdot 0.9}{3 \cdot 0.1}\right) = \ln(60) \approx 4.094.$$

Thus the decision rule is

$$-2x_1 + 8 \geq \ln(60) \quad \Longleftrightarrow \quad x_1 \;\leq\; \frac{8 - \ln(60)}{2} \approx 1.953.$$

Therefore,

$$x^{(1)} = \begin{pmatrix} 4.1 \\ 5 \end{pmatrix}: \quad x_1 = 4.1 > 1.953 \implies \text{classified as} \quad \pi_2,$$

$$x^{(2)} = \begin{pmatrix} 3.9 \\ 9 \end{pmatrix}: \quad x_1 = 3.9 > 1.953 \implies \text{classified as} \quad \pi_2.$$

(d)

With two bivariate normals having sample means $\bar{x}_1, \bar{x}_2$ and covariances S_1, S_2, and under equal priors and costs, allocate x to π_1 if

$$d(x) \;\geq\; \ln\!\left(\frac{c(1 \mid 2)\, p_2}{c(2 \mid 1)\, p_1}\right),$$

and to π_2 otherwise.

Since $c(1 \mid 2) = c(2 \mid 1)$,$p_1 = p_2$, allocate x to π_1 if

$$d(x) \;\geq 0,$$

where

$$d(x) = -\tfrac{1}{2}\, x^\top (S_1^{-1} - S_2^{-1}) x \;+\; (\bar{x}_1^\top S_1^{-1} - \bar{x}_2^\top S_2^{-1}) x \;-\; \hat{k},$$

and

$$\hat{k} = \tfrac{1}{2}\Big(\bar{x}_1^\top S_1^{-1} \bar{x}_1 \;-\; \bar{x}_2^\top S_2^{-1} \bar{x}_2\Big) + \tfrac{1}{2} \ln \frac{|S_1|}{|S_2|}.$$

We have

$$\bar{x}_1 = \begin{pmatrix} 3 \\ 6 \end{pmatrix}, \quad \bar{x}_2 = \begin{pmatrix} 5 \\ 8 \end{pmatrix},$$

$$S_1 = \begin{pmatrix} 1 & 1.5 \\ 1.5 & 3 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix},$$

$$S_1^{-1} = \frac{1}{0.75} \begin{pmatrix} 3 & -1.5 \\ -1.5 & 1 \end{pmatrix} = \begin{pmatrix} 4 & -2 \\ -2 & \frac{4}{3} \end{pmatrix}, \quad S_2^{-1} = \frac{1}{0.75} \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix} = \begin{pmatrix} \frac{4}{3} & -\frac{2}{3} \\ -\frac{2}{3} & \frac{4}{3} \end{pmatrix}.$$

Compute $\hat{k}$

$$\bar{x}_1^\top S_1^{-1}\bar{x}_1 = \begin{pmatrix}3 & 6\end{pmatrix}\begin{pmatrix}4 & -2\\ -2 & \frac{4}{3}\end{pmatrix}\begin{pmatrix}3\\ 6\end{pmatrix} = 12,$$

$$\bar{x}_2^\top S_2^{-1}\bar{x}_2 = \begin{pmatrix}5 & 8\end{pmatrix}\begin{pmatrix}\frac{4}{3} & -\frac{2}{3}\\ -\frac{2}{3} & \frac{4}{3}\end{pmatrix}\begin{pmatrix}5\\ 8\end{pmatrix} = \frac{196}{3}.$$

Since $|S_1| = |S_2| = 0.75$, the log-term is zero. Hence

$$\hat{k} = \tfrac{1}{2}\Big(12 - \tfrac{196}{3}\Big) = -\tfrac{80}{3}.$$

Simplify $d(x)$

Let

$$\Delta = S_1^{-1} - S_2^{-1} = \begin{pmatrix}4 - \frac{4}{3} & -2 - (-\frac{2}{3})\\ -2 - (-\frac{2}{3}) & \frac{4}{3} - \frac{4}{3}\end{pmatrix} = \begin{pmatrix}\frac{8}{3} & -\frac{4}{3}\\ -\frac{4}{3} & 0\end{pmatrix},$$

and

$$b^\top = \bar{x}_1^\top S_1^{-1} - \bar{x}_2^\top S_2^{-1} = \begin{pmatrix}3 & 6\end{pmatrix}\begin{pmatrix}4 & -2\\ -2 & \frac{4}{3}\end{pmatrix} - \begin{pmatrix}5 & 8\end{pmatrix}\begin{pmatrix}\frac{4}{3} & -\frac{2}{3}\\ -\frac{2}{3} & \frac{4}{3}\end{pmatrix} = \begin{pmatrix}-\frac{4}{3} & -\frac{16}{3}\end{pmatrix}.$$

Thus

$$d(x) = -\tfrac{1}{2}\,x^\top \Delta\, x + b^\top x - \hat{k},$$

$$\Delta = \begin{pmatrix}\frac{8}{3} & -\frac{4}{3}\\ -\frac{4}{3} & 0\end{pmatrix}, \quad b = \begin{pmatrix}-\frac{4}{3}\\ -\frac{16}{3}\end{pmatrix}, \quad \hat{k} = -\tfrac{80}{3}.$$

$$x^\top \Delta\, x = \begin{pmatrix}x_1 & x_2\end{pmatrix}\begin{pmatrix}\frac{8}{3} & -\frac{4}{3}\\ -\frac{4}{3} & 0\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} = \frac{8}{3}\,x_1^2 - 2\cdot\frac{4}{3}\,x_1x_2 + 0\cdot x_2^2 = \frac{8}{3}x_1^2 - \frac{8}{3}x_1x_2.$$

$$-\tfrac{1}{2}\,x^\top \Delta\, x = -\tfrac{1}{2}\Big(\frac{8}{3}x_1^2 - \frac{8}{3}x_1x_2\Big) = -\frac{4}{3}x_1^2 + \frac{4}{3}x_1x_2.$$

$$b^\top x = \begin{pmatrix}-\frac{4}{3} & -\frac{16}{3}\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} = -\frac{4}{3}x_1 - \frac{16}{3}x_2,$$

$$-\hat{k} = -(-\tfrac{80}{3}) = \frac{80}{3}.$$

Putting these together,

$$d(x) = -\tfrac{1}{2}\,x^\top \Delta\, x \;+\; b^\top x \;-\; \hat{k} = \Big(-\tfrac{4}{3}x_1^2 + \tfrac{4}{3}x_1x_2\Big) + \Big(-\tfrac{4}{3}x_1 - \tfrac{16}{3}x_2\Big) + \tfrac{80}{3},$$

$$d(x) = -\frac{4}{3}x_1^2 + \frac{4}{3}x_1x_2 - \frac{4}{3}x_1 - \frac{16}{3}x_2 + \frac{80}{3}.$$

Classification of $x^{(1)} = (4.1, 5)^\top$ **and** $x^{(2)} = (3.9, 9)^\top$

$$\begin{aligned} d\big(4.1, 5\big) &= -\tfrac{4}{3}(4.1)^2 + \tfrac{4}{3}(4.1)(5) - \tfrac{4}{3}(4.1) - \tfrac{16}{3}(5) + \tfrac{80}{3} \approx -0.547 < 0 \implies \text{classified as} \quad \pi_2,\\ d\big(3.9, 9\big) &= -\tfrac{4}{3}(3.9)^2 + \tfrac{4}{3}(3.9)(9) - \tfrac{4}{3}(3.9) - \tfrac{16}{3}(9) + \tfrac{80}{3} \approx -0.013 < 0 \implies \text{classified as} \quad \pi_2. \end{aligned}$$

Both observations are classified into π_2 since $d(x) < 0$.

(e)

Comparison:

Observation	LDA decision (b)	QDA decision (d)
$x^{(1)} = (4.1, 5)^\top$	π_2	π_2
$x^{(2)} = (3.9, 9)^\top$	π_1	π_2

LDA (equal covariances)

Under the assumption $\Sigma_1 = \Sigma_2$, the discriminant function simplifies to

$$y \;=\; -2\,x_1 + 8,$$

so the decision boundary is the vertical line $x_1 = 4$. Consequently, only the first coordinate x_1 matters:

- $x_1 = 4.1 > 4 \implies \pi_2$.
- $x_1 = 3.9 < 4 \implies \pi_1$.

QDA (unequal covariances)

Allowing $\Sigma_1 \neq \Sigma_2$ leads to the quadratic rule

$$d(x) = -\tfrac{1}{2}\,x^\top\big(S_1^{-1} - S_2^{-1}\big)x + (\bar{x}_1^\top S_1^{-1} - \bar{x}_2^\top S_2^{-1})\,x - \hat{k},$$

which yields a curved decision boundary depending on both x_1 and x_2. Evaluating:

$$d(4.1, 5) \approx -0.547 < 0 \implies \pi_2, \quad d(3.9, 9) \approx -0.013 < 0 \implies \pi_2.$$

We can see that

- **LDA** assumes $\Sigma_1 = \Sigma_2$, hence produces a linear boundary that, in this case, ignores x_2 entirely.
- **QDA** respects the heterogeneity $S_1 \neq S_2$, so the classification surface bends and assigns extra weight to both the second coordinate x_2 and the quadratic term in x_1^2.

In particular, $(3.9, 9)$ has $x_1 < 4$ (favoring π_1 under LDA) but a very large x_2, which QDA deems more consistent with the covariance of π_2.

If the true populations have different covariance structures, QDA is generally more accurate because it accounts for feature-specific variances and correlations. However, with small samples or minor covariance differences, LDA's simpler linear boundary can be more stable.

(f)

We test

$$H_0: \; \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 \quad \text{vs.} \quad H_1: \; \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$$

for two bivariate samples of sizes $n_1 = 3$, $n_2 = 3$, dimension $p = 2$.

Sample means and pooled covariance

$$\bar{x}_1 = \begin{pmatrix} 3 \\ 6 \end{pmatrix}, \quad \bar{x}_2 = \begin{pmatrix} 5 \\ 8 \end{pmatrix}, \quad \bar{x}_1 - \bar{x}_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}.$$

Sample covariances:

$$S_1 = \begin{pmatrix} 1 & 1.5 \\ 1.5 & 3 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

Pooled covariance:

$$S_p = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1+n_2-2} = \frac{1}{4}\Big(S_1 + S_2\Big) = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix},$$

$$S_p^{-1} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}.$$

Hotelling's T^2 statistic under $H_0 : \mu_1 = \mu_2$

$$T^2 = \frac{n_1\, n_2}{n_1+n_2}\,(\bar{x}_1 - \bar{x}_2)^\top S_p^{-1}\,(\bar{x}_1 - \bar{x}_2).$$

Here

$$(\bar{x}_1 - \bar{x}_2)^\top S_p^{-1} = \begin{pmatrix} -2 & -2 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} -2 & 0 \end{pmatrix},$$

$$T^2 = \frac{3 \times 3}{6}\,\begin{pmatrix} -2, & 0 \end{pmatrix} \begin{pmatrix} -2 \\ -2 \end{pmatrix} = \tfrac{9}{6} \times 4 = 6.$$

Under H_0,

$$\frac{n_1+n_2-p-1}{p\,(n_1+n_2-2)}\,T^2 \;\sim\; F_{p,\,n_1+n_2-p-1}$$

$$F = \frac{3}{8}\,T^2 \sim F_{2,3}.$$

Numerically,

$$F = \frac{3}{8} \times 6 = 2.25.$$

At the 5% level, the critical value is

$$F_{2,3;0.95} \approx 9.55,$$

and since $2.25 < 9.55$, we don't reject H_0.

Thus, there is no significant evidence at the 5% level to conclude that the two population mean vectors differ.

Question 5

(a)

Using the following R code, we got the plot:

```
# 1. Input the two classes
X1 <- matrix(c(3,2,4,
               7,4,7),
             ncol = 2, byrow = FALSE)
X2 <- matrix(c(6,5,4,
               9,7,8),
             ncol = 2, byrow = FALSE)

# 2. Compute class means and pooled covariance
mu1 <- colMeans(X1)
mu2 <- colMeans(X2)
S1  <- cov(X1)
S2  <- cov(X2)
n1 <- nrow(X1)
n2 <- nrow(X2)
```

```
Sp <- ((n1 - 1)*S1 + (n2 - 1)*S2) / (n1 + n2 - 2)

# 3. LDA coefficients: w' x = m
w <- solve(Sp) %*% (mu1 - mu2)          # weight vector
m <- 0.5 * t(mu1 - mu2) %*% solve(Sp) %*% (mu1 + mu2)  # threshold

# 4. Base scatter plot
plot(X1, col='red', pch=16, xlim=c(1,8), ylim=c(2,10),
     xlab=expression(x[1]), ylab=expression(x[2]),
     main='LDA Boundary with New Observations')
points(X2, col='blue', pch=17)
legend('topright',
       legend=c('Class 1','Class 2','New Obs'),
       pch=c(16,17,8), col=c('red','blue','black'))

# 5. Plot the LDA decision boundary: w[1]*x + w[2]*y = m
if (abs(w[2]) > 1e-6) {
  abline(a = m/w[2], b = -w[1]/w[2], col='darkgreen', lwd=2)
} else {
  abline(v = m/w[1], col='darkgreen', lwd=2)
}

# 6. Add the two new points
new_pts <- rbind(c(4.1, 5), c(3.9, 9))
points(new_pts, pch=8, col='black', cex=1.5)
```

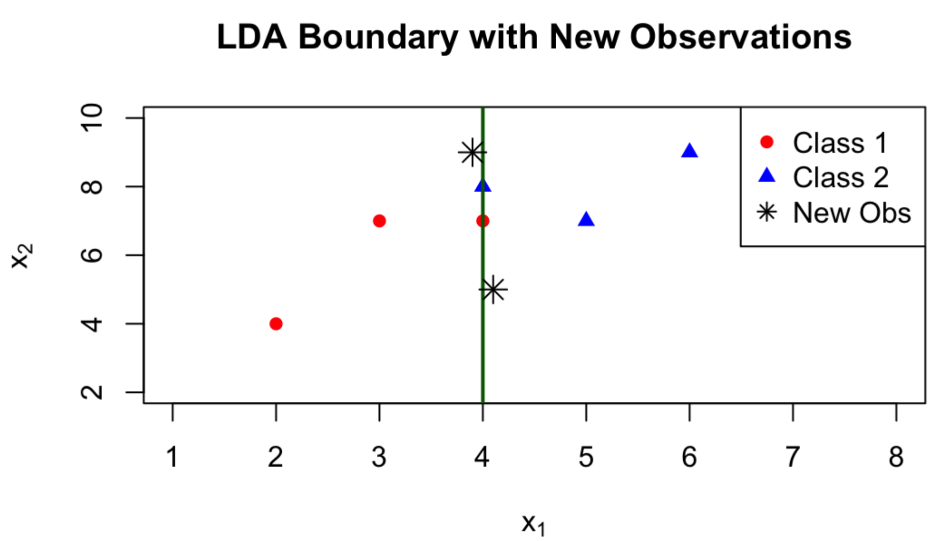

Figure 5: LDA Plot

(b)

(i)

We have two training classes

$$X_1 = \{(3,7)',\ (2,4)',\ (4,7)'\}, \quad y = +1, \qquad X_2 = \{(6,9)',\ (5,7)',\ (4,8)'\}, \quad y = -1.$$

A hard-margin SVM finds the unique separating hyperplane

$$w^\top x + b = 0$$

that maximizes the margin subject to

$$y_i\big(w^\top x_i + b\big) \ \geq\ 1 \quad \text{for all training points.}$$

First, plot all six points on the x_1-x_2 plane. By sliding a pair of parallel lines of maximum separation between the two classes, finds that three points lie on these margin lines, namely

$$(4,7)'\ (y = +1), \quad (5,7)',\ (4,8)'\ (y = -1).$$

These are the support vectors. They satisfy

$$w^\top x + b = +1 \quad \text{at } (4,7)', \qquad w^\top x + b = -1 \quad \text{at } (5,7)',\ (4,8)',$$

and therefore uniquely determine the separator $w^\top x + b = 0$ by solving those three equations for w and b.

$$\begin{cases} 4w_1 + 7w_2 + b = +1, \\ 5w_1 + 7w_2 + b = -1, \\ 4w_1 + 8w_2 + b = -1, \end{cases}$$

$$\begin{aligned} \text{Subtract the first from the second:} \quad & (5-4)w_1 = -1 - 1 \implies w_1 = -2, \\ \text{Subtract the first from the third:} \quad & (8-7)w_2 = -1 - 1 \implies w_2 = -2. \end{aligned}$$

Substitute $w_1 = w_2 = -2$ into the first equation:

$$4(-2) + 7(-2) + b = 1 \implies -8 - 14 + b = 1 \implies b = 23.$$

Hence the SVM solution is

$$w = \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \quad b = 23$$

so that the decision boundary is

$$\begin{gathered} w^\top x + b = 0 \\ -2x_1 - 2x_2 + 23 = 0 \\ x_1 + x_2 = 11.5. \end{gathered}$$

Add this line to the base plot:

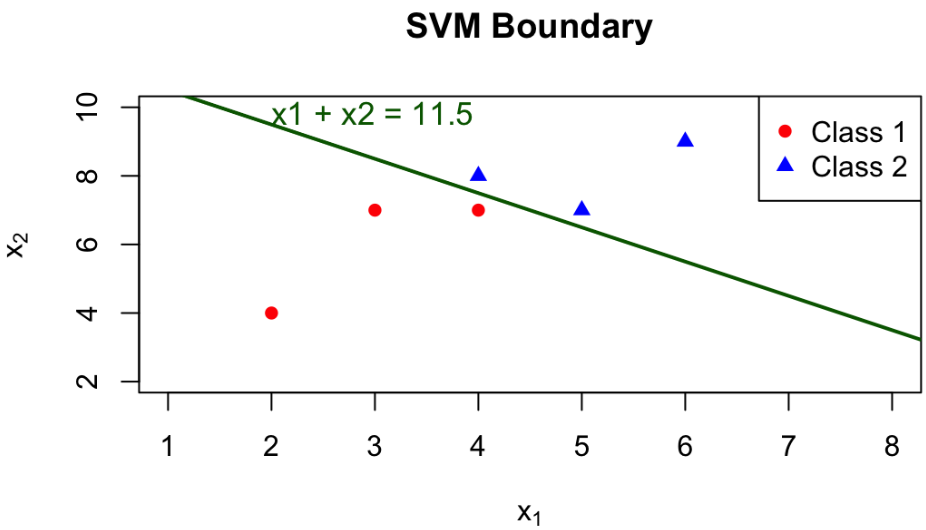

Figure 6: SVM Boundary

(ii)

The support vectors are exactly those training points satisfying:

$$y_i\big(w^\top x_i + b\big) = 1.$$

Using $w = (-2, -2)^\top$ and $b = 23$:

$$\begin{aligned}
w^\top(4,7) + b &= -2(4+7) + 23 = 1, \quad & y = +1,\\
w^\top(5,7) + b &= -2(5+7) + 23 = -1, \quad & y = -1,\\
w^\top(4,8) + b &= -2(4+8) + 23 = -1, \quad & y = -1.
\end{aligned}$$

Hence the support vectors are

$$(4,7)^\top\ (\pi_1), \quad (5,7)^\top,\ (4,8)^\top\ (\pi_2).$$

(iii)

Add the point to the plot:

- The point $(4.1, 5)$ lies on the π_1 side of the decision boundary and is therefore classified as belonging to class π_1.
- The point $(3.9, 9)$ lies on the π_2 side of the decision boundary and is therefore classified as belonging to class π_2.

(c)

Comparison:

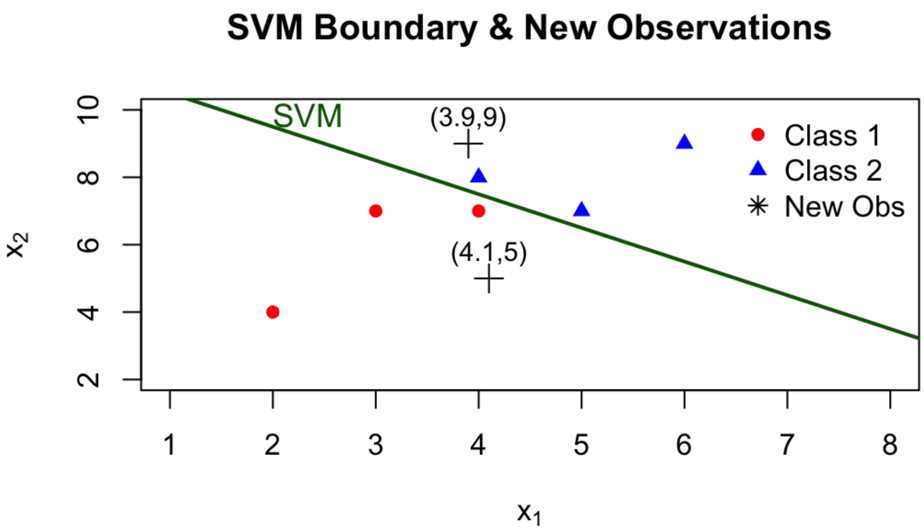

Figure 7: SVM with New Observations

Name	LDA (Part a)	SVM (Part b)
Boundary equation	$x_1 = 4$ (vertical line)	$x_1 + x_2 = 11.5$ (diagonal line)
Features used	only x_1	both x_1 and x_2
Underlying assumption	Gaussian classes, equal Σ	none (margin-maximization)
Robustness	sensitive if $\Sigma_1 \neq \Sigma_2$	robust to outliers, uses support vectors

$$\hat{y}_{\text{LDA}}(4.1, 5) = \pi_2, \quad \hat{y}_{\text{LDA}}(3.9, 9) = \pi_1,$$
$$\hat{y}_{\text{SVM}}(4.1, 5) = \pi_1, \quad \hat{y}_{\text{SVM}}(3.9, 9) = \pi_2.$$

I prefer the **SVM** classifier in this setting, because:

- It uses both features x_1 and x_2, whereas LDA's vertical cut ignores the large x_2 value of $(3.9, 9)$.
- Margin maximization focuses on the hardest points to classify, often yielding better generalization with small samples.

(d)

(i)

According to Question 4(d):

$$d(x) = -\frac{4}{3}x_1^2 + \frac{4}{3}x_1x_2 - \frac{4}{3}x_1 - \frac{16}{3}x_2 + \frac{80}{3}.$$

In expanded form for our estimates this becomes

$$3d(x) = -4x_1^2 \ + \ 4x_1x_2 \ - \ 4x_1 \ - \ 16x_2 \ + \ 80 \ = \ 0.$$

Writing this as

$$Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F = 0$$

we identify

$$A = -4, \quad B = 4, \quad C = 0, \quad D = -4, \quad E = -16, \quad F = 80.$$

The discriminant of the conic is

$$\Delta = B^2 - 4AC = 4^2 - 4(-4)(0) = 16 > 0,$$

which characterizes a hyperbola, not an ellipse (for which $\Delta < 0$).

In fact one can factor the left-hand side as

$$-4x_1^2 + 4x_1x_2 - 4x_1 - 16x_2 + 80 = -4\,(x_1 - 4)\,(x_2 - (x_1 + 5)) = 0,$$

so the boundary degenerates into the two straight lines

$$x_1 = 4 \quad \text{and} \quad x_2 = x_1 + 5.$$

Within our plotting region the relevant branch is the hyperbola (degenerate here to two lines).

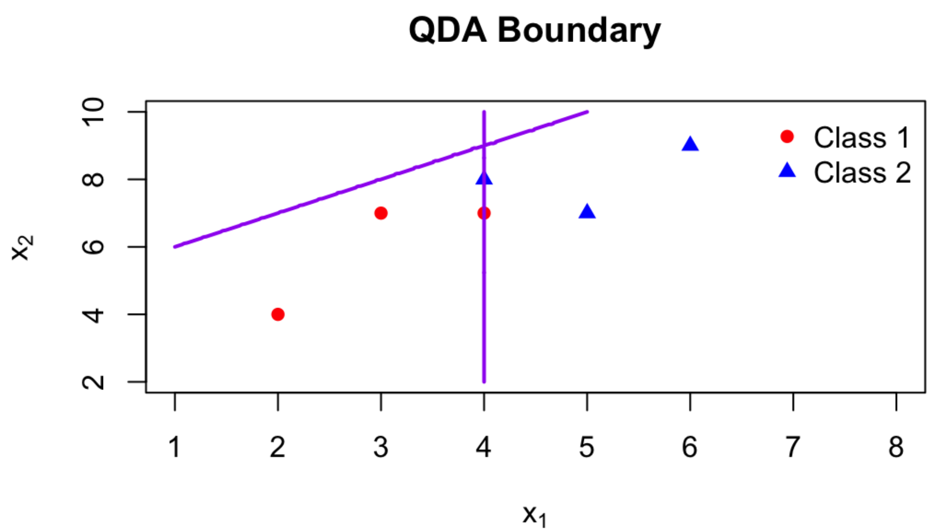

Figure 8: Bounday of QDA

(ii)

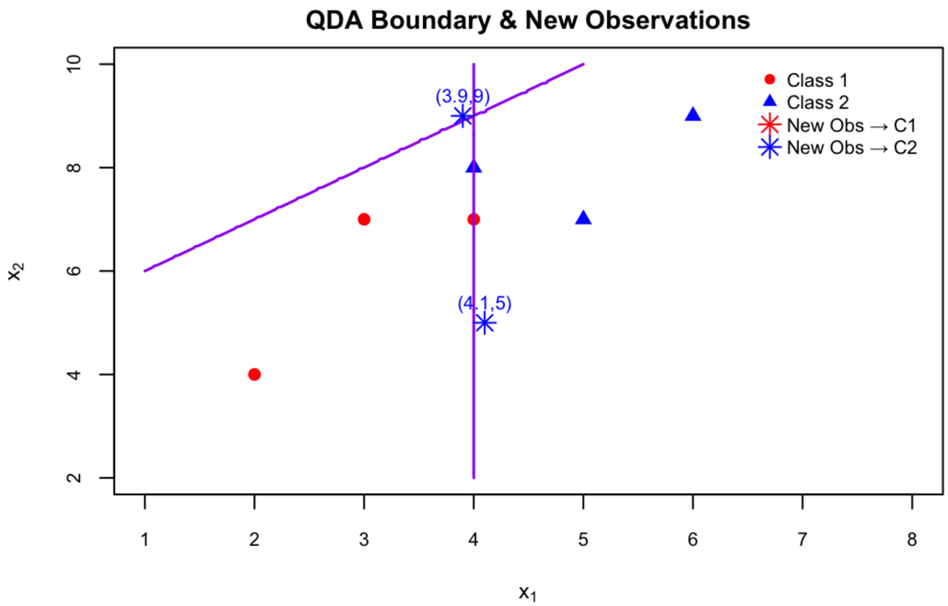

Figure 9: QDA with Boundary

The output shows:

x_1	x_2	class
4.1	5	π_2
3.9	9	π_2

Hence under the QDA rule,

$$(4.1, 5) \mapsto \pi_2, \quad (3.9, 9) \mapsto \pi_2.$$

(iii)

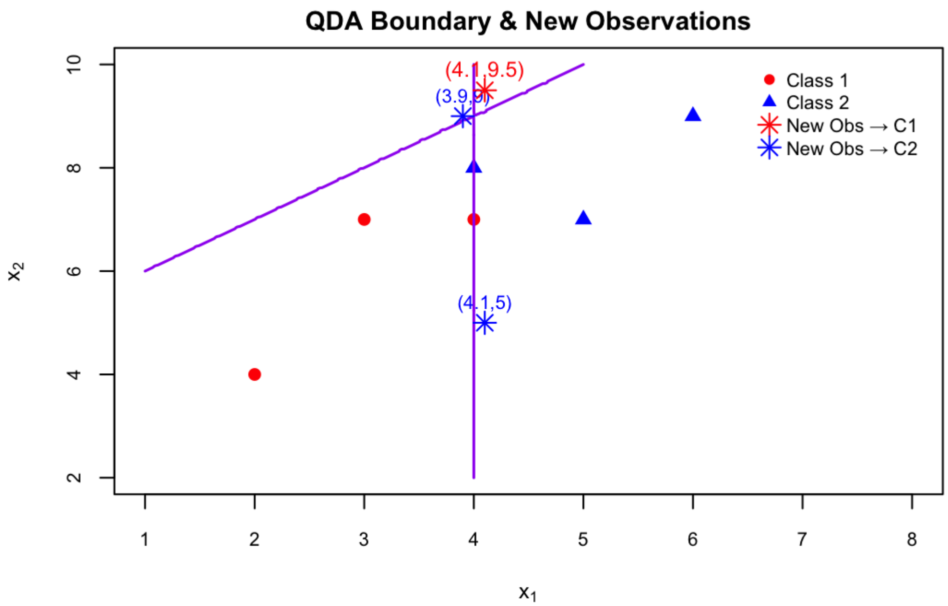

Figure 10: QDA with New Observations

For the new point $x = (4.1, 9.5)^\top$, the integer-scaled discriminant is

$$d_3(x) = -4(4.1)^2 + 4\,(4.1)(9.5) - 4(4.1) - 16(9.5) + 80 \;=\; 2.4 \;>\; 0.$$

Thus $d_3(x) > 0$ and the quadratic rule assigns x to population π_1.

However, it is not reasonable, since from the plotted QDA boundary (the purple lines) and the training-point clusters, it is clear that:

All Class 1 points (red circles) lie in the lower-left wedge bounded by the lines $\{x_1 = 4\}$ and $\{x_2 = x_1 + 5\}$. The new point $(4.1, 9.5)$ falls in the upper-right wedge, on the opposite side of both boundary lines from the red points.

Yet the QDA rule assigns $(4.1, 9.5)$ to π_1, simply because algebraically

$$d_3(4.1, 9.5) > 0 \implies \text{“}\pi_1\text{” region,}$$

even though geometrically it lies far from the red cluster and on the “wrong” side of the margin. Thus, the QDA rule produces a classification region that does not capture the actual cluster shape of π_1. In this case, the classification of $(4.1, 9.5)$ into π_1 is therefore not reasonable.

(iv)

In theory, Quadratic Discriminant Analysis (QDA) is more flexible than Linear Discriminant Analysis (LDA) because it estimates a separate covariance matrix for each class and thus can produce curved decision boundaries that adapt to non-elliptical or non-linearly separable clusters. However, the extra flexibility comes at the cost of estimating many more parameters (two covariance matrices instead of one) and hence greater variance, especially with small samples.

In our example with only three points per class, the estimated QDA boundary turns out to be the degenerate conic

$$-4x_1^2 \;+\; 4x_1x_2 \;-\; 4x_1 \;-\; 16x_2 \;+80 \;=\; 0,$$

which factors exactly into the two straight lines

$$x_1 = 4 \quad \text{and} \quad x_2 = x_1 + 5.$$

Consequently, the QDA "region" for class π_1 is

$$R_1^{\text{QDA}} = \{\, x_1 > 4,\ x_2 > x_1 + 5\} \,\cup\, \{\, x_1 < 4,\ x_2 < x_1 + 5\},$$

two disjoint wedges in the plane. By contrast, the simpler LDA rule

$$x_1 = 4 \quad \Longrightarrow \quad R_1^{\text{LDA}} = \{x_1 \le 4\}$$

is a single contiguous half-space.

Why LDA is preferable here

1. **Cluster shape.** The 3 "Yes" points $\{(3,7),(2,4),(4,7)\}$ form one cohesive cluster on the left of $x_1 = 4$. LDA's half-space $\{x_1 \le 4\}$ cleanly contains that cluster. QDA's two-wedge region splits the cluster into two disconnected pieces—some "Yes" points end up on the "wrong" side of one of its boundary lines.
2. **Overfitting / variance.** With only $n_1 = n_2 = 3$, estimating separate covariances produces a noisy boundary that perfectly "fits" the small sample but misaligns with the true cluster geometry. LDA's single pooled covariance is more stable, yielding a more reliable linear separator.
3. **Interpretability and simplicity.** The vertical line $x_1 = 4$ is easy to interpret and implement; the QDA region requires checking two inequalities (x_1 vs. 4 and x_2 vs. $x_1 + 5$), and its disconnected nature is counter-intuitive for these data.

Conclusion.

Although QDA can be "better" when classes truly have different covariance structures and ample data are available, in this small-sample setting its estimated quadratic boundary is mis-shaped and overfits the training points. The linear classifier from part (a) is more aligned with the empirical clusters and is therefore the preferable rule here.