STAT 32950 Assignment 6

Bin Yu

May 11, 2025

Question 1

(a)

Using the following R code:

```
dmat <- matrix(
  c(
    0.00, 0.76, 2.97, 4.88, 3.86,
    0.76, 0.00, 0.80, 4.17, 1.96,
    2.97, 0.80, 0.00, 0.21, 1.51,
    4.88, 4.17, 0.21, 0.00, 0.51,
    3.86, 1.96, 1.51, 0.51, 0.00
  ),
  nrow = 5,
  byrow = TRUE,
  dimnames = list(
    paste0("Ch", 1:5),
    paste0("Ch", 1:5)
  )
)

d <- as.dist(dmat)

# Perform hierarchical clustering with three linkage methods
hc_single   <- hclust(d, method = "single")
hc_complete <- hclust(d, method = "complete")
hc_average  <- hclust(d, method = "average")

par(mfrow = c(1,3))
plot(hc_single,   main = "Single Linkage",   sub = "", xlab = "")
plot(hc_complete, main = "Complete Linkage", sub = "", xlab = "")
plot(hc_average,  main = "Average Linkage",  sub = "", xlab = "")
```

We get the following output plot:

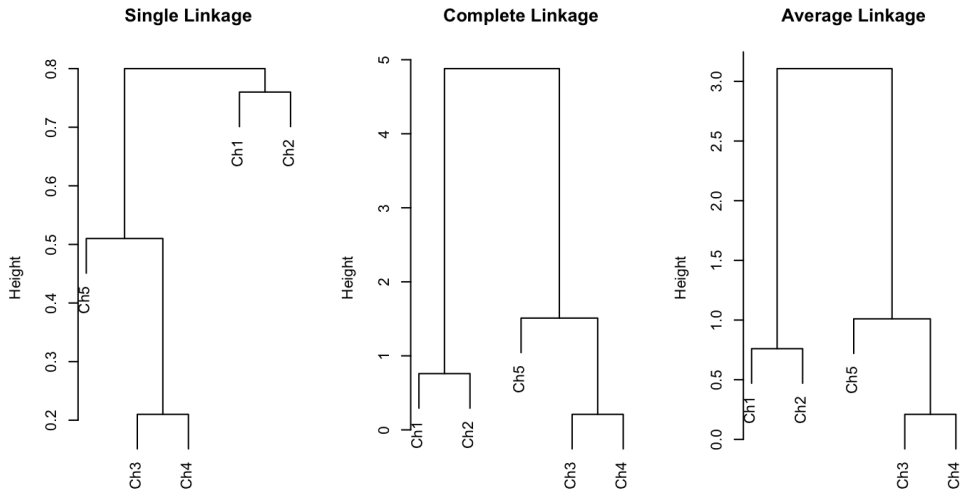

Figure 1: Output

All three linkage methods (single, complete, average) produce the similiar grouping: Chapters 1 and 2 form one tight cluster, and Chapters 3, 4 form one cluster, then join cluster 5, forming a larger cluster. However, the heights (vertical scales) on the three plots differ:

In single linkage, the algorithm first merges Chapters 3 and 4 at height ≈ 0.21, then adds Chapter 5 at height ≈ 0.51, and finally joins Chapters 1 and 2 at height ≈ 0.78. In complete linkage, it first merges Chapters 1 and 2 at height ≈ 0.76, then Chapters 3 and 4 at ≈ 0.21, next merges the $\{3, 4\}$ cluster with Chapter 5 at ≈ 1.51, and ultimately merges all chapters at ≈ 4.88. Average linkage lies between these two: it merges Chapters 3 and 4 at ≈ 0.21, then Chapters 1 and 2 at ≈ 0.76, next merges the $\{3, 4\}$ cluster with Chapter 5 at ≈ 1.01, and finally merges the $\{1, 2\}$ and $\{3, 4, 5\}$ clusters at ≈ 3.10.

Thus while the cluster topology is consistent, the absolute linkage heights are not the same across methods.

(b) Clusters for $k = 3$

Using the following R code:

```
clust_single   <- cutree(hc_single,   k = 3)
clust_complete <- cutree(hc_complete, k = 3)
clust_average  <- cutree(hc_average,  k = 3)

cat("Clusters (k=3) - Single Linkage:\n")
print(split(names(clust_single),   clust_single))
cat("\nClusters (k=3) - Complete Linkage:\n")
print(split(names(clust_complete), clust_complete))
cat("\nClusters (k=3) - Average Linkage:\n")
print(split(names(clust_average),  clust_average))
```

We get the following output:

```
> # Print cluster assignments
> cat("Clusters (k=3) - Single Linkage:\n")
Clusters (k=3) - Single Linkage:
> print(split(names(clust_single),   clust_single))
$`1`
[1] "Ch1"

$`2`
[1] "Ch2"

$`3`
[1] "Ch3" "Ch4" "Ch5"

> cat("\nClusters (k=3) - Complete Linkage:\n")

Clusters (k=3) - Complete Linkage:
> print(split(names(clust_complete), clust_complete))
$`1`
[1] "Ch1" "Ch2"

$`2`
[1] "Ch3" "Ch4"

$`3`
[1] "Ch5"

> cat("\nClusters (k=3) - Average Linkage:\n")

Clusters (k=3) - Average Linkage:
> print(split(names(clust_average),  clust_average))
$`1`
[1] "Ch1" "Ch2"

$`2`
[1] "Ch3" "Ch4"

$`3`
[1] "Ch5"

>
```

Cutting each tree at three clusters gives:

- **Single linkage**:
$$C_1 = \{\text{Ch1}\}, \quad C_2 = \{\text{Ch2}\}, \quad C_3 = \{\text{Ch3}, \text{Ch4}, \text{Ch5}\}.$$
- **Complete linkage**:
$$C_1 = \{\text{Ch1}, \text{Ch2}\}, \quad C_2 = \{\text{Ch3}, \text{Ch4}\}, \quad C_3 = \{\text{Ch5}\}.$$
- **Average linkage**:
$$C_1 = \{\text{Ch1}, \text{Ch2}\}, \quad C_2 = \{\text{Ch3}, \text{Ch4}\}, \quad C_3 = \{\text{Ch5}\}.$$

(c)

Linkage definitions Let $d(x, y)$ be the distance between observations x and y. If A and B are two clusters, define:

$$\textbf{Single linkage:} \quad d_{\min}(A,B) = \min_{x\in A,\ y\in B} d(x,y),$$
$$\textbf{Complete linkage:} \quad d_{\max}(A,B) = \max_{x\in A,\ y\in B} d(x,y),$$
$$\textbf{Average linkage:} \quad d_{\text{avg}}(A,B) = \frac{1}{|A|\ |B|}\sum_{x\in A}\sum_{y\in B} d(x,y).$$

- **Single linkage** tends to merge clusters along the shortest inter-point links. In our data, Chapters 3 and 4 are extremely close ($d \approx 0.21$), and Chapter 5 is only slightly farther from 4 ($d \approx 0.51$). Thus single linkage thus merges $3 \to 4 \to 5$ into one cluster, leaving 1 and 2 isolated until a much higher threshold (≈ 0.78).
- **Complete linkage** uses the maximum pairwise distance between clusters. Even though Ch3–Ch4 are close, Chapter 5 is relatively far from 3 ($d \approx 1.51$), so complete linkage first merges Ch3–Ch4 at ≈ 0.21 then Ch1–Ch2 at ≈ 0.76, then only later brings in Ch5 at a higher level (≈ 1.51), before finally uniting all at ≈ 4.88. This produces two tight two-element clusters $\{1,2\}$ and $\{3,4\}$, with 5 seperated.
- **Average linkage** averages all inter-cluster distances. It merges Ch3–Ch4 first (lowest average distance), then Ch1–Ch2, then the $\{3,4\}$ cluster with Ch5 (average distance ≈ 1.01). The resulting partition for $k = 3$ is consistent with complete linkage ($\{1,2\}, \{3,4\}, \{5\}$), reflecting that Chapters 1–2 and 3–4 each form relatively homogeneous groups under the average-distance criterion.

Takeaway: Single linkage will link points through successive small distances even if the overall cluster becomes different. Complete linkage avoids this by requiring all inter-point distances to be small, producing more compact clusters. Average linkage offers a middle ground, often agreeing with complete linkage on well-separated groups but without the extreme sensitivity to single long distances.

Question 2

(a)

Initial clusters and centroids

$$C_1^{(0)} = \{A, B\}, \quad C_2^{(0)} = \{C, D\},$$

The initial cluster centroids are the mean points:

$$\mu_1^{(0)} = \left(\tfrac{5+1}{2},\ \tfrac{-4+(-2)}{2}\right) = (3,\ -3), \quad \mu_2^{(0)} = \left(\tfrac{-1+3}{2},\ \tfrac{1+1}{2}\right) = (1,\ 1).$$

Assign each point to the nearest centroid

	$d^2\big(\cdot, \mu_1^{(0)}\big)$	$d^2\big(\cdot, \mu_2^{(0)}\big)$
$A = (5,-4)$	$(5-3)^2 + (-4+3)^2 = 5$	$(5-1)^2 + (-4-1)^2 = 41$
$B = (1,-2)$	$(1-3)^2 + (-2+3)^2 = 5$	$(1-1)^2 + (-2-1)^2 = 9$
$C = (-1,1)$	$(-1-3)^2 + (1+3)^2 = 32$	$(-1-1)^2 + (1-1)^2 = 4$
$D = (3,1)$	$(3-3)^2 + (1+3)^2 = 16$	$(3-1)^2 + (1-1)^2 = 4$

Since each point remains in its original cluster, the algorithm has converged.

Final clusters and centroids

$$C_1 = \{A, B\}, \quad \mu_1 = (3,\ -3), \qquad C_2 = \{C, D\}, \quad \mu_2 = (1,\ 1).$$

Within-cluster squared distances

squared distances to cluster centroids for each points:

$$\{d^2(A,\mu_1)=5,\ d^2(B,\mu_1)=5\},\quad \{d^2(C,\mu_2)=4,\ d^2(D,\mu_2)=4\}.$$

(b)

We use the following R code:

```
# Define the data
df <- data.frame(
  x1 = c(5,  1, -1,  3),
  x2 = c(-4, -2,  1,  1),
  row.names = c("A", "B", "C", "D")
)

set.seed(123)
km <- kmeans(df,2)

# Final cluster assignments
print(km$cluster)
# A B C D
# 2 1 1 1

# Cluster centroids
print(km$centers)
#   x1 x2
# 1  1  0
# 2  5 -4

# Squared distances to assigned centroid
dist2 <- rowSums((df - km$centers[km$cluster, ])^2)
print(dist2)
# A B C D
# 0 4 5 5
```

Results

	A	B	C	D
Cluster	2	1	1	1
Squared distance to centroids	0	4	5	5

$$\text{Centroids:}\quad \mu_1=(1,\,0),\qquad \mu_2=(5,\,-4).$$

(c)

The two procedures give different clusterings:

(a) Clusters $\{A,B\}$ and $\{C,D\}$ with within-cluster squared distances

$$d^2(A,\mu_1)=5,\ d^2(B,\mu_1)=5,\ d^2(C,\mu_2)=4,\ d^2(D,\mu_2)=4,$$

giving total sum of squares distances to cluster centroids:

$$\mathrm{SS}_a = 5 + 5 + 4 + 4 = 18.$$

(b) Clusters $\{B, C, D\}$ and $\{A\}$ with distances

$$d^2(A, \mu_2) = 0,\ d^2(B, \mu_1) = 4,\ d^2(C, \mu_1) = 5,\ d^2(D, \mu_1) = 5,$$

giving

$$\mathrm{SS}_b = 0 + 4 + 5 + 5 = 14.$$

Since $\mathrm{SS}_b = 14 < 18 = \mathrm{SS}_a$, the R function's solution in (b) achieves a strictly lower total sum of squares distances to cluster centroids. Thus the results are not the same, and the (b) clustering is preferable in terms of minimizing the sum of squared distances in Kmeans algorithm.

Question 3

(a)

Using the normalized track-time data for each country, we computed the full matrix of pairwise Euclidean distances using the following R code:

```
ladyrun = read.table("/Users/yubin/Desktop/Multivariate Analysis/ladyrun25.dat")
colnames(ladyrun)=c("Country","100m","200m","400m","800m","1500m","3000m","Marathon")

X <- ladyrun[, 2:8]
NormX <- as.matrix(X) %*% solve(diag(sqrt(diag(var(X)))))

distobs <- dist(NormX, method = "euclidean")

dm <- as.matrix(distobs)
diag(dm) <- NA

max_idx <- which(dm == max(dm, na.rm = TRUE), arr.ind = TRUE)[1, ]
min_idx <- which(dm == min(dm, na.rm = TRUE), arr.ind = TRUE)[1, ]

countries <- ladyrun$Country
max_pair  <- countries[max_idx]
min_pair  <- countries[min_idx]
max_dist  <- dm[max_idx[1], max_idx[2]]
min_dist  <- dm[min_idx[1], min_idx[2]]

cat("Maximum Euclidean distance =", round(max_dist, 3),
    "between", max_pair[1], "and", max_pair[2], "\n")
cat("Minimum Euclidean distance =", round(min_dist, 3),
    "between", min_pair[1], "and", min_pair[2], "\n")
```

.

Output:

```
cat("Maximum Euclidean distance =", round(max_dist, 3),
```

```
+     "between", max_pair[1], "and", max_pair[2], "\n")
Maximum Euclidean distance = 12.031 between SAM and CHN
> cat("Minimum Euclidean distance =", round(min_dist, 3),
+     "between", min_pair[1], "and", min_pair[2], "\n")
Minimum Euclidean distance = 0.304 between SUI and POR
```

We found

$$\max_{i<j} d_{ij} \;=\; 12.031,$$

which occurs between SAM and CHN, and

$$\min_{i<j} d_{ij} \;=\; 0.304,$$

which occurs between SUI and POR.

(b)

We use the following R code for 3 smallest cluster with $k = 7$ and $k = 8$:

```
hc_complete <- hclust(distobs, method = "complete")
k        <- 8
clust8   <- cutree(hc_complete, k = k)

sizes    <- table(clust8)
smallest_ids    <- names(sort(sizes))[1:3]
clusters_small  <- split(ladyrun$Country, clust8)[smallest_ids]
print(clusters_small)

hc_complete <- hclust(distobs, method = "complete")
k        <- 7
clust7   <- cutree(hc_complete, k = k)
sizes    <- table(clust7)
smallest_ids    <- names(sort(sizes))[1:3]
clusters_small  <- split(ladyrun$Country, clust7)[smallest_ids]
print(clusters_small)
```

Results:

```
> print(clusters_small)
$‘4‘
[1] "COK"

$‘6‘
[1] "KORN"

$‘7‘
[1] "PNG"

> print(clusters_small)
$‘4‘
[1] "COK"
```

```
$`6`
[1] "PNG"

$`7`
[1] "SAM"
```

For $k = 8$ clusters:

Cluster 4 : {COK},
Cluster 6 : {KORN},
Cluster 7 : {PNG}.

For $k = 7$ clusters:

Cluster 4 : {COK},
Cluster 6 : {PNG},
Cluster 7 : {SAM}.

(c)

We ran K-means for $K = 2, \ldots, 8$ and computed

$$P_K \;=\; 1 \;-\; \frac{\displaystyle\sum_{i=1}^{K} \sum_{x \in C_i} \|x - m_i\|^2}{\displaystyle\sum_{x} \|x - \bar{x}\|^2}$$

where m_i is the centroid of cluster C_i and $\bar{x}$ is the grand mean using the following code:

```
xbar <- colMeans(X)
totss <- sum(rowSums((X - matrix(xbar, nrow = nrow(X),
                                 ncol = ncol(X), byrow = TRUE))^2))

set.seed(42)
Kvals <- 2:8
PK    <- numeric(length(Kvals))

for (i in seq_along(Kvals)) {
  K  <- Kvals[i]
  km <- kmeans(X, centers = K, nstart = 25)

  # numerator =  within-cluster squared distances
  wss <- km$tot.withinss

  PK[i] <- 1 - wss / totss
}

results <- data.frame(K = Kvals, P_K = round(PK, 4))
print(results)

# Plot P_K vs. K
plot(results$K, results$P_K, type = "b", pch = 19, lwd = 2,
     xlab = "Number of clusters K",
```

```
     ylab = expression(P[K]),
     main = expression("Proportion of SS Explained" ~ P[K]))
```

Results:

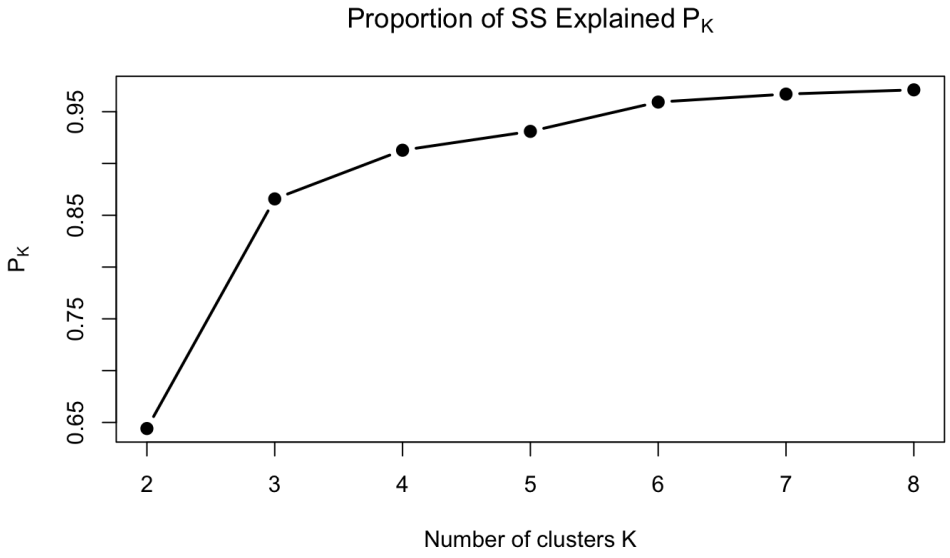

Figure 2: Results

The corresponding "elbow" plot (see above) shows a large gain in explained variance going from $K = 2$ to $K = 3$, and a small increase from $K = 3$ to $K = 4$. Beyond $K = 4$, the marginal improvements in P_K are quite small. Hence a sensible choice is

$$K = 3 \quad \text{or} \quad K = 4,$$

with $K = 3$ explaining about 85.8% of the total sum of squares and $K = 4$ explaining about 91.4%.

Question 4

Mixture of Two Densities: Likelihood and MLE

Let

$$f(x) = p_1\, f_1(x) + p_2\, f_2(x), \quad p_2 = 1 - p_1,$$

where

$$f_1(x) = \begin{cases} 2x, & 0 \le x \le 1, \\ 0, & \text{otherwise}, \end{cases} \qquad f_2(x) = \begin{cases} 2(1-x), & 0 \le x \le 1, \\ 0, & \text{otherwise}. \end{cases}$$

For a sample $\{x_1, \ldots, x_5\}$ the likelihood and log-likelihood are

$$L(p_1) = \prod_{i=1}^{5} \big[p_1 f_1(x_i) + (1-p_1) f_2(x_i)\big], \qquad \ell(p_1) = \sum_{i=1}^{5} \log\big[p_1 f_1(x_i) + (1-p_1) f_2(x_i)\big].$$

(a)

We have data: $\{0.1, 0.2, 0.3, 0.4, 0.7\}$

$$L_a(p_1) = \prod_{x\in\{0.1,0.2,0.3,0.4,0.7\}} \big[p_1 \cdot 2x + (1-p_1)\cdot 2(1-x)\big].$$

$$\begin{aligned} L_a(p_1) &= \prod_{x\in\{0.1,0.2,0.3,0.4,0.7\}} \big[p_1 \cdot 2x + (1-p_1)\cdot 2(1-x)\big] \\ &= \big[p_1\cdot 2\cdot 0.1 + (1-p_1)\cdot 2\cdot 0.9\big]\ \big[p_1\cdot 2\cdot 0.2 + (1-p_1)\cdot 2\cdot 0.8\big] \\ &\quad \big[p_1\cdot 2\cdot 0.3 + (1-p_1)\cdot 2\cdot 0.7\big]\ \big[p_1\cdot 2\cdot 0.4 + (1-p_1)\cdot 2\cdot 0.6\big] \\ &\quad \big[p_1\cdot 2\cdot 0.7 + (1-p_1)\cdot 2\cdot 0.3\big]. \end{aligned}$$

We have:

$$\begin{aligned} p_1\cdot 0.2 + (1-p_1)\cdot 1.8 &= 1.8 - 1.6p_1, \\ p_1\cdot 0.4 + (1-p_1)\cdot 1.6 &= 1.6 - 1.2p_1, \\ p_1\cdot 0.6 + (1-p_1)\cdot 1.4 &= 1.4 - 0.8p_1, \\ p_1\cdot 0.8 + (1-p_1)\cdot 1.2 &= 1.2 - 0.4p_1, \\ p_1\cdot 1.4 + (1-p_1)\cdot 0.6 &= 0.6 + 0.8p_1. \end{aligned}$$

Hence

$$L_a(p_1) = (1.8-1.6p_1)\,(1.6-1.2p_1)\,(1.4-0.8p_1)\,(1.2-0.4p_1)\,(0.6+0.8p_1).$$

(b)

Data: $\{0.1, 0.2, 0.3, 0.4, 0.9\}$

$$L_b(p_1) = \prod_{x\in\{0.1,0.2,0.3,0.4,0.9\}} \big[p_1 \cdot 2x + (1-p_1)\cdot 2(1-x)\big].$$

$$\begin{aligned} L_b(p_1) &= \prod_{x\in\{0.1,0.2,0.3,0.4,0.9\}} \big[p_1 \cdot 2x + (1-p_1)\cdot 2(1-x)\big] \\ &= \big[p_1\cdot 2\cdot 0.1 + (1-p_1)\cdot 2\cdot 0.9\big]\ \big[p_1\cdot 2\cdot 0.2 + (1-p_1)\cdot 2\cdot 0.8\big] \\ &\quad \times \big[p_1\cdot 2\cdot 0.3 + (1-p_1)\cdot 2\cdot 0.7\big]\ \big[p_1\cdot 2\cdot 0.4 + (1-p_1)\cdot 2\cdot 0.6\big] \\ &\quad \times \big[p_1\cdot 2\cdot 0.9 + (1-p_1)\cdot 2\cdot 0.1\big]. \end{aligned}$$

$$\begin{aligned} p_1\cdot 0.2 + (1-p_1)\cdot 1.8 &= 1.8 - 1.6\,p_1, \\ p_1\cdot 0.4 + (1-p_1)\cdot 1.6 &= 1.6 - 1.2\,p_1, \\ p_1\cdot 0.6 + (1-p_1)\cdot 1.4 &= 1.4 - 0.8\,p_1, \\ p_1\cdot 0.8 + (1-p_1)\cdot 1.2 &= 1.2 - 0.4\,p_1, \\ p_1\cdot 1.8 + (1-p_1)\cdot 0.2 &= 0.2 + 1.6\,p_1. \end{aligned}$$

Hence

$$L_b(p_1) = (1.8-1.6p_1)\,(1.6-1.2p_1)\,(1.4-0.8p_1)\,(1.2-0.4p_1)\,(0.2+1.6p_1).$$

(c)

Data: $\{0.1, 0.2, 0.3, 0.6, 0.9\}$

$$L_c(p_1) = \prod_{x\in\{0.1,0.2,0.3,0.6,0.9\}} \big[p_1 \cdot 2x + (1-p_1)\cdot 2(1-x)\big].$$

$$\begin{aligned} L_c(p_1) &= \prod_{x\in\{0.1,0.2,0.3,0.6,0.9\}} \big[p_1 \cdot 2x + (1-p_1)\cdot 2(1-x)\big] \\ &= \big[p_1\cdot 2\cdot 0.1 + (1-p_1)\cdot 2\cdot 0.9\big]\ \big[p_1\cdot 2\cdot 0.2 + (1-p_1)\cdot 2\cdot 0.8\big] \\ &\quad \times \big[p_1\cdot 2\cdot 0.3 + (1-p_1)\cdot 2\cdot 0.7\big]\ \big[p_1\cdot 2\cdot 0.6 + (1-p_1)\cdot 2\cdot 0.4\big] \\ &\quad \times \big[p_1\cdot 2\cdot 0.9 + (1-p_1)\cdot 2\cdot 0.1\big]. \end{aligned}$$

$$\begin{aligned} p_1\cdot 0.2 + (1-p_1)\cdot 1.8 &= 1.8 - 1.6\,p_1, \\ p_1\cdot 0.4 + (1-p_1)\cdot 1.6 &= 1.6 - 1.2\,p_1, \\ p_1\cdot 0.6 + (1-p_1)\cdot 1.4 &= 1.4 - 0.8\,p_1, \\ p_1\cdot 1.2 + (1-p_1)\cdot 0.8 &= 0.8 + 0.4\,p_1, \\ p_1\cdot 1.8 + (1-p_1)\cdot 0.2 &= 0.2 + 1.6\,p_1. \end{aligned}$$

Hence

$$L_c(p_1) = (1.8 - 1.6p_1)\,(1.6 - 1.2p_1)\,(1.4 - 0.8p_1)\,(0.8 + 0.4p_1)\,(0.2 + 1.6p_1).$$

(d) Log-Likelihood Plots

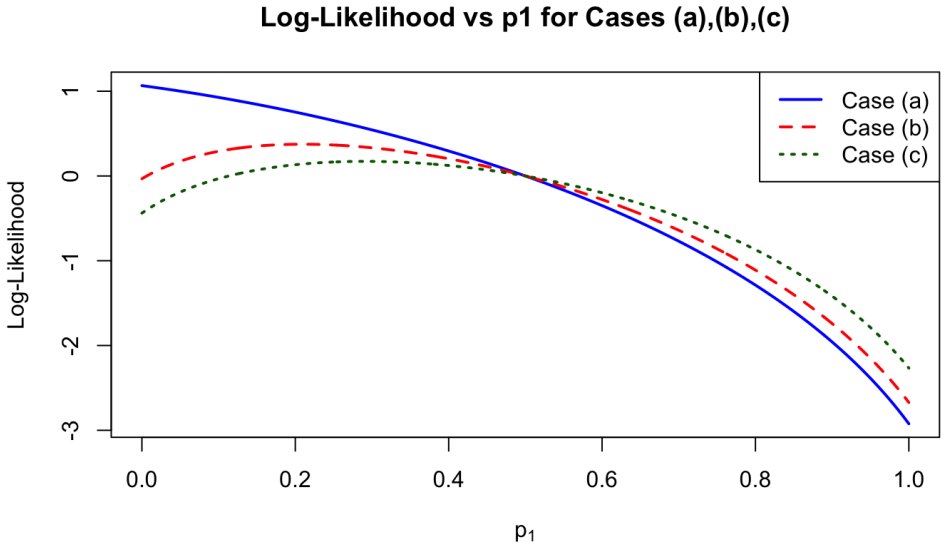

Figure 3: Log-likelihood vs. p_1 for cases (a), (b), and (c).

The figure shows the curves of $\ell_a(p_1)$, $\ell_b(p_1)$, and $\ell_c(p_1)$ as functions of p_1.

(e) Maximum Likelihood Estimates

After plotting the log–likelihood functions $\ell_a(p_1)$, $\ell_b(p_1)$, and $\ell_c(p_1)$ over $p_1 \in [0,1]$, we observe the following:

Case (a): $\hat{p}_1 = \arg\max_{p_1} \ell_a(p_1) \approx 0.00, \quad \hat{p}_2 = 1 - \hat{p}_1 \approx 1.00,$

since the curve is strictly decreasing on $(0,1]$, its maximum occurs at the boundary $p_1 = 0$.

Case (b): $\hat{p}_1 = \arg\max_{p_1} \ell_b(p_1) \approx 0.21, \quad \hat{p}_2 = 0.79,$

the log–likelihood attains a unique interior maximum near $p_1 = 0.21$, indicating about 21% weight on f_1.

Case (c): $\hat{p}_1 = \arg\max_{p_1} \ell_c(p_1) \approx 0.29, \quad \hat{p}_2 = 0.71,$

similarly, the interior peak at $p_1 \approx 0.29$ suggests roughly 29% weight on f_1.

Interpretation:

- In Case (a), all five observations are clustered in the lower half of $[0,1]$, where $f_2(x) = 2(1-x)$ dominates. Consequently, the mixture assigns zero weight to f_1 (i.e. $\hat{p}_1 = 0$), a degenerate boundary solution that may be unrealistic if one believes both components should contribute.
- In Case (b), the single large observation at 0.9 shifts the optimum away from the boundary. The MLE $\hat{p}_1 \approx 0.21$ balances the fit to small x (favoring f_2) against the lone large x (favoring f_1).
- In Case (c), two large values $\{0.6, 0.9\}$ further increase the estimated weight on f_1. The interior solution $\hat{p}_1 \approx 0.29$ reflects this by assigning almost 30% of the probability mass to the increasing-density component $f_1(x) = 2x$.

Thus, the estimate in case (a) lies at the boundary and may be degenerate and unreasonable, whereas the estimates in (b) and (c) are interior and thus more plausible.

Question 5

(a)

Initial imputation using column means (ignoring missing entries):

$$\bar{x}_1 = \frac{3+4+5}{3} = 4, \quad \bar{x}_2 = \frac{6+4+8}{3} = 6, \quad \bar{x}_3 = \frac{0+3+3}{3} = 2.$$

Imputed data matrix is:

$$\widetilde{X} = \begin{pmatrix} 3 & 6 & 0 \\ 4 & 4 & 3 \\ 4 & 8 & 3 \\ 5 & 6 & 2 \end{pmatrix}.$$

(b)

First compute the sample mean (ML estimate):

$$\tilde{\mu} = \frac{1}{4}\sum_{i=1}^{4} \widetilde{x}_i = \frac{1}{4}\begin{pmatrix} 3+4+4+5 \\ 6+4+8+6 \\ 0+3+3+2 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 2 \end{pmatrix}.$$

Next compute the ML estimate of the covariance,

$$\tilde{\Sigma} = \frac{1}{4}\sum_{i=1}^{4}(\widetilde{x}_i - \tilde{\mu})(\widetilde{x}_i - \tilde{\mu})^T.$$

$$\widetilde{x}_1 - \tilde{\mu} = \begin{pmatrix}-1\\0\\-2\end{pmatrix}, \quad \widetilde{x}_2 - \tilde{\mu} = \begin{pmatrix}0\\-2\\1\end{pmatrix}, \quad \widetilde{x}_3 - \tilde{\mu} = \begin{pmatrix}0\\2\\1\end{pmatrix}, \quad \widetilde{x}_4 - \tilde{\mu} = \begin{pmatrix}1\\0\\0\end{pmatrix}.$$

Their outer-products sum to

$$\sum_{i=1}^{4}(\widetilde{x}_i - \tilde{\mu})(\widetilde{x}_i - \tilde{\mu})^T = \begin{pmatrix}2&0&2\\0&8&0\\2&0&6\end{pmatrix},$$

Using ML formula for Σ : $\hat{\Sigma}_{ML} = \dfrac{n-1}{n}\,S = \dfrac{1}{n}\sum_{i=1}^{4}(\widetilde{x}_i - \tilde{\mu})(\widetilde{x}_i - \tilde{\mu})^T$, with $n = 4$

$$\tilde{\Sigma} = \frac{1}{4}\begin{pmatrix}2&0&2\\0&8&0\\2&0&6\end{pmatrix} = \begin{pmatrix}\frac{1}{2}&0&\frac{1}{2}\\0&2&0\\\frac{1}{2}&0&\frac{3}{2}\end{pmatrix}.$$

(c)

We start from the ML estimates (step (b)):

$$\tilde{\mu} = \begin{pmatrix}4\\6\\2\end{pmatrix}, \qquad \tilde{\Sigma} = \begin{pmatrix}\frac{1}{2}&0&\frac{1}{2}\\0&2&0\\\frac{1}{2}&0&\frac{3}{2}\end{pmatrix},$$

and the partially observed data

$$x_1 = (3,6,0), \quad x_2 = (4,4,3), \quad x_3 = (*,8,3), \quad x_4 = (5,6,2).$$

(i) Impute x_{31}

Partition $\tilde{\mu}$ and $\tilde{\Sigma}$ as

$$\tilde{\mu} = \begin{pmatrix}\mu_1\\\mu_{2:3}\end{pmatrix} = \begin{pmatrix}4\\6\\2\end{pmatrix}, \qquad \tilde{\Sigma} = \begin{pmatrix}\Sigma_{11}&\Sigma_{12}\\\Sigma_{21}&\Sigma_{22}\end{pmatrix} = \begin{pmatrix}\frac{1}{2}&[\,0&\frac{1}{2}\,]\\0&\begin{pmatrix}2&0\\0&\frac{3}{2}\end{pmatrix}\\\frac{1}{2}&\end{pmatrix}.$$

By the standard multivariate-normal regression formula,

$$E[X_1 \mid X_2 = 8, X_3 = 3] = \mu_1 + \Sigma_{12}\,\Sigma_{22}^{-1}\big((8,3)^T - \mu_{2:3}\big).$$

$$\Sigma_{22}^{-1} = \begin{pmatrix}\frac{1}{2}&0\\0&\frac{2}{3}\end{pmatrix}, \quad (8,3)^T - \mu_{2:3} = (2,1)^T,$$

$$\Sigma_{12}\,\Sigma_{22}^{-1}(2,1)^T = [\,0,\tfrac{1}{2}\,]\begin{pmatrix}\frac{1}{2}&0\\0&\frac{2}{3}\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix} = \frac{1}{3}.$$

Hence

$$\widetilde{x}_{31}^{(1)} = 4 + \tfrac{1}{3} = \frac{13}{3}.$$

We now have

$$x_3^{(1)} = \Big(\tfrac{13}{3},\, 8,\, 3\Big).$$

Re-compute the ML estimates on

$$x_1 = (3,6,0), \quad x_2 = (4,4,3), \quad x_3^{(1)} = \Big(\tfrac{13}{3}, 8, 3\Big), \quad x_4 = (5,6,2).$$

The updated mean is:

$$\mu^{(1)} = \frac{1}{4}\sum_{i=1}^{4} x_i = \begin{pmatrix} \frac{3+4+\frac{13}{3}+5}{4} \\ \frac{6+4+8+6}{4} \\ \frac{0+3+3+2}{4} \end{pmatrix} = \begin{pmatrix} 49/12 \\ 6 \\ 2 \end{pmatrix}.$$

The updated covariance matrix is:

$$\Sigma^{(1)} = \frac{1}{4}\sum_{i=1}^{4} \big(x_i - \mu^{(1)}\big)\big(x_i - \mu^{(1)}\big)^T = \begin{pmatrix} \frac{25}{48} & \frac{1}{6} & \frac{7}{12} \\ \frac{1}{6} & 2 & 0 \\ \frac{7}{12} & 0 & \frac{3}{2} \end{pmatrix}.$$

We use the following R code to verify:

```
X <- matrix(c(
  3, 6, 0,
  4, 4, 3,
  NA, 8, 3,
  5, 6, 2
), nrow = 4, byrow = TRUE)
colnames(X) <- paste0("X", 1:3)
rownames(X) <- paste0("obs", 1:4)

col_means <- colMeans(X, na.rm = TRUE)
X_imp <- X
for(j in seq_len(ncol(X_imp))) {
  X_imp[is.na(X_imp[, j]), j] <- col_means[j]
}

n <- nrow(X_imp)
mu_hat  <- colMeans(X_imp)
Sigma_hat <- cov(X_imp) * (n - 1) / n

mu1   <- mu_hat[1]
mu23  <- mu_hat[2:3]
S11   <- Sigma_hat[1, 1]
S12   <- Sigma_hat[1, 2:3]
S21   <- Sigma_hat[2:3, 1]
S22   <- Sigma_hat[2:3, 2:3]
obs3_23 <- X[3, 2:3]                          # (8,3)
x31_new <- mu1 + S12 %*% solve(S22) %*% (obs3_23 - mu23)
X_imp[3, 1] <- x31_new
print(x31_new)

mu_hat1   <- colMeans(X_imp)
Sigma_hat1 <- cov(X_imp) * (n - 1) / n

print(mu_hat1)
print(Sigma_hat1)
```

Output:

```
print(x31_new)
         [,1]
[1,] 4.333333
>
> print(mu_hat1)
      X1       X2       X3
4.083333 6.000000 2.000000

> print(Sigma_hat1)
          X1        X2        X3
X1 0.5208333 0.1666667 0.5833333
X2 0.1666667 2.0000000 0.0000000
X3 0.5833333 0.0000000 1.5000000
```

Thus, the result is correct.

(ii) Impute (x_{42}, x_{43})

Now treat X_1 observed and (X_2, X_3) missing.

Partition $\mu^{(1)}$, $\Sigma^{(1)}$ as

$$\mu^{(1)} = \begin{pmatrix} \mu_1^{(1)} \\ \mu_{2:3}^{(1)} \end{pmatrix} = \begin{pmatrix} 49/12 \\ 6 \\ 2 \end{pmatrix}, \quad \Sigma^{(1)} = \begin{pmatrix} \Sigma_{11}^{(1)} & \Sigma_{12}^{(1)} \\ \Sigma_{21}^{(1)} & \Sigma_{22}^{(1)} \end{pmatrix},$$

with

$$\Sigma_{11}^{(1)} = \tfrac{25}{48}, \quad \Sigma_{21}^{(1)} = \begin{pmatrix} \frac{1}{6} \\ \frac{7}{12} \end{pmatrix}, \quad \Sigma_{22}^{(1)} = \begin{pmatrix} 2 & 0 \\ 0 & \frac{3}{2} \end{pmatrix}.$$

Then

$$E\Big[\begin{pmatrix} X_2 \\ X_3 \end{pmatrix} \;\Big|\; X_1 = 5\Big] = \mu_{2:3}^{(1)} + \Sigma_{21}^{(1)} \big(\Sigma_{11}^{(1)}\big)^{-1} (5 - \mu_1^{(1)}),$$

where

$$(5 - \mu_1^{(1)}) = 5 - \tfrac{49}{12} = \tfrac{11}{12}, \quad \big(\Sigma_{11}^{(1)}\big)^{-1} = \tfrac{48}{25}.$$

Thus

$$\Sigma_{21}^{(1)} \; \tfrac{48}{25} \; \tfrac{11}{12} = \begin{pmatrix} \frac{1}{6} \\ \frac{7}{12} \end{pmatrix} \cdot \frac{44}{25} = \begin{pmatrix} \frac{22}{75} \\ \frac{77}{75} \end{pmatrix},$$

and

$$\widetilde{x}_{42}^{(2)} = 6 + \frac{22}{75} = \frac{472}{75}, \quad \widetilde{x}_{43}^{(2)} = 2 + \frac{77}{75} = \frac{227}{75}.$$

So

$$x_4^{(2)} = \Big(5,\ \tfrac{472}{75},\ \tfrac{227}{75}\Big).$$

Re-compute the ML estimates using the following R code:

```
mu1_1   <- mu_hat1[1]
mu23_1  <- mu_hat1[2:3]
S11_1   <- Sigma_hat1[1, 1]
S21_1   <- Sigma_hat1[2:3, 1]
obs4_1 <- X[4, 1]
```

```
x23_new_obs4 <- mu23_1 + S21_1 * (1/S11_1) * (obs4_1 - mu1_1)
X_imp[4, 2:3] <- as.numeric(x23_new_obs4)

print(x23_new_obs4)

mu_hat2    <- colMeans(X_imp)
Sigma_hat2 <- cov(X_imp) * (n - 1) / n

print(mu_hat2)
print(Sigma_hat2)
```

Output:

```
 print(x23_new_obs4)
      X2       X3
6.293333 3.026667

> print(mu_hat2)
      X1       X2       X3
4.083333 6.073333 2.256667
> print(Sigma_hat2)
          X1         X2         X3
X1 0.5208333 0.23388889 0.81861111
X2 0.2338889 2.01613333 0.05646667
X3 0.8186111 0.05646667 1.69763333
```

We have:

$$\mu^{(2)} = \frac{1}{4}\sum_{i=1}^{4} x_i = \begin{pmatrix} 49/12 \\ \frac{6+4+8+472/75}{4} \\ \frac{0+3+3+227/75}{4} \end{pmatrix} = \begin{pmatrix} \frac{49}{12} \\ \frac{911}{150} \\ \frac{677}{300} \end{pmatrix}.$$

Finally,

$$\Sigma^{(2)} = \frac{1}{4}\sum_{i=1}^{4}\big(x_i - \mu^{(2)}\big)\big(x_i - \mu^{(2)}\big)^T \approx \begin{pmatrix} 0.521 & 0.234 & 0.819 \\ 0.234 & 2.016 & 0.056 \\ 0.819 & 0.056 & 1.698 \end{pmatrix}.$$

Thus after one EM iteration our updated estimates are

$$\mu^{(2)} = \begin{pmatrix} 49/12 \\ 911/150 \\ 677/300 \end{pmatrix}, \quad \Sigma^{(2)} \approx \begin{pmatrix} 0.521 & 0.234 & 0.819 \\ 0.234 & 2.016 & 0.056 \\ 0.819 & 0.056 & 1.698 \end{pmatrix}.$$

Question 6

(a)

we have spectral decomposition of A:

$$A = P\,\mathrm{diag}(\lambda_1, \ldots, \lambda_p)\,P^T = \sum_{i=1}^{p} \lambda_i\, e_i e_i^T, \quad e_i^T e_j = \delta_{ij}.$$

and we have matrix square-root:

$$A^{1/2} = P\,\mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_p})\,P^T = \sum_{i=1}^{p} \sqrt{\lambda_i}\, e_i e_i^T.$$

$$\begin{aligned}
(A^{1/2})^T &= \Big(\sum_{i=1}^{p} \sqrt{\lambda_i}\, e_i e_i^T\Big)^T \\
&= \sum_{i=1}^{p} \Big(\sqrt{\lambda_i}\, e_i e_i^T\Big)^T && \text{By linearity of the transpose} \\
&= \sum_{i=1}^{p} \sqrt{\lambda_i}\, \big(e_i e_i^T\big)^T \\
&= \sum_{i=1}^{p} \sqrt{\lambda_i}\, \big(e_i^T\big)^T \big(e_i\big)^T && \text{By } (AB)^T = B^T A^T,\ A = e_i,\ B = e_i^T \\
&= \sum_{i=1}^{p} \sqrt{\lambda_i}\, e_i\, e_i^T && \text{By } (e_i^T)^T = e_i \\
&= A^{1/2}.
\end{aligned}$$

(b)

$$\begin{aligned}
A^{1/2}A^{1/2} &= \Big(\sum_{i=1}^{p} \sqrt{\lambda_i}\, e_i e_i^T\Big)\Big(\sum_{j=1}^{p} \sqrt{\lambda_j}\, e_j e_j^T\Big) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} \sqrt{\lambda_i}\,\sqrt{\lambda_j}\,(e_i e_i^T)(e_j e_j^T) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} \sqrt{\lambda_i\lambda_j}\; e_i\,(e_i^T e_j)\,e_j^T && \text{By } (AB)(CD) = A(BC)D \\
&= \sum_{i=1}^{p}\left[\sqrt{\lambda_i\lambda_i}\; e_i\,(e_i^T e_i)\,e_i^T + \sum_{\substack{j=1\\ j\neq i}}^{p} \sqrt{\lambda_i\lambda_j}\; e_i\,(e_i^T e_j)\,e_j^T\right] && \text{split into the terms } j = i \text{ and } j \neq i \\
&= \sum_{i=1}^{p}\Big[\sqrt{\lambda_i^2}\; e_i\,(1)\,e_i^T + \sum_{\substack{j=1\\ j\neq i}}^{p} \sqrt{\lambda_i\lambda_j}\; e_i\,(0)\,e_j^T\Big] && \text{By } e_i^T e_i = 1,\ e_i^T e_j = 0 \text{ for } j \neq i \\
&= \sum_{i=1}^{p} \sqrt{\lambda_i^2}\; e_i\, e_i^T \\
&= \sum_{i=1}^{p} \lambda_i\, e_i e_i^T && (\text{since } \sqrt{\lambda_i^2} = \lambda_i) \\
&= A
\end{aligned}$$

(c)

We want to show:

$$\left(A^{1/2}\right)^{-1} = \sum_{i=1}^{p} \frac{1}{\sqrt{\lambda_i}}\, e_i e_i^T.$$

To verify, compute the product is the identity:

$$A^{1/2}\left(\sum_{j=1}^{p} \tfrac{1}{\sqrt{\lambda_j}}\, e_j e_j^T\right) = I.$$

$$\begin{aligned}
&A^{1/2}\left(\sum_{j=1}^{p} \tfrac{1}{\sqrt{\lambda_j}}\, e_j e_j^T\right) \\
&= \left(\sum_{i=1}^{p} \sqrt{\lambda_i}\, e_i e_i^T\right)\left(\sum_{j=1}^{p} \tfrac{1}{\sqrt{\lambda_j}}\, e_j e_j^T\right) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} \sqrt{\lambda_i}\, \tfrac{1}{\sqrt{\lambda_j}}\, (e_i e_i^T)(e_j e_j^T) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_j}}\, e_i\, (e_i^T e_j)\, e_j^T \\
&= \sum_{i=1}^{p}\Big[\frac{\sqrt{\lambda_i}}{\sqrt{\lambda_i}}\, e_i\, (e_i^T e_i)\, e_i^T + \sum_{\substack{j=1\\ j\neq i}}^{p} \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_j}}\, e_i\, (e_i^T e_j)\, e_j^T\Big] \quad \text{(split into } j = i \text{ and } j \neq i\text{)} \\
&= \sum_{i=1}^{p}\Big[\frac{\sqrt{\lambda_i}}{\sqrt{\lambda_i}}\, e_i\, (1)\, e_i^T + \sum_{\substack{j=1\\ j\neq i}}^{p} \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_j}}\, e_i\, (0)\, e_j^T\Big] \qquad \text{(By } e_i^T e_i = 1,\ e_i^T e_j = 0 \text{ for } j \neq i\text{)} \\
&= \sum_{i=1}^{p}\Big[1 \cdot e_i e_i^T + \sum_{j\neq i} 0 \cdot (\cdots)\Big] = \sum_{i=1}^{p} e_i e_i^T = I.
\end{aligned}$$

Hence $\left(A^{1/2}\right)^{-1} = \sum_{i=1}^{p} \frac{1}{\sqrt{\lambda_i}}\, e_i e_i^T$, is proved.