HOME ASSIGNMENT 1

Multivariate Data Analysis

Simon Styrefors

Felix Karlsson

Oscar Lindquist

September, 2025

Part I: Theoretical problems

Exercise 2.32

Problem Description

You are given the random vector

$$\mathbf{X}' = [X_1, X_2, \ldots, X_5]$$

with mean vector

$$\mu_{\mathbf{X}} = [2, 4, -1, 3, 0]$$

and variance–covariance matrix

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} 4 & -1 & \frac{1}{2} & -\frac{1}{2} & 0 \\ -1 & 3 & 1 & -1 & 0 \\ \frac{1}{2} & 1 & 6 & 1 & -1 \\ -\frac{1}{2} & -1 & 1 & 4 & 0 \\ 0 & 0 & -1 & 0 & 2 \end{bmatrix}.$$

Partition $\mathbf{X}$ as

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} \mathbf{X}^{(1)} \\ \cdots \\ \mathbf{X}^{(2)} \end{bmatrix}$$

Let

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \end{bmatrix}.$$

Solution

Using (2–38):

$$\mu_{\mathbf{x}} = \begin{bmatrix} \boldsymbol{\mu}^{(1)} \\ \cdots \\ \boldsymbol{\mu}^{(2)} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \cdots \\ \mu_3 \\ \mu_4 \\ \mu_5 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ \cdots \\ -1 \\ 3 \\ 0 \end{bmatrix}$$

where $p = 5$, $q = 2$ and $p - q = 3$

Furthermore, using the definition (2-40) we get

$$\mathbf{\Sigma}_x = \begin{bmatrix} \mathbf{\Sigma}_{11} & \vdots & \mathbf{\Sigma}_{12} \\ \cdots & \ddots & \cdots \\ \mathbf{\Sigma}_{21} & \vdots & \mathbf{\Sigma}_{22} \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \vdots & \sigma_{13} & \sigma_{14} & \sigma_{15} \\ \sigma_{21} & \sigma_{22} & \vdots & \sigma_{23} & \sigma_{24} & \sigma_{25} \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\ \sigma_{31} & \sigma_{32} & \vdots & \sigma_{33} & \sigma_{34} & \sigma_{35} \\ \sigma_{41} & \sigma_{42} & \vdots & \sigma_{43} & \sigma_{44} & \sigma_{45} \\ \sigma_{51} & \sigma_{52} & \vdots & \sigma_{53} & \sigma_{54} & \sigma_{55} \end{bmatrix}$$

a) $E\left(\mathbf{X}^{(1)}\right)$

$$E\left(\mathbf{X}^{(1)}\right) = \boldsymbol{\mu}^{(1)} = \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix} = \begin{bmatrix}2 \\ 4\end{bmatrix}$$

b) $E\left(\mathbf{A}\mathbf{X}^{(1)}\right)$

Using (2–45):

$$E\left(A\mathbf{X}^{(1)}\right) = A\boldsymbol{\mu}^{(1)} = \begin{bmatrix}1 & -1 \\ 1 & 1\end{bmatrix}\begin{bmatrix}2 \\ 4\end{bmatrix} = \begin{bmatrix}-2 \\ 6\end{bmatrix}$$

c) $\text{Cov}\left(\mathbf{X}^{(1)}\right)$

$$\text{Cov}\left(\mathbf{X}^{(1)}\right) = \boldsymbol{\Sigma}_{11} = \begin{bmatrix}\sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22}\end{bmatrix} = \begin{bmatrix}4 & -1 \\ -1 & 3\end{bmatrix},$$

d) $\text{Cov}\left(\mathbf{A}\mathbf{X}^{(1)}\right)$

Using (2–45):

$$\text{Cov}(\boldsymbol{AX}_{(1)}) = \mathbf{A}\boldsymbol{\Sigma}_{11}\mathbf{A}' = \begin{bmatrix}1 & -1 \\ 1 & 1\end{bmatrix}\begin{bmatrix}4 & -1 \\ -1 & 3\end{bmatrix}\begin{bmatrix}1 & 1 \\ -1 & 1\end{bmatrix} = \begin{bmatrix}9 & 1 \\ 1 & 5\end{bmatrix}$$

e) $E\left(\mathbf{X}^{(2)}\right)$

$$E\left(\mathbf{X}^{(2)}\right) = \boldsymbol{\mu}^{(2)} = \begin{bmatrix}-1 \\ 3 \\ 0\end{bmatrix}$$

f) $E\left(\mathbf{B}\mathbf{X}^{(2)}\right)$

Using (2–45):

$$E\left(\mathbf{B}\mathbf{X}^{(2)}\right) = \mathbf{B}\boldsymbol{\mu}^{(2)} = \begin{bmatrix}1 & 1 & 1 \\ 1 & 1 & -2\end{bmatrix}\begin{bmatrix}-1 \\ 3 \\ 0\end{bmatrix} = \begin{bmatrix}2 \\ 2\end{bmatrix}$$

g) $\text{Cov}\left(\mathbf{X}^{(2)}\right)$

$$\text{Cov}\left(\mathbf{X}^{(2)}\right) = \boldsymbol{\Sigma}_{22} = \begin{bmatrix}6 & 1 & -1 \\ 1 & 4 & 0 \\ -1 & 0 & 2\end{bmatrix}$$

h) $\text{Cov}\left(\mathbf{BX}^{(2)}\right)$

Using 2–45

$$\text{Cov}\left(\mathbf{BX}^{(2)}\right) = \mathbf{B}\boldsymbol{\Sigma}_{22}\mathbf{B}' = \begin{bmatrix}1 & 1 & 1\\ 1 & 1 & -2\end{bmatrix}\begin{bmatrix}6 & 1 & -1\\ 1 & 4 & 0\\ -1 & 0 & 2\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & 1\\ 1 & -2\end{bmatrix} = \begin{bmatrix}12 & 9\\ 9 & 24\end{bmatrix}$$

i) $\text{Cov}\left(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}\right)$

$$\text{Cov}\left(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}\right) = \Sigma_{12} = \begin{bmatrix}\frac{1}{2} & -\frac{1}{2} & 0\\ 1 & -1 & 0\end{bmatrix}$$

j) $\text{Cov}\left(\mathbf{AX}^{(1)}, \mathbf{BX}^{(2)}\right)$

Using (2–45):

$$\text{Cov}\left(\mathbf{AX}^{(1)}, \mathbf{BX}^{(2)}\right) = \mathbf{A}\boldsymbol{\Sigma}_{12}\mathbf{B}' = \begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{2} & -\frac{1}{2} & 0\\ 1 & -1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & 1\\ 1 & -2\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$$

Exercise 3.18

Problem Description

Energy consumption in 2001, by state, from the major source

$$x_1 = \text{petroleum}, \quad x_2 = \text{natural gas}, \quad x_3 = \text{hydroelectric power}, \quad x_4 = \text{nuclear electric power}$$

is recorded in quadrillions (10^{15}) of BTUs (Source: *Statistical Abstract of the United States 2006*).
The resulting mean and covariance matrix are

$$\bar{\mathbf{x}} = \begin{bmatrix} 0.766 \\ 0.508 \\ 0.438 \\ 0.161 \end{bmatrix}, \qquad \mathbf{S} = \begin{bmatrix} 0.856 & 0.635 & 0.173 & 0.096 \\ 0.635 & 0.568 & 0.128 & 0.067 \\ 0.173 & 0.127 & 0.171 & 0.039 \\ 0.096 & 0.067 & 0.039 & 0.043 \end{bmatrix}$$

(a) Using summary statistics, determine the sample mean and variance of a state's total energy consumption for these major sources.
(b) Determine the sample mean and variance of the excess of petroleum consumption over natural gas consumption. Also find the sample covariance of this variable with the total variable in part a.

Solution

a)

The total energy consumption y is a *linear transformation* of the random vector $\mathbf{X} = (x_1, x_2, x_3, x_4)^\top$, since it can be written as

$$y = \mathbf{1}^\top X, \qquad \mathbf{1} = (1,1,1,1)^\top.$$

For any linear combination $y = a^\top X$, the mean and variance are given by

$$\bar{y} = a^\top \bar{\mathbf{x}}, \qquad s_y^2 = a^\top \mathbf{S} a,$$

where $\bar{\mathbf{x}}$ is the sample mean vector and $\mathbf{S}$ is the sample covariance matrix of X. In this case, setting $a = \mathbf{1}$ leads to:

$$\bar{y} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \bar{\mathbf{x}} = 1.873$$

$$s_y^2 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \mathbf{S} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 3.913$$

Thus, the sample mean of total energy consumption is 1.873 and the sample variance is 3.913 (std. dev. ≈ 1.978)

b)

Let the excess of petroleum over natural gas consumption be defined as

$$y = x_1 - x_2.$$

This can be written as a linear transformation of $X = (x_1, x_2, x_3, x_4)^\top$, with

$$y = a^\top X, \qquad a = (1, -1, 0, 0)^\top.$$

Using the same linear transformation formulas for the mean and variance as in part (a), we can compute $\bar{y}$ and s_y^2.

$$\bar{y} = \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix} \bar{\mathbf{x}} = 0.766 - 0.508 = 0.258,$$

$$s_y^2 = \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix} \mathbf{S} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = 0.154.$$

The sample mean of the excess petroleum over natural gas consumption is 0.258, and the corresponding sample variance is 0.154. To calculate the sample covariance of y with the total variable from a), denoted as: $T = x_1 + x_2 + x_3 + x_4$, we set

$$a = (1, -1, 0, 0), \qquad b = (1, 1, 1, 1).$$

Then

$$\mathrm{Cov}(y, T) = a \mathbf{S} b^\top = \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix} \mathbf{S} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Evaluating gives

$$\mathrm{Cov}(y, T) = 0.362.$$

Exercise 4.16

Problem Description

Let $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4$ be independent $N_p(\mu, \Sigma)$ random vectors.

(a) Find the marginal distributions for each of the random vectors

$$\mathbf{V}_1 = \tfrac{1}{4}\mathbf{X}_1 - \tfrac{1}{4}\mathbf{X}_2 + \tfrac{1}{4}\mathbf{X}_3 - \tfrac{1}{4}\mathbf{X}_4$$

and

$$\mathbf{V}_2 = \tfrac{1}{4}\mathbf{X}_1 + \tfrac{1}{4}\mathbf{X}_2 - \tfrac{1}{4}\mathbf{X}_3 - \tfrac{1}{4}\mathbf{X}_4.$$

(b) Find the joint density of the random vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ defined in (a).

Solution

a)

We know that if $X_1, X_2, \ldots, X_n$ are mutually independent and identically distributed $\sim N_p(\mu, \Sigma)$, then (by Result 4.8):

$$V_1 = c_1X_1 + c_2X_2 + \cdots + c_nX_n \sim N_p\left(\left(\sum_{j=1}^{n} c_j\right)\mu,\ \left(\sum_{j=1}^{n} c_j^2\right)\Sigma\right).$$

where

$$1 \leq j \leq n,$$

In this case:

$$V_1 = \tfrac{1}{4}X_1 - \tfrac{1}{4}X_2 + \tfrac{1}{4}X_3 - \tfrac{1}{4}X_4.$$

Then:

$$E(V_1) = \left(\tfrac{1}{4} - \tfrac{1}{4} + \tfrac{1}{4} - \tfrac{1}{4}\right)\mu = 0 \cdot \mu = 0,$$

$$\begin{aligned}\mathrm{Var}(V_1) &= \left(\left(\tfrac{1}{4}\right)^2 + \left(-\tfrac{1}{4}\right)^2 + \left(\tfrac{1}{4}\right)^2 + \left(-\tfrac{1}{4}\right)^2\right)\Sigma \\ &= 4 \cdot \tfrac{1}{16}\Sigma = \tfrac{1}{4}\Sigma.\end{aligned}$$

From this, the marginal distribution of V_1 is $N_p(0, \frac{1}{4}\Sigma)$.

In this exercise we also have:

$$V_2 = \tfrac{1}{4}X_1 + \tfrac{1}{4}X_2 - \tfrac{1}{4}X_3 - \tfrac{1}{4}X_4.$$

Then:

$$E(V_2) = \left(\tfrac{1}{4} + \tfrac{1}{4} - \tfrac{1}{4} - \tfrac{1}{4}\right)\mu = 0 \cdot \mu = 0,$$

$$\begin{aligned}\mathrm{Var}(V_2) &= \left(\left(\tfrac{1}{4}\right)^2 + \left(\tfrac{1}{4}\right)^2 + \left(-\tfrac{1}{4}\right)^2 + \left(-\tfrac{1}{4}\right)^2\right)\Sigma \\ &= 4 \cdot \tfrac{1}{16}\Sigma = \tfrac{1}{4}\Sigma.\end{aligned}$$

From this, the marginal distribution of V_2 is $N_p(0, \frac{1}{4}\Sigma)$.

b)

We know that if $X_1, X_2, \ldots, X_n$ are mutually independent and identically distributed $\sim N_p(\mu, \Sigma)$, then (by Result 4.8)

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \sim N_{2p}\left(\begin{bmatrix} \left(\sum_{j=1}^n c_j\right)\mu \\ \left(\sum_{j=1}^n b_j\right)\mu \end{bmatrix}, \begin{bmatrix} \left(\sum_{j=1}^n c_j^2\right)\Sigma & \left(\sum_{j=1}^n c_j b_j\right)\Sigma \\ \left(\sum_{j=1}^n b_j c_j\right)\Sigma & \left(\sum_{j=1}^n b_j^2\right)\Sigma \end{bmatrix} \right),$$

where $V_1 = c_1X_1 + c_2X_2 + \cdots + c_nX_n$, $V_2 = b_1X_1 + b_2X_2 + \cdots + b_nX_n$ and $1 \le j \le n$

For this problem we have:

$$V_1 = \tfrac{1}{4}X_1 - \tfrac{1}{4}X_2 + \tfrac{1}{4}X_3 - \tfrac{1}{4}X_4, \qquad V_2 = \tfrac{1}{4}X_1 + \tfrac{1}{4}X_2 - \tfrac{1}{4}X_3 - \tfrac{1}{4}X_4.$$

Computing the covariance term:

$$\left(\sum_{j=1}^4 c_j b_j\right)\Sigma = \left(\tfrac{1}{4}\cdot\tfrac{1}{4} + (-\tfrac{1}{4})\cdot\tfrac{1}{4} + \tfrac{1}{4}\cdot(-\tfrac{1}{4}) + (-\tfrac{1}{4})\cdot(-\tfrac{1}{4})\right)\Sigma = 0\cdot\Sigma = 0.$$

From part (a) we obtained that:

$$\left(\sum_{j=1}^4 c_j\right)\mu = \left(\sum_{j=1}^4 b_j\right)\mu = 0,$$

and

$$\left(\sum_{j=1}^4 c_j^2\right)\Sigma = \left(\sum_{j=1}^4 b_j^2\right)\Sigma = \tfrac{1}{4}\Sigma.$$

Hence, the joint distribution of V_1 and V_2 is

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \sim N_{2p}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{4}\Sigma & 0 \\ 0 & \frac{1}{4}\Sigma \end{bmatrix} \right).$$

Since (V_1, V_2) is jointly Gaussian, the fact that the off–diagonal blocks of the covariance matrix are zero implies that V_1 and V_2 are independent. Therefore, the joint density factors (from 2–28) as the product of the two marginal densities becomes:

$$f_{V_1,V_2}(v_1, v_2) = f_{V_1}(v_1)\, f_{V_2}(v_2), \quad V_1, V_2 \overset{\text{i.i.d.}}{\sim} N_p\big(0, \tfrac{1}{4}\Sigma\big).$$

Each marginal density is

$$f_{V_i}(v) = \frac{1}{(2\pi)^{p/2}\,|\frac{1}{4}\Sigma|^{1/2}} \exp\Big(-\tfrac{1}{2}\, v^\top (\tfrac{1}{4}\Sigma)^{-1} v\Big), \qquad i = 1, 2.$$

Equivalently, writing the joint density explicitly,

$$f_{V_1,V_2}(v_1, v_2) = \frac{1}{(2\pi)^p\,|\frac{1}{4}\Sigma|} \exp\Big(-\tfrac{1}{2}\Big[v_1^\top (\tfrac{1}{4}\Sigma)^{-1} v_1 + v_2^\top (\tfrac{1}{4}\Sigma)^{-1} v_2\Big]\Big) = \frac{4^p}{(2\pi)^p\,|\Sigma|} \exp\Big(-2\,[\,v_1^\top \Sigma^{-1} v_1 + v_2^\top \Sigma^{-1} v_2\,]\Big).$$

Part II: Gaussian model, Gaussian discriminant analysis, GMM, and EM algorithm in R.

Task 1

The purpose of this task is to investigate whether the dataset **scor** follows the assumption of multivariate normality and to identify potential outliers. We apply statistical tests and diagnostic plots from the MVN package.

```
library(MVN)
library(bootstrap)
library(mixtools)
data(scor, package = "bootstrap")
```

Normality tests

Two formal statistical tests were conducted:

Mardia's Test: Evaluates multivariate skewness and kurtosis. Significant skewness indicates deviation from normality, whereas kurtosis close to the expected value supports normality. Both components are required to confirm multivariate normality.

Henze–Zirkler Test: A test of multivariate normality based on a weighted integral of the squared difference between the empirical characteristic function and that of the normal distribution.

```
#Mardia
result = mvn(data = scor, mvn_test = "mardia")
result$multivariate_normality

#Henze-Zirker
result = mvn(data = scor, mvn_test = "hz")
result$multivariate_normality
```

```
            Test Statistic p.value     Method          MVN
1 Mardia Skewness    62.473   0.003 asymptotic   Not normal
2 Mardia Kurtosis     0.057   0.955 asymptotic       Normal
```

```
           Test Statistic p.value     Method          MVN
1 Henze-Zirkler     1.267  <0.001 asymptotic   Not normal
```

The normality tests provided clear evidence against the assumption of multivariate normality. Mardia's test showed a significant skewness statistic (62.473, p = 0.003), indicating asymmetry in the data distribution, while the kurtosis statistic (0.057, p = 0.955) was not significant, suggesting that tail behavior is consistent with normality. This combination implies that the dataset deviates from normality due to skewness rather than abnormal kurtosis.

The Henze–Zirkler test confirmed this conclusion, showing a test statistic of 1.267 with $p < 0.001$. These two test combined provides strong support for rejecting the null hypothesis of multivariate normality.

Diagnostic Plots

To visually inspect normality and detect potential outliers, the following plots were generated:

Chi-square Q-Q Plots: Compare squared Mahalanobis distances with theoretical chi-square quantiles. Data following a multivariate normal distribution should approximately align with the diagonal reference line.

Outlier Detection: The package implements methods based on robust Mahalanobis distances. Observations exceeding a chi-square threshold can be flagged as potential outliers, indicating departures from normality.

```
# Multivariate diagnostic plot with qq
multivariate_diagnostic_plot(
  scor,
  type = c("qq"),
  tol = 1e-25,
  use_population = TRUE
)
```

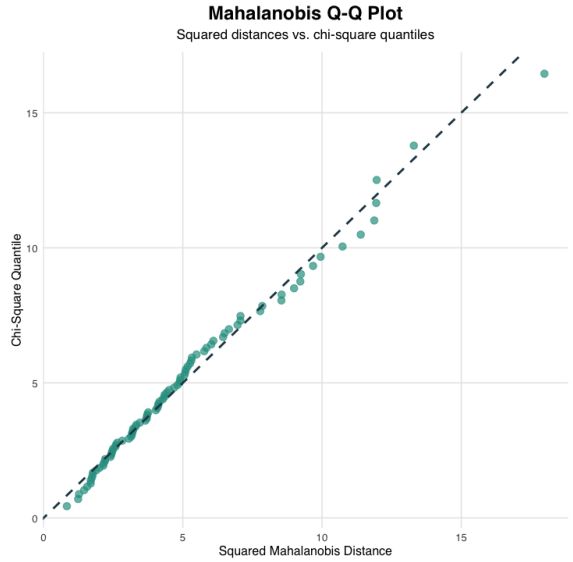

Figure 1: Multivariate diagnostic plot to asses multivariate normality

The majority of points lie close to the 45-degree reference line, indicating an approximate alignment with chi-square quantiles. However, deviations at the upper tail suggest that some observations do not follow the expected multivariate normal pattern.

```
# Plot outliers with quan
mv_outlier(
  scor,
  outlier = TRUE,
  qqplot = TRUE,
```

```
  alpha = 0.05,
  method = c("quan"),
  label = TRUE,
  title = "Chi-Square Q-Q Plot"
)
```

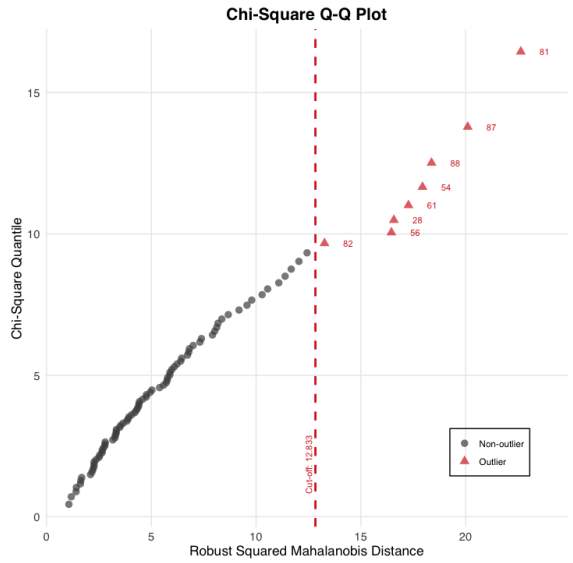

Figure 2: Chi-Square Q-Q Plot to detect outliers

A cut-off line was established at a chi-square quantile (≈ 12.83 at $\alpha = 0.05$). Several points beyond this threshold (e.g., observations 28, 54, 56, 61, 82, 87, 88) are flagged as potential outliers. These outliers deviate strongly from the expected distribution, explaining part of the non-normality observed in the statistical tests. In this test, method "quan" was used for the quantile method based on Mahalanobis distance; "adj" could also be used for the adjusted quantile method based on Mahalanobis distance.

Conclusions:

- Both Mardia's Skewness Test and the Henze–Zirkler Test indicate that the scor dataset does not satisfy the multivariate normality assumption.
- The Q-Q plots confirm deviations from theoretical chi-square quantiles.
- Multiple outliers were detected using robust Mahalanobis distances, which may account for the observed lack of multivariate normality.

Task 2:

The goal of this task is to predict whether a tumor is **benign (B)** or **malignant (M)** based on selected features from breast cancer diagnostic data. The dataset contains diagnostic results for 569 patients, with 30 features extracted from medical images of breast masses. For this exercise, we use three features: *radius*, *texture*, and *smoothness*. The data is divided into two parts: 469 patients are used as a training set (stored in `train.txt`), while the remaining 100 patients form the test set (stored in `test.txt`).

The classification is based on the **Gaussian mixture model (GMM)**, which assumes that the data points from each class are generated from a multivariate Gaussian distribution.

```
train_df <- read.table("train.txt", header = TRUE, sep = ",", stringsAsFactors
    = FALSE)
test_df  <- read.table("test.txt",  header = TRUE, sep = ",", stringsAsFactors
    = FALSE)
feat <- c("radius", "texture", "smoothness")
)
```

Class Labels

Each observation has a scalar label $y \in \{0, 1\}$:

$$y = \begin{cases} 1 & \text{malignant (M)} \\ 0 & \text{benign (B)} \, . \end{cases}$$

```
Make_Binary <- function(y) {
  y <- as.character(y)
  if (!all(y %in% c("M", "B"))) stop("Labels must be 'M' or 'B'")
  as.integer(y == "M")
}
y_tr <- Make_Binary(train_df$Diagnosis)
y_te <- Make_Binary(test_df$Diagnosis)
```

Feature Vectors

Each patient is described by the three-dimensional feature vector:

$$\mathbf{x} = (x_{\text{radius}}, x_{\text{texture}}, x_{\text{smoothness}})^\top \in \mathbb{R}^3.$$

```
# Creating Matrices
X_tr <- as.matrix(train_df[, feat, drop = FALSE])
X_te <- as.matrix(test_df[,  feat, drop = FALSE])

p_dim <- ncol(X_tr)      # number of features/variables

## 1) Split training rows by class
idx0 <- which(y_tr == 0)
idx1 <- which(y_tr == 1)
```

Class Priors

The prior probability of class membership is estimated from the training set:

$$\Pr(y = k) = \pi_k, \quad \sum_{k=1}^{2} \pi_k = 1, \quad \pi_k = \frac{n_k}{n}.$$

where n_k is the number of patients in class k, and n is the total number of patients in the training set.

```
## 2) Class priors pi_k = n_k / n (Percentage)
n0_tr <- length(idx0)
n1_tr <- length(idx1)
n_tr  <- nrow(X_tr)
prior0 <- n0_tr / n_tr
prior1 <- n1_tr / n_tr
```

Class Means and Covariances

Next, we estimate the **class-specific parameters** of the Gaussian mixture model. For each class $k \in \{0, 1\}$, we compute the sample mean vector μ_k and the sample covariance matrix Σ_k from the training data.

The class means are given by:

$$\mu_k = \frac{1}{n_k} \sum_{i:y_i=k} \mathbf{x}_i,$$

where n_k is the number of observations in class k.

The class covariance matrices are given by:

$$\Sigma_k = \frac{1}{n_k - 1} \sum_{i:y_i=k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top.$$

In practice, these parameters are estimated directly from the training data as shown below:

```
## 3) Class means
mu0_vec <- colMeans(X_tr[idx0, , drop = FALSE])
mu1_vec <- colMeans(X_tr[idx1, , drop = FALSE])

## 4) Class covariances
Sigma0 <- cov(X_tr[idx0, , drop = FALSE])
Sigma1 <- cov(X_tr[idx1, , drop = FALSE])
```

Class-Conditional Distributions

We assume that the feature vectors in each class follow a multivariate Gaussian distribution:

$$\mathbf{x} \mid y_k = 1 \sim \mathcal{N}_3(\mu_k, \Sigma_k),$$

with class-specific mean vector μ_k and covariance matrix Σ_k.

The Gaussian density is given by:

$$\phi(\mathbf{x}; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{3/2} |\Sigma_k|^{1/2}} \exp\left(-\tfrac{1}{2} (\mathbf{x} - \mu_k)^\top \Sigma_k^{-1} (\mathbf{x} - \mu_k) \right).$$

For classification, it is often more convenient to work with the logarithm of the class-conditional density function. Taking the log of $\phi(\mathbf{x};\mu_k,\Sigma_k)$

$$\log \phi(\mathbf{x};\mu_k,\Sigma_k) = -\tfrac{p}{2}\log(2\pi) - \tfrac{1}{2}\log|\Sigma_k| - \tfrac{1}{2}(\mathbf{x}-\mu_k)^\top \Sigma_k^{-1}(\mathbf{x}-\mu_k).$$

This form is computationally more stable and easier to manipulate when deriving discriminant functions, since it replaces exponents with quadratic forms.

```
# Constant derived from log-likelihood Multvariate Gaussian Density Function
log_const <- -0.5 * p_dim * log(2*pi)

qda_logscore <- function(X, mu_vec, Sigma, prior, log_const) {
  cholS  <- chol(Sigma)
  xcent  <- sweep(X, 2, mu_vec, FUN = "-")
  z      <- t(backsolve(cholS, t(xcent), transpose = TRUE))
  maha   <- rowSums(z^2)
  logdet <- sum(log(diag(cholS)))
  ll     <- log_const - logdet - 0.5 * maha
  score  <- log(prior) + ll
  list(score = score, ll = ll, maha = maha, logdet = logdet)
}

res0   <- qda_logscore(X_te, mu0_vec, Sigma0, prior0, log_const)
res1   <- qda_logscore(X_te, mu1_vec, Sigma1, prior1, log_const)
```

Joint Distribution of Features and Labels

The joint probability of observing a feature vector $\mathbf{x}$ and label y is:

$$\Pr(\mathbf{x},y) = \prod_{k=1}^{2}\Big(\pi_k\,\phi(\mathbf{x};\mu_k,\Sigma_k)\Big)^{y_k}.$$

Classification Rule

For a new test case with feature vector $\mathbf{x}$, the posterior probability is proportional to:

$$\Pr(y=k \mid \mathbf{x}) \propto \pi_k\,\phi(\mathbf{x};\mu_k,\Sigma_k).$$

Taking logarithms, the log-posterior score for class k is:

$$\text{score}_k(\mathbf{x}) = \log\pi_k - \tfrac{1}{2}\log|\Sigma_k| - \tfrac{1}{2}(\mathbf{x}-\mu_k)^\top\Sigma_k^{-1}(\mathbf{x}-\mu_k).$$

The classifier assigns the observation to the class with the highest score:

$$\hat{y}(\mathbf{x}) = \arg\max_{k\in\{0,1\}} \text{score}_k(\mathbf{x}).$$

```
score0 <- res0$score
score1 <- res1$score

# Pick class with larger posterior log-prob
yhat_qda <- as.integer(score1 > score0)   # 1 if class1 wins, else 0
acc_qda  <- mean(yhat_qda == y_te)
cat(sprintf("[QDA] Accuracy: %.3f\n", acc_qda))
print(table(truth = y_te, pred = yhat_qda))
```

Model Evaluation

The unknown parameters (π_k, μ_k, Σ_k) are estimated from the training set. Predictions are then made on the test set using the classification rule above. The accuracy of the classifier is defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of test samples}}.$$

Results:

Model Parameter Estimations from Training set

Covariance Matrix for Class 0 ($\hat{\Sigma}_0$)

```
                radius     texture   smoothness
radius      3.060564655 -0.38974525 -0.005319831
texture    -0.389745251 14.05021417 -0.010345276
smoothness -0.005319831 -0.01034528  0.000175353
```

Covariance Matrix for Class 1 ($\hat{\Sigma}_1$)

```
                radius      texture    smoothness
radius     10.163052279  1.278731809 -0.0040871513
texture     1.278731809 13.558412024 -0.0064242477
smoothness -0.004087151 -0.006424248  0.0001493527
```

Mean vector for features in Class 0 ($\hat{\mu}_0$)

```
     radius     texture  smoothness
12.19653901 17.63549645  0.09320482
```

Mean vector for features in Class 1 ($\hat{\mu}_1$)

```
    radius    texture smoothness
17.4620231 21.6172254  0.1029252
```

Prior probability of membership in Class 0 ($\hat{\pi}_0$)

```
0.6197802
```

Prior probability of membership in Class 1 ($\hat{\pi}_1$)

```
0.3802198
```

Accuracy

When applied to the independent test set, the classifier achieved an overall accuracy of:

```
Accuracy: 0.921
```

The confusion matrix is shown below:

```
     pred
truth  0  1
    0 71  4
    1  5 34
```

Out of 75 benign cases, the model correctly classified 71 and misclassified 4 as malignant. Out of 39 malignant cases, the model correctly classified 34 and misclassified 5 as benign.

Task 3:

3.1 Simulation from a bivariate Gaussian

We created 1000 samples from a bivariate normal distribution with

$$\mu = (2,3)^\top, \quad \Sigma = \begin{bmatrix} 1 & 1.4 \\ 1.4 & 4 \end{bmatrix}.$$

Figure 3 shows the scatter plot.

```
mu0 = c(2,3)
var1 = 1
var2 = 4
corr = 0.7

sigma = matrix(c(var1, sqrt(var1) * corr * sqrt(var2),
                 sqrt(var1) * corr * sqrt(var2), var2), nrow = 2)
sims = rmvnorm(n = 1000, mu = mu0, sigma = sigma)
plot(sims[,1], sims[,2], xlab = "X1", ylab = "X2",
     pch = 19, col = "purple")
```

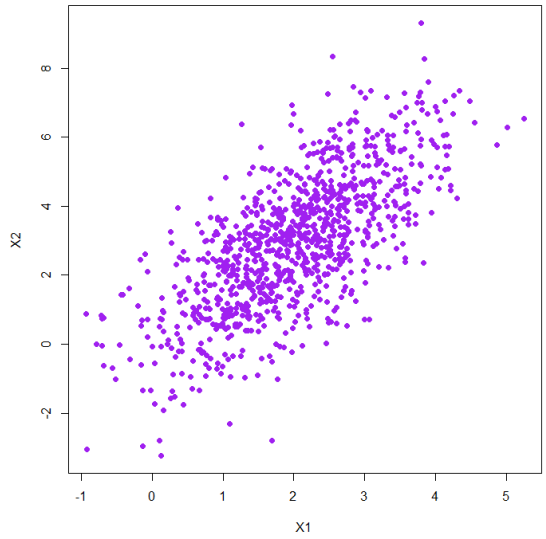

Figure 3: Scatter plot of 1000 simulations from Bivariate Gaussian Distribution

3.2 Simulation from a Gaussian Mixture Model

We generated 1000 samples from a Gaussian Mixture Model with latent variable $z_i \sim \text{Ber}(0.6)$. Conditional on z_i, the distributions are

$$X_i|z_i = 1 \sim N_2(\mu_1, \Sigma_1), \quad X_i|z_i = 0 \sim N_2(\mu_2, \Sigma_2),$$

where

$$\mu_1 = (2,3)^\top, \quad \mu_2 = (3,2)^\top.$$

The covariance matrices are

$$\Sigma_1 = \begin{bmatrix} 0.04 & 0.06 \\ 0.06 & 0.36 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0.16 & 0.06 \\ 0.06 & 0.09 \end{bmatrix}.$$

Figure 4 shows the resulting scatter plot.

```
mu1 = c(2,3)
mu2 = c(3,2)
sigma1 <- matrix(c(0.2^2, 0.5*0.2*0.6,
                   0.5*0.2*0.6, 0.6^2), nrow = 2)
sigma2 <- matrix(c(0.4^2, 0.5*0.4*0.3,
                   0.5*0.4*0.3, 0.3^2), nrow = 2)

z <- rbinom(1000, 1, 0.6)    # latent variable
sims <- matrix(0, nrow = 1000, ncol = 2)
sims[z == 1, ] <- rmvnorm(sum(z == 1), mu1, sigma1)
sims[z == 0, ] <- rmvnorm(sum(z == 0), mu2, sigma2)

plot(sims, xlab = "X1", ylab = "X2", pch = 19, col = "purple")
```

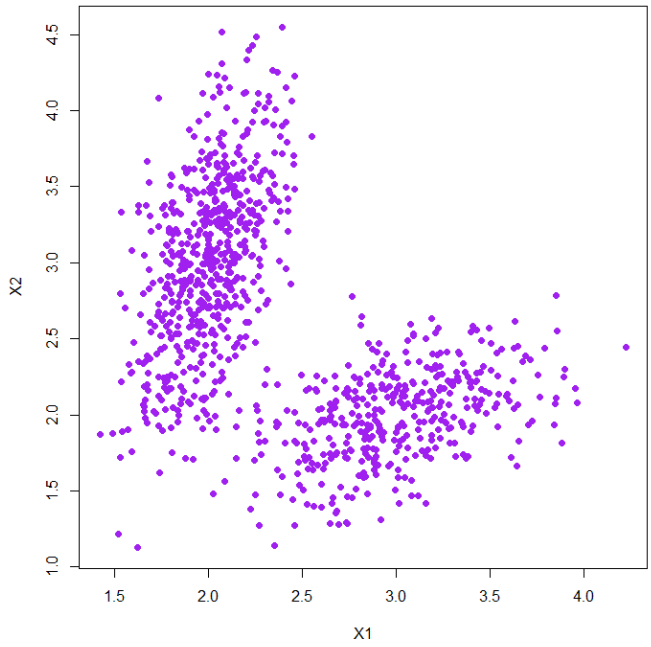

Figure 4: Scatter plot of 1000 samples from a Gaussian Mixture Model.

3.3 Estimation using EM algorithm

We fitted a Gaussian mixture model using the `mvnormalmixEM` function with $k = 2$. The EM algorithm converged in 15 iterations, and the estimated parameters are compared to the true parameters in Table 1.

```r
gmm <- mvnormalmixEM(sims, k = 2)
summary(gmm)
print(gmm$sigma)
```

```
         comp 1   comp 2
lambda 0.403769 0.596231
mu1    2.992829 1.999801
mu2    1.995967 3.016668
loglik at estimate:  -1205.936

[[1]]
           [,1]       [,2]
[1,] 0.15293814 0.05221295
[2,] 0.05221295 0.08829722

[[2]]
           [,1]       [,2]
[1,] 0.04051293 0.06223618
[2,] 0.06223618 0.33792923
```

	True values	Estimated values
Mixing proportions	$(0.6, 0.4)$	$(0.596, 0.404)$
Means	$\mu_1 = (2,3)$, $\mu_2 = (3,2)$	$\hat{\mu}_1 = (1.999, 2.993)$, $\hat{\mu}_2 = (3.017, 1.996)$
Covariances	$\Sigma_1 = \begin{bmatrix} 0.04 & 0.06 \\ 0.06 & 0.36 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.16 & 0.06 \\ 0.06 & 0.09 \end{bmatrix}$	$\hat{\Sigma}_1 = \begin{bmatrix} 0.153 & 0.052 \\ 0.052 & 0.088 \end{bmatrix}, \hat{\Sigma}_2 = \begin{bmatrix} 0.041 & 0.062 \\ 0.062 & 0.338 \end{bmatrix}$

Table 1: Comparison of true vs estimated GMM parameters.

Conclusion for Task 3

In this task the group simulated data from both a bivariate Gaussian distribution and a Multivariate Gaussian mixture model with two components. The scatter plots let us observe the different covariance structures and the presence of two clusters in the mixture model. When we fitted a Gaussian mixture model using the EM algorithm, the estimated mixing proportions, means, and covariance matrices were very close to the true values, with only small deviations due to sampling variability. The EM algorithm converged quickly (15 iterations), indicating a stable parameter estimation. In general, the results confirm that the EM procedure implemented in `mvnormalmixEM` is effective when recovering the underlying parameters of the mixture model.

Task 4:

In this task, we consider the following dataset that records the daily number of houses sold in a certain area in Sweden:

$$\{6,9,3,6,6,13,1,10,5,4\}.$$

Real estate markets exhibit clear seasonal variation. To capture this, we model the data as coming from a mixture of two Poisson distributions:

- λ_1: average number of houses sold per day in the off-season,
- λ_2: average number of houses sold per day in the peak season,
- π: proportion of days that belong to the off-season (so $1-\pi$ is the proportion in the peak season).

```
# Data and initial values
sales   = c(6, 9, 3, 6, 6, 13, 1, 10, 5, 4)
lambda1 = 5
lambda2 = 7
pi      = 0.5

# EM settings
max_iter  = 1000
tolerance = 1e-6

iterations = 0
converged = FALSE
```

The EM Algorithm

Because we do not know which data point comes from which season, we treat this as a latent-variable problem and apply the Expectation-Maximization (EM) algorithm to estimate the parameters π, λ_1, and λ_2. We start with initial values $\lambda_1^{(0)} = 5$, $\lambda_2^{(0)} = 7$, and $\pi^{(0)} = 0.5$.

E-step

In the E-step of the EM algorithm, we calculate the *responsibilities*, that is, the posterior probability that observation x_i was generated by component k. This probability is given by:

$$z_{i,k}^* = \Pr(z_{i,k} = 1 \mid x_i) = \frac{\pi_k f(x_i \mid \lambda_k)}{\sum_{j=1}^{K} \pi_j f(x_i \mid \lambda_j)}$$

```
while (!converged && iterations < max_iter) {
  iterations = iterations + 1

  # E-step
  p1 = dpois(sales, lambda1)
  p2 = dpois(sales, lambda2)
  z =(pi * p1) / (pi * p1 + (1 - pi) * p2)
```

M-step

The M-step involves maximizing the expected complete-data log-likelihood with respect to the parameters π, λ_1, and λ_2. Carrying out this maximization leads to the following parameter update rules:

$$\pi^{\text{new}} = \frac{1}{N}\sum_{i=1}^{N} z_i,$$

$$\lambda_1^{\text{new}} = \frac{\sum_{i=1}^{N} z_i x_i}{\sum_{i=1}^{N} z_i}, \qquad \lambda_2^{\text{new}} = \frac{\sum_{i=1}^{N} (1-z_i) x_i}{\sum_{i=1}^{N} (1-z_i)}.$$

Thus, π^{new} is updated as the average responsibility, while λ_1^{new} and λ_2^{new} are updated as weighted averages of the data, with weights given by the responsibilities.

```
  # M-step
  pi_new      = mean(z)
  l1_new = sum(z * sales) / sum(z)
  l2_new = sum((1 - z) * sales) / sum(1 - z)
```

Convergence and Results

In each iteration, we check for convergence by computing the maximum absolute change in the parameters λ_1, λ_2, and π. If the maximum change is smaller than the chosen tolerance, the algorithm is considered converged:

```
  # Convergence check (maximum parameter change)
  diffs = c(abs(l1_new - lambda1),
            abs(l2_new - lambda2),
            abs(pi_new - pi))

  if (max(diffs) < tolerance) {
    converged = TRUE
  }

  # Update parameters
  lambda1 = l1_new
  lambda2 = l2_new
  pi      = pi_new
}
```

After running the EM algorithm, the program reports whether it has converged, the number of iterations performed, and the final parameter estimates.

```
# Results
cat("Converged:", converged, "after", iterations, "iterations\n")
lambda1
lambda2
pi
```

The algorithm successfully converged after 149 iterations. The final parameter estimates are:

$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\pi}$
4.50	9.15	0.61

Table 2: Estimates of the unknown parameters from the random sample