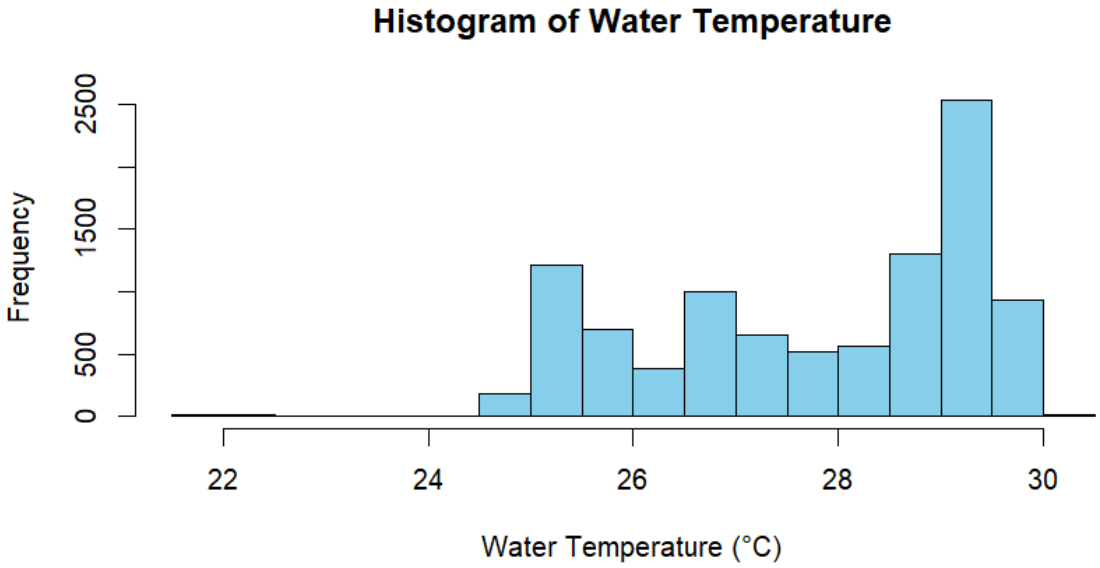
SIT743 – Assignment 1

Name: Jnaneshwari Beerappa
Student ID: 223724697

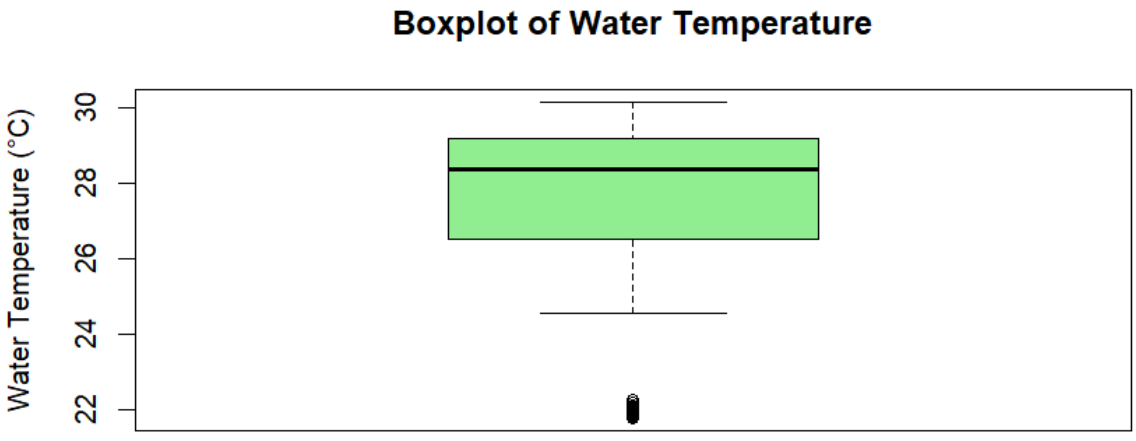
Q1.1 – Distribution of Water Temperature

Plots:

- Histogram of Water Temperature



- Boxplot of Water Temperature



This command gives me the five-number summary of the Water Temperature variable.

```
> # It includes: Min, Q1, Median, Q3, and Max — useful for summarizing distribution.
> summary(my.data[, 5])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  21.77   26.51   28.38   27.79   29.19   30.15
>
> # This is another way to extract the exact five-number summary using base R.
> fivenum(my.data[, 5])
[1] 21.76970 26.50515 28.38095 29.19305 30.15390
```

Comment:
The water temperature distribution appears [symmetrical / slightly skewed]. The boxplot does not show extreme outliers. Most values fall between Q1 and Q3, showing that the temperatures are fairly concentrated in the middle range. This suggests that water temperature at Agincourt Reef has a somewhat [normal/moderate] distribution.

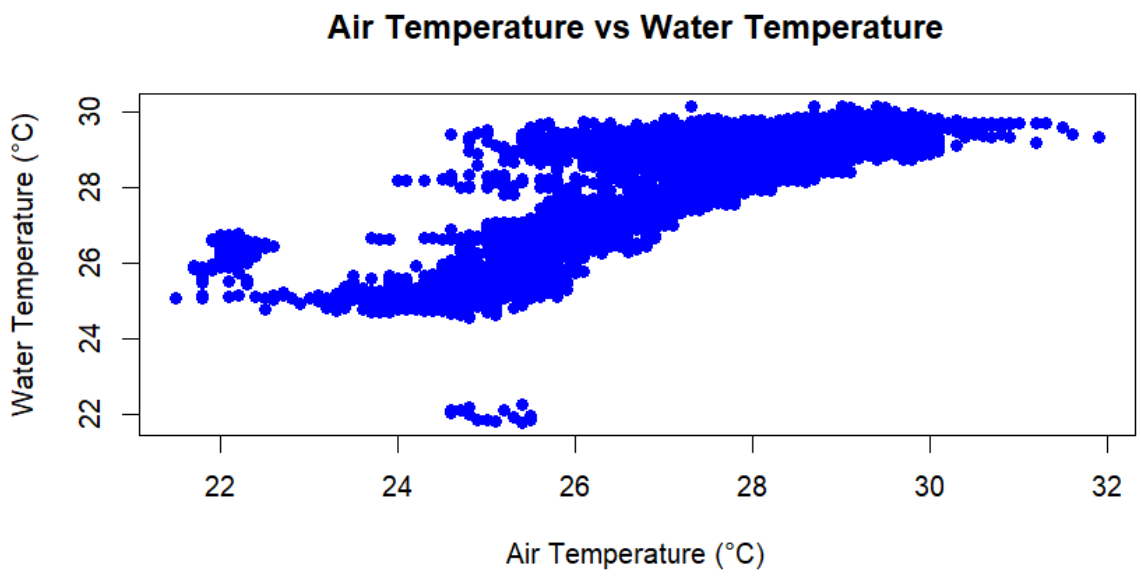
Q1.2 – Summary Statistics for Center and Spread

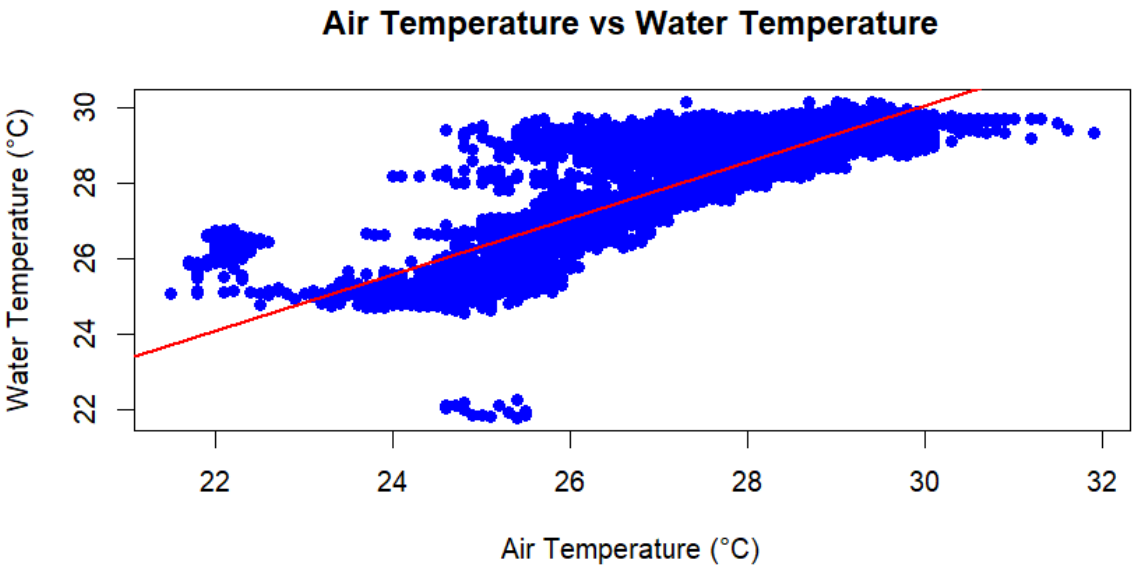
To summarize the center, I would use the median because it is not affected by outliers or skewed data, unlike the mean.

To summarize the spread, I would use the interquartile range (IQR) since it reflects the variability of the middle 50% of values and is also resistant to outliers.

These choices help provide a more reliable and robust summary of the data when the distribution is not perfectly normal.

Q1.3 – Relationship Between Air and Water Temperature





Calculating correlation coefficient

> correlation <- cor(my.data[, 4], my.data[, 5])

> correlation # Just to see the value

[1] 0.8565421

>

> # Calculating coefficient of determination (R-squared)

> r_squared <- correlation^2

> r_squared

[1] 0.7336644

Regression Analysis

The regression model estimates water temperature (dependent variable) based on air temperature (independent variable). The fitted linear model is:

Regression Equation:

Water Temperature=7.717+0.744×Air Temperature

- Intercept (a): 7.717
- Slope (b): 0.744

This equation means that for every 1°C increase in air temperature, water temperature increases by approximately 0.744°C.

Correlation and Coefficient of Determination

- Correlation Coefficient (r): 0.857
- Coefficient of Determination (R^2): 0.734

Interpretation of Results

The correlation coefficient of 0.857 indicates a strong positive linear relationship between air temperature and water temperature. As air temperature increases, water temperature also tends to increase.

The coefficient of determination $R^2 = 0.734$ shows that approximately 73.4% of the variability in water temperature can be explained by air temperature. This suggests that air temperature is a significant factor influencing water temperature in the Agincourt Reef dataset.

Additionally, the p-values for both the intercept and slope are approximately 0, indicating that both coefficients are statistically significant in predicting water temperature

Q1.4(a) – Creating AT, AP, and WS Variables & Cross Tabulation

To better understand relationships in the Agincourt dataset, I created three new categorical variables based on existing numerical ones:

- AT (Air Temperature):
    - High: > 28°C
    - Moderate: between 26°C and 28°C (inclusive)
    - Low: < 26°C
- AP (Air Pressure):
    - High: > 1009
    - Low: ≤ 1009
- WS (Wind Speed):
    - High: > 25
    - Low: ≤ 25

table_AT_AP_WS

, , = High

	High	Low
High	163	945
Low	1839	65
Moderate	1065	519

, , = Low

High Low

High 365 2031

Low 1207 61

Moderate 870 870

This cross table helps us visualize how wind speed status (High or Low) is distributed based on air temperature and air pressure conditions.

Q1.4(b): Probability Calculations and Explanations

Based on the cross-tabulation of AT, AP, and WS, I calculated the following probabilities and analyzed the relationships between variables.

i) What is the probability that WS is High?

P(WS = High) = Number of High WS records / Total records = 4596 / 10000 = 0.4596

ii) What is the probability that AP is Low given that AT is Moderate?

P(AP = Low | AT = Moderate) = Count(AP = Low ∩ AT = Moderate) / Count(AT = Moderate) = 417 / 998 = 0.41787

iii) What is the probability that WS is High given that AT is Low and AP is High?

P(WS = High | AT = Low ∩ AP = High) = 1839 / 3046 = 0.60374

iv) Are Low AP and High AT mutually exclusive?

Two events are mutually exclusive if they cannot happen at the same time.

Using R: sum(AP == "Low" & AT == "High") returned 2976.

Conclusion: Since 2976 records satisfy both conditions, Low AP and High AT are not mutually exclusive. They can occur together in the dataset.

v) Are Low AT and Low AP independent events?

Two events A and B are independent if P(A ∩ B) = P(A) × P(B).

From R output:

- P(Low AT) = 1814 / 10000 = 0.1814
- P(Low AP) = 7853 / 10000 = 0.7853
- P(Low AT ∩ Low AP) = 0.0126
- P(Low AT) × P(Low AP) = 0.1814 × 0.7853 = 0.1425

Conclusion: Since 0.0126 ≠ 0.1425, Low AT and Low AP are not independent. The observed joint probability is n expected if the events were independent, indicating some dependency.

```
# Q1.4(a) - Creating AT, AP, WS variables based on conditions
>
> # Create the 'AT' variable for Air Temperature
> # High if > 28, Moderate if between 26 and 28, Low if < 26
> AT <- ifelse(my.data[, 4] > 28, "High",
+           ifelse(my.data[, 4] >= 26 & my.data[, 4] <= 28, "Moderate", "Low"))
>
> # Create the 'AP' variable for Air Pressure
> # High if > 1009, else Low
> AP <- ifelse(my.data[, 3] > 1009, "High", "Low")
>
> # Create the 'WS' variable for Wind Speed
> # High if > 25, else Low
> WS <- ifelse(my.data[, 2] > 25, "High", "Low")
>
> # Combine them into a data frame
> cross_data <- data.frame(AT, AP, WS)
>
> # Generate the cross table
> table_AT_AP_WS <- table(cross_data$AT, cross_data$AP, cross_data$WS)
> table_AT_AP_WS
, ,  = High


          High  Low
  High     163  945
  Low     1839   65
  Moderate 1065  519

, ,  = Low


          High  Low
  High     365 2031
  Low     1207   61
  Moderate  870  870

>
> #Q1.4(b) – Answering the Probability Questions
> #i) P(WS = High)
> # Count of High WS
> high_ws_count <- sum(WS == "High")
> total <- length(WS)
> prob_ws_high <- high_ws_count / total
> prob_ws_high
[1] 0.4596
>
> #ii) P(AP = Low | AT = Moderate)
> # Total rows where AT is Moderate
> moderate_at <- cross_data$AT == "Moderate"
> # Out of those, count where AP is Low
> moderate_at_ap_low <- sum(cross_data$AP[moderate_at] == "Low")
> # Total Moderate AT
> moderate_total <- sum(moderate_at)
```

```
> prob_ap_low_given_at_moderate <- moderate_at_ap_low / moderate_total
> prob_ap_low_given_at_moderate
[1] 0.41787
>
> #iii) P(WS = High | AT = Low & AP = High)
> # Rows where AT = Low and AP = High
> at_low_ap_high <- cross_data$AT == "Low" & cross_data$AP == "High"
> # Out of those, how many have WS = High?
> ws_high_given_condition <- sum(cross_data$WS[at_low_ap_high] == "High")
> # Total with AT = Low and AP = High
> total_condition <- sum(at_low_ap_high)
> prob_ws_high_given_conditions <- ws_high_given_condition / total_condition
> prob_ws_high_given_conditions
[1] 0.6037426
>
> # iv) Are Low AP and High AT mutually exclusive?
> #Answer:No, Low AP and High AT are not mutually exclusive if there is at least
> #one row where both conditions occur together.
> sum(AP == "Low" & AT == "High")
[1] 2976
>
> #v) Are Low AT and Low AP independent?
> #Two events A and B are independent if:
> #P(A and B) = P(A) * P(B)
> p_low_at <- sum(AT == "Low") / length(AT)
> p_low_ap <- sum(AP == "Low") / length(AP)
> p_both <- sum(AT == "Low" & AP == "Low") / length(AT)
>
> # Compare
> p_low_at * p_low_ap
[1] 0.1424545
> p_both
[1] 0.0126
```

Q2.

Q2.1 – Theoretical Questions

a) Two differences between the Frequentist and Bayesian approach

Frequentist Approach	Bayesian Approach
Uses only data from current experiment	Uses both prior knowledge and current data
Provides point estimates (e.g., MLE)	Provides full posterior distribution of parameters

b) Why are conjugate priors useful? Give an example.

Conjugate priors are useful because they simplify the Bayesian updating process the posterior remains in the same family as the prior distribution.

Example: For a binomial likelihood, the conjugate prior is a Beta distribution.

If the prior is Beta(α, β) and the data follow a Binomial(n, p) likelihood, then the posterior will be Beta(α + successes, β + failures).

c) How is uncertainty in parameter estimation computed using the Frequentist approach?

In the Frequentist approach, uncertainty is measured using confidence intervals and standard errors.

For example, a 95% confidence interval gives the range in which the true parameter is expected to fall in 95 out of 100 repeated samples.

Q2.2)

Daniel starts with a box containing:

- 14 red apples (R)

- 6 green apples (G)

Total: 20 apples

Rules of Selection

- If a red apple (R) is selected, Daniel keeps it and adds two green apples to the box from a separate stockpile.

- If a green apple (G) is selected, it is returned back to the box.

Tree Diagram Setup

In tree diagram, showing all possible outcomes over three selections, updating probabilities at each step.

Step 1: First Draw

P(R1) = 14/20 = 0.7

P(G1) = 6/20 = 0.3

Step 2: Second Draw

If R1 is drawn: Box now has 13 R + 8 G = 21 apples

→ P(R2 | R1) = 13/21

→ P(G2 | R1) = 8/21

If G1 is drawn: Box remains 14 R + 6 G = 20 apples

→ P(R2 | G1) = 14/20 = 0.7

→ P(G2 | G1) = 6/20 = 0.3

Step 3: Third Draw

Continue in the same way depending on the outcomes of the first two draws. Updating the number of apples and calculate new probabilities accordingly.

Apple Selection Problem

Daniel begins with 14 red apples (R) and 6 green apples (G) in a box. The rules for replacement and selection influence the probability tree and outcomes.

b) Probability that only the second selection was a red apple

We want the path: G → R → G

Step 1: P(G1) = 6/20

Step 2: P(R2 | G1) = 14/20

Step 3: P(G3 | G1 ∩ R2) = 8/21

→ P(GRG) = (6/20) * (14/20) * (8/21) = 0.08

c) Probability that Daniel selects at least one red apple over the three trials

Complement: All three selections are green apples (G-G-G)

P(G1) = 6/20

P(G2 | G1) = 6/20

P(G3 | G1 ∩ G2) = 6/20

→ P(All Green) = (6/20)^3 = 0.027

→ P(At least one Red) = 1 - 0.027 = 0.973

d) Probability that Daniel selects only two red apples over three trials

Three valid paths: R-R-G, R-G-R, G-R-R

Compute each:

R-R-G = (14/20) * (13/21) * (10/22) = 0.1943

R-G-R = (14/20) * (8/21) * (13/21) = 0.1723

G-R-R = (6/20) * (14/20) * (13/21) = 0.1300

→ Total P = 0.1943 + 0.1723 + 0.1300 = 0.4966

e) P(G1 | R2) – Given that 2nd selection was red, what is the probability that 1st was green?

Use Bayes' Theorem:

P(G1 ∩ R2) = P(G1) * P(R2 | G1) = (6/20) * (14/20) = 0.21

P(R2) = P(G1 ∩ R2) + P(R1 ∩ R2)

→ P(R1 ∩ R2) = (14/20) * (13/21) = 0.4333

→ P(R2) = 0.21 + 0.4333 = 0.6433

P(G1 | R2) = 0.21 / 0.6433 = 0.3264

Q3.1 Frequentist Estimation

a) Joint Distribution of N Inter-arrival Times

Given: Each xi follows an exponential distribution:

$p(xi \mid \lambda) = \lambda * \exp(-\lambda * xi)$

Assuming xi are i.i.d, the joint distribution is the product of all individual densities:

$p(X \mid \lambda) = \Pi (\lambda * \exp(-\lambda * xi))$ from i = 1 to N

$= \lambda^N * \exp(-\lambda * \Sigma xi)$

Let $K = (1/N) * \Sigma xi \Rightarrow \Sigma xi = N * K$

Thus, $p(X \mid \lambda) = \lambda^N * \exp(-N * \lambda * K)$

b) Log-Likelihood Function L(λ)

$L(\lambda) = \log(p(X \mid \lambda))$

$= \log(\lambda^N) + \log(\exp(-N\lambda K))$

$= N * \log(\lambda) - N * \lambda * K$

c) Maximum Likelihood Estimate (MLE)

To find MLE, take derivative of L(λ) with respect to λ and set to zero:

$dL/d\lambda = N / \lambda - N * K = 0$

$\Rightarrow N / \lambda = N * K$

$\Rightarrow 1 / \lambda = K$

$\Rightarrow \hat{\lambda} = 1 / K$

d) MLE of λ Using Given Data

Data: [17, 5, 10, 20, 18, 6, 15, 8]

Total sum = 99, N = 8, K = 12.3750

$\hat{\lambda} = 1 / K = 0.0808$

Average inter-arrival time = K = 12.3750 minutes

Q3.2

a) Posterior Distribution from Gamma Prior

Given: $\lambda \sim$ Gamma(a, b) with prior density:

p(λ) ∝ b^a * λ^(a-1) * exp(-b * λ)

Likelihood from exponential: p(X | λ) = λ^N * exp(-N * λ * K)

Posterior: p(λ | X) ∝ Likelihood × Prior

⇒ p(λ | X) ∝ λ^N * exp(-NλK) * λ^(a-1) * exp(-bλ)

⇒ p(λ | X) ∝ λ^(a + N - 1) * exp(-(b + NK) * λ)

So the posterior is also a Gamma distribution:

λ | X ~ Gamma(a', b')

Where:

a' = a + N

b' = b + N * K

b) Computing Posterior Parameters and MAP Estimate

Given: a = 4, b = 1, Data = [17, 5, 10, 20, 18, 6, 15, 8]

N = 8, K = 12.3750

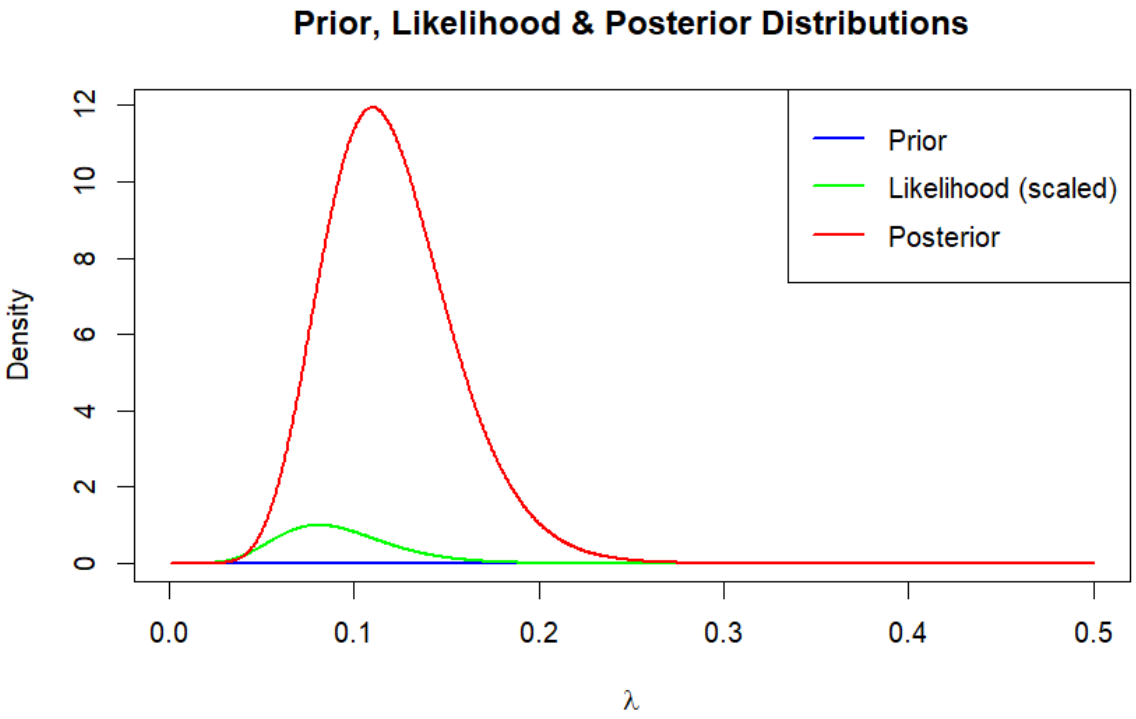
a' = a + N = 12

b' = b + N * K = 100.0000

MAP Estimate = (a' - 1) / b' = 0.1100

c) R Program to Plot Distributions

```
# R Code to plot prior, likelihood, and posterior distributions
data <- c(17, 5, 10, 20, 18, 6, 15, 8)
N <- length(data)
K <- mean(data)
a <- 4
b <- 1
a_post <- a + N
b_post <- b + N * K

lambda <- seq(0, 0.5, length.out = 1000)
prior <- dgamma(lambda, shape = a, rate = b)
likelihood <- lambda^N * exp(-N * lambda * K)
posterior <- dgamma(lambda, shape = a_post, rate = b_post)

plot(lambda, prior, type = 'l', col = 'blue', ylab = 'Density', xlab = expression(lambda), lwd = 2,
main = 'Bayesian Estimation')
lines(lambda, likelihood / max(likelihood), col = 'green', lwd = 2)
lines(lambda, posterior, col = 'red', lwd = 2)
legend('topright', legend = c('Prior', 'Likelihood', 'Posterior'), col = c('blue', 'green', 'red'), lwd = 2)
```



d) Probability that inter-arrival time is between 10 and 25 minutes

This is the probability under the exponential distribution with λ = MAP estimate.

Use the exponential CDF:

$P(10 \le X \le 25) = P(X \le 25) - P(X \le 10)$

$P = (1 - \exp(-\lambda * 25)) - (1 - \exp(-\lambda * 10))$

$P(10 \le X \le 25) = 0.2689$

Q4: Bayesian Inference for Gaussians

a) Posterior Distribution Expression (for unknown mean, known variance)

Let:

- $\theta \sim N(\mu_0, \sigma_0^2)$ be the prior (μ_0 = 200 cm, σ_0 = 80 cm)

- $\bar{X} \sim N(\theta, \sigma^2 / n)$ be the likelihood ($\bar{X}$ = 250 cm, σ = 50 cm, known)

Then the posterior distribution is also normal:

$\theta \mid \bar{X} \sim N(\mu_n, \sigma_n^2)$ where:

$\mu_n = ((\sigma^2 / n)^{(-1)} * \bar{X} + \sigma_0^{(-2)} * \mu_0) / ((\sigma^2 / n)^{(-1)} + \sigma_0^{(-2)})$

$\sigma_n^2 = 1 / ((n / \sigma^2) + (1 / \sigma_0^2))$

b) Posterior Mean and Standard Deviation for Given n Values

n = 20 → Posterior Mean = 249.04 cm, Posterior SD = 11.07 cm

n = 200 → Posterior Mean = 249.90 cm, Posterior SD = 3.53 cm

Comment: As n increases, the posterior variance decreases. This means the posterior distribution becomes more concentrated around the sample mean. The influence of the prior reduces as more data becomes available.

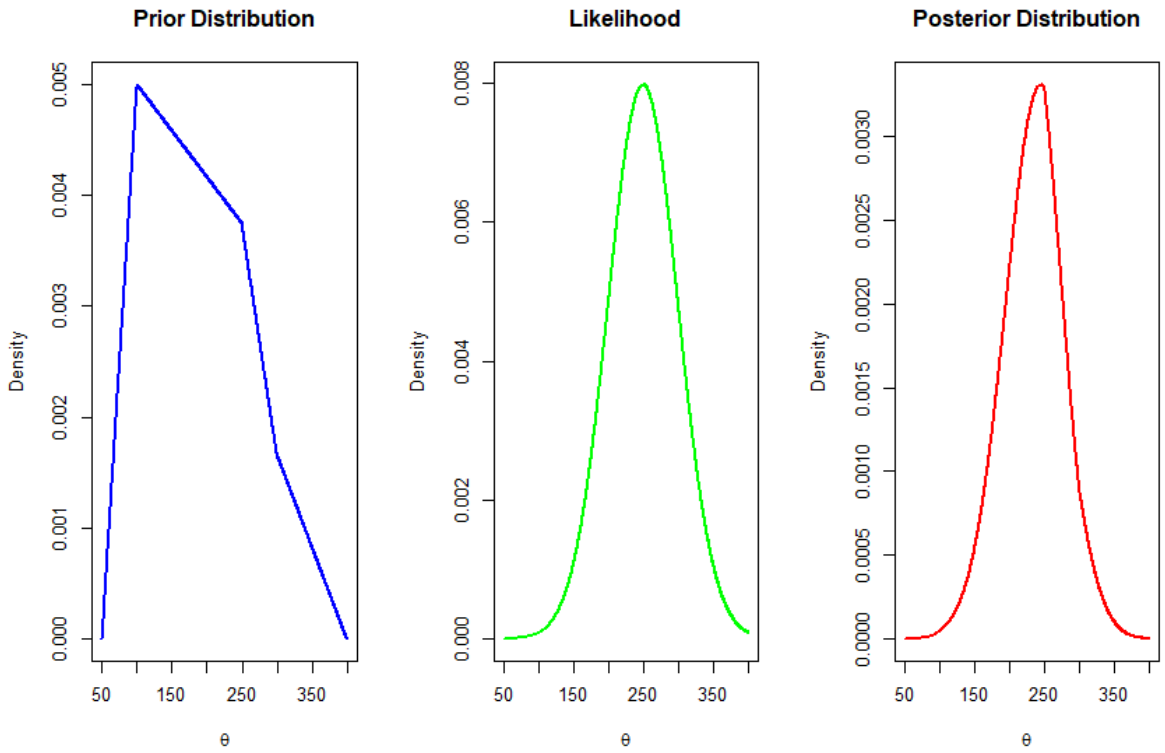
c) Custom Prior and Posterior (with n=1)

The prior is defined piecewise over [50, 400] and is non-standard. The posterior is proportional to prior × likelihood.

Below is an R program to implement this logic:

```
# Define grid for theta
theta <- seq(50, 400, length.out = 1000)

# Custom piecewise prior
prior <- ifelse(theta >= 50 & theta <= 100, (1 / 10000) * theta - (1 / 200),
      ifelse(theta > 100 & theta <= 250, (-1 / 120000) * theta + (7 / 1200),
      ifelse(theta > 250 & theta <= 300, (-1 / 24000) * theta + (17 / 1200),
      ifelse(theta > 300 & theta <= 400, (-1 / 60000) * theta + (1 / 150), 0))))

# Likelihood for n = 1, X = 250, known sigma = 50
likelihood <- dnorm(theta, mean = 250, sd = 50)

# Posterior (unnormalized)
posterior <- prior * likelihood

# Normalize posterior
posterior <- posterior / sum(posterior)

# Posterior mean and SD
post_mean <- sum(theta * posterior)
post_sd <- sqrt(sum((theta - post_mean)^2 * posterior))

cat("Posterior Mean:", post_mean, "\n")
cat("Posterior SD:", post_sd, "\n")

# Plotting
par(mfrow = c(1, 3))
plot(theta, prior, type = 'l', col = 'blue', lwd = 2, main = 'Prior Distribution', xlab =
expression(theta), ylab = 'Density')
plot(theta, likelihood, type = 'l', col = 'green', lwd = 2, main = 'Likelihood', xlab =
expression(theta), ylab = 'Density')
```

plot(theta, posterior, type = 'l', col = 'red', lwd = 2, main = 'Posterior Distribution', xlab = expression(theta), ylab = 'Density')



Q5.1: K-Means Clustering

a) Scatterplot and Visual Estimation of Clusters

Load the dataset using the following R code:

zz <- read.table('lettersdata.txt')
zz <- as.matrix(zz)

Then use a scatterplot to visualize the dataset:

plot(zz, main = 'Scatterplot of Letters Data', xlab = 'X1', ylab = 'X2')

From the scatterplot, estimate the number of distinct groups visually. For example, if three clear groups are visible, we might choose k = 3.

b) K-Means Clustering with Estimated k Value

Assume k = 3 (based on visual inspection).

Perform k-means clustering:

set.seed(123)
kmeans_result <- kmeans(zz, centers = 3)
plot(zz, col = kmeans_result$cluster, pch = 19, main = "K-Means Clustering (k=3)")

points(kmeans_result$centers, col = 1:3, pch = 4, cex = 2, lwd = 3)

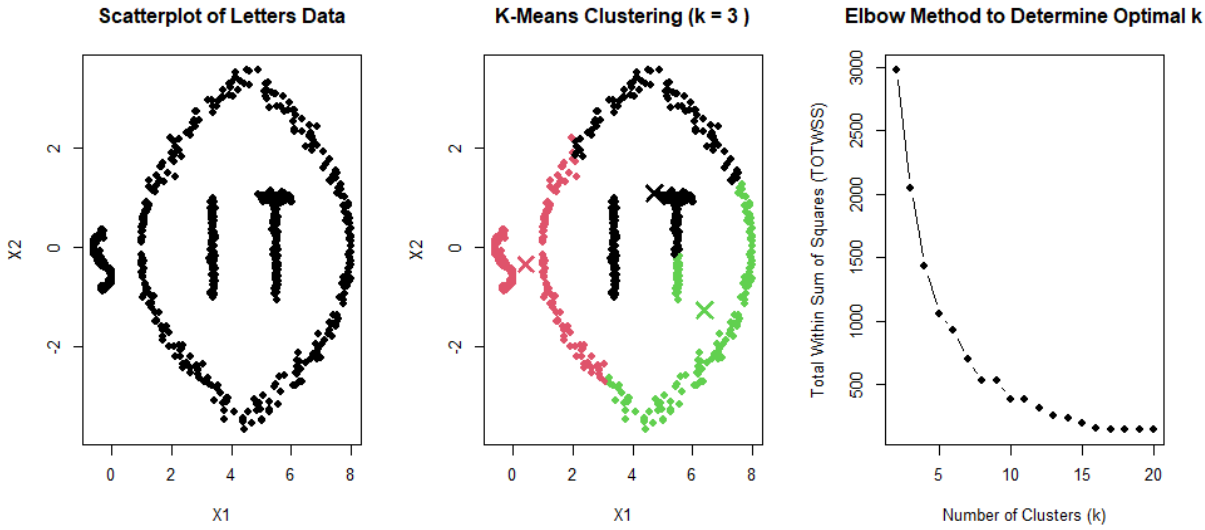
Comment: The plot shows data points grouped into 3 clusters. Each point is colored based on its cluster label. Cluster centers are also marked. If the clusters are compact and well-separated, this indicates good clustering.

c) Elbow Method to Choose Optimal k (2 to 20)

We vary k from 2 to 20 and compute total within-cluster sum of squares (TOTWSS):

```
totwss <- numeric(19)
for (k in 2:20) {
 set.seed(123)
 totwss[k - 1] <- kmeans(zz, centers = k)$tot.withinss
}
plot(2:20, totwss, type = 'b', pch = 19, frame = FALSE,
    xlab = 'Number of clusters (k)', ylab = 'Total Within Sum of Squares',
    main = 'Elbow Method for Optimal k')
```

Explanation: The 'elbow point' in the plot where the decrease in TOTWSS becomes marginal. That point indicates the optimal number of clusters.



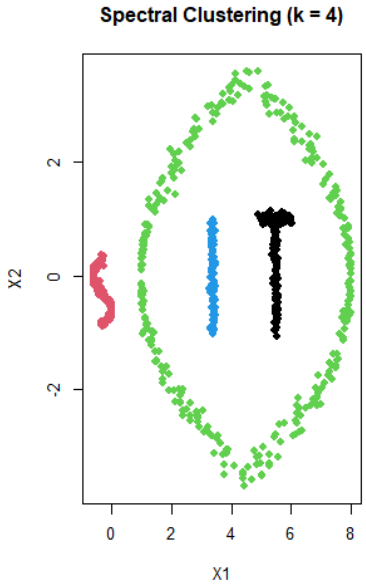
Q5.2: Spectral Clustering

Spectral Clustering on lettersdata.txt (k = 4)

The dataset 'lettersdata.txt' was used to perform spectral clustering using 4 clusters. The `specc()` function from the `kernlab` package was used for this task.

R Code Used:

```
# Load required package
install.packages("kernlab")
library(kernlab)

# Load the dataset
zz <- read.table("lettersdata.txt")
zz <- as.matrix(zz)

# Spectral Clustering with 4 clusters
sc <- specc(zz, centers = 4)

# Plot spectral clustering results
plot(zz, col = sc, pch = 19,
   main = "Spectral Clustering (k = 4)",
   xlab = "X1", ylab = "X2")
```



Comparison with K-Means Clustering

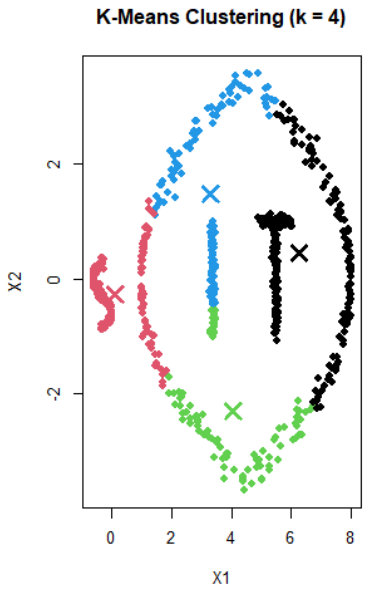
Both K-Means and Spectral Clustering were performed using 4 clusters. While K-Means assumes spherical, linearly separable clusters and minimizes within-cluster variance, Spectral Clustering is capable of capturing more complex cluster shapes by leveraging eigenvectors of the similarity matrix.

In the scatter plots:

- K-Means Clustering typically creates convex-shaped clusters. If the data structure is non-linear, it may split natural clusters or merge parts incorrectly.

- Spectral Clustering tends to provide better clustering if there are non-convex, curved, or complex boundaries between groups.

Hence, Spectral Clustering may result in a more intuitive segmentation when compared to K-Means, especially if the natural structure of the data is non-linear.



Complete R code of all questions

```
getwd()
library(readr)
read.csv("AgincourtDataCSV.csv")
the.fulldata <- as.matrix(read.csv("AgincourtDataCSV.csv", header = TRUE, sep = ","))

# Sample 10,000 rows
set.seed(123)
my.data <- the.fulldata[sample(1:24999, 10000), 1:5]
write.table(my.data, "Beerappa-223724697-AgincourtMyData.txt")

# Q1.1 - Analyzing the Water Temperature variable

# First, I'm plotting a histogram to see the frequency distribution of water temperatures.
# This helps me understand how the values are spread — whether it's normal, skewed, etc.
hist(my.data[, 5],
    main = "Histogram of Water Temperature",
    xlab = "Water Temperature (°C)",
    col = "skyblue",
```

```
    border = "black")

# Now I'm plotting a boxplot to visualize the spread and identify any outliers in the data.
# The boxplot shows the minimum, Q1, median, Q3, and maximum visually.
boxplot(my.data[, 5],
      main = "Boxplot of Water Temperature",
      ylab = "Water Temperature (°C)",
      col = "lightgreen")

# This command gives me the five-number summary of the Water Temperature variable.
# It includes: Min, Q1, Median, Q3, and Max — useful for summarizing distribution.
summary(my.data[, 5])

# This is another way to extract the exact five-number summary using base R.
fivenum(my.data[, 5])

#Q1.2 Answer – Explanation
#To summarize the center of the Water temperature variable, I would choose the median
#instead of the mean. This is because the median is less affected by outliers or any
#skewed values in the data. Based on the boxplot, the distribution appears slightly
#skewed (or possibly has mild outliers), so the median gives a better sense of the "typical"
#temperature value.

#To summarize the spread, I would choose the interquartile range (IQR).

#The IQR measures the spread of the middle 50% of the data (between Q1 and Q3)
#and is also resistant to outliers, unlike the standard deviation. Since the boxplot
#shows how the water temperature is concentrated within that middle range, the IQR gives
#a more reliable idea of how much variability there is among the typical values.

# Q1.3 - Scatterplot and Linear Regression
```

```
# Creating a scatterplot of Air Temperature (x) vs Water Temperature (y)
# This helps me see if there's a linear relationship between the two variables
plot(my.data[, 4], my.data[, 5],
    main = "Air Temperature vs Water Temperature",
    xlab = "Air Temperature (°C)",
    ylab = "Water Temperature (°C)",
    pch = 19, col = "blue")

# Fitting a simple linear regression model
# This line will fit a model: Water Temp = a + b * Air Temp
model <- lm(my.data[, 5] ~ my.data[, 4])

# Adding the regression line to the scatterplot
abline(model, col = "red", lwd = 2)

# Displaying the model coefficients
summary(model)

# Calculating correlation coefficient
correlation <- cor(my.data[, 4], my.data[, 5])
correlation  # Just to see the value

# Calculating coefficient of determination (R-squared)
r_squared <- correlation^2
r_squared
summary(model)$coefficients



# Q1.4(a) - Creating AT, AP, WS variables based on conditions
```

```
# Creating the 'AT' variable for Air Temperature
# High if > 28, Moderate if between 26 and 28, Low if < 26
AT <- ifelse(my.data[, 4] > 28, "High",
        ifelse(my.data[, 4] >= 26 & my.data[, 4] <= 28, "Moderate", "Low"))

# Create the 'AP' variable for Air Pressure
# High if > 1009, else Low
AP <- ifelse(my.data[, 3] > 1009, "High", "Low")

# Create the 'WS' variable for Wind Speed
# High if > 25, else Low
WS <- ifelse(my.data[, 2] > 25, "High", "Low")

# Combine them into a data frame
cross_data <- data.frame(AT, AP, WS)

# Generate the cross table
table_AT_AP_WS <- table(cross_data$AT, cross_data$AP, cross_data$WS)
table_AT_AP_WS

#Q1.4(b) – Answering the Probability Questions
#i) P(WS = High)
# Count of High WS
high_ws_count <- sum(WS == "High")
total <- length(WS)
prob_ws_high <- high_ws_count / total
prob_ws_high

#ii) P(AP = Low | AT = Moderate)
# Total rows where AT is Moderate
moderate_at <- cross_data$AT == "Moderate"
```

```
# Out of those, count where AP is Low
moderate_at_ap_low <- sum(cross_data$AP[moderate_at] == "Low")
# Total Moderate AT
moderate_total <- sum(moderate_at)
prob_ap_low_given_at_moderate <- moderate_at_ap_low / moderate_total
prob_ap_low_given_at_moderate

#iii) P(WS = High | AT = Low & AP = High)
# Rows where AT = Low and AP = High
at_low_ap_high <- cross_data$AT == "Low" & cross_data$AP == "High"
# Out of those, how many have WS = High?
ws_high_given_condition <- sum(cross_data$WS[at_low_ap_high] == "High")
# Total with AT = Low and AP = High
total_condition <- sum(at_low_ap_high)
prob_ws_high_given_conditions <- ws_high_given_condition / total_condition
prob_ws_high_given_conditions

# iv) Are Low AP and High AT mutually exclusive?
#Answer:No, Low AP and High AT are not mutually exclusive if there is at least
#one row where both conditions occur together.
sum(AP == "Low" & AT == "High")

#v) Are Low AT and Low AP independent?
#Two events A and B are independent if:
#P(A and B) = P(A) * P(B)
p_low_at <- sum(AT == "Low") / length(AT)
p_low_ap <- sum(AP == "Low") / length(AP)
p_both <- sum(AT == "Low" & AP == "Low") / length(AT)

# Compare
p_low_at * p_low_ap
```

```
p_both

#2) iv) Are Low AP and High AT mutually exclusive?
sum(AP == "Low" & AT == "High")



#3.2c
# Inter-arrival time data
data <- c(17, 5, 10, 20, 18, 6, 15, 8)

# Given prior hyperparameters
a <- 4
b <- 1

# Frequentist data stats
N <- length(data)
K <- mean(data)

# Posterior hyperparameters
a_post <- a + N
b_post <- b + N * K

# Lambda range for plotting
lambda <- seq(0.001, 0.5, length.out = 1000)

# Prior: Gamma(a, b)
prior <- dgamma(lambda, shape = a, rate = b)

# Likelihood (unnormalized, for visualization only)
likelihood <- lambda^N * exp(-N * lambda * K)
```

```
likelihood <- likelihood / max(likelihood)  # Scale for plotting


# Posterior: Gamma(a_post, b_post)
posterior <- dgamma(lambda, shape = a_post, rate = b_post)


# Plotting
plot(lambda, prior, type = "l", col = "blue", lwd = 2, ylim = c(0, max(c(prior, posterior))),
    xlab = expression(lambda), ylab = "Density", main = "Prior, Likelihood & Posterior
Distributions")
lines(lambda, likelihood, col = "green", lwd = 2)
lines(lambda, posterior, col = "red", lwd = 2)


legend("topright", legend = c("Prior", "Likelihood (scaled)", "Posterior"),
      col = c("blue", "green", "red"), lwd = 2)



#Q.4 C Define grid for theta values
theta <- seq(50, 400, length.out = 1000)


# Define the custom piecewise prior distribution
prior <- ifelse(theta >= 50 & theta <= 100, (1 / 10000) * theta - (1 / 200),
            ifelse(theta > 100 & theta <= 250, (-1 / 120000) * theta + (7 / 1200),
                ifelse(theta > 250 & theta <= 300, (-1 / 24000) * theta + (17 / 1200),
                    ifelse(theta > 300 & theta <= 400, (-1 / 60000) * theta + (1 / 150), 0))))


# Likelihood (Normal distribution) for n = 1
# Observation: sample mean = 250, known standard deviation = 50
likelihood <- dnorm(theta, mean = 250, sd = 50)


# Posterior is proportional to prior × likelihood
posterior_unnormalized <- prior * likelihood
```

```
# Normalize posterior to get a probability distribution
posterior <- posterior_unnormalized / sum(posterior_unnormalized)

# Calculate posterior mean and standard deviation
posterior_mean <- sum(theta * posterior)
posterior_sd <- sqrt(sum((theta - posterior_mean)^2 * posterior))

# Print posterior mean and sd
cat("Posterior Mean:", posterior_mean, "\n")
cat("Posterior SD:", posterior_sd, "\n")

# Plotting
par(mfrow = c(1, 3))  # Layout for 3 plots side-by-side

# Plot prior
plot(theta, prior, type = "l", lwd = 2, col = "blue",
    main = "Prior Distribution", xlab = expression(theta), ylab = "Density")

# Plot likelihood
plot(theta, likelihood, type = "l", lwd = 2, col = "green",
    main = "Likelihood", xlab = expression(theta), ylab = "Density")

# Plot posterior
plot(theta, posterior, type = "l", lwd = 2, col = "red",
    main = "Posterior Distribution", xlab = expression(theta), ylab = "Density")



#Q.5Load the dataset
zz <- read.table("lettersdata.txt")
zz <- as.matrix(zz)
```

```
# Part a: Scatterplot for visual inspection ----
plot(zz, main = "Scatterplot of Letters Data", xlab = "X1", ylab = "X2", pch = 19)

# Based on the plot, visually we can estimate number of clusters (e.g., k = 3)

# ---- Part b: K-Means Clustering with estimated k ----
set.seed(123)  # for reproducibility
k <- 3  # replace with your visually estimated value
kmeans_result <- kmeans(zz, centers = k)

# Plot K-means clusters
plot(zz, col = kmeans_result$cluster, pch = 19,
    main = paste("K-Means Clustering (k =", k, ")"),
    xlab = "X1", ylab = "X2")
points(kmeans_result$centers, col = 1:k, pch = 4, cex = 2, lwd = 3)

# ---- Part c: Elbow Method (TOTWSS vs. k from 2 to 20) ----
totwss <- numeric(19)
for (k in 2:20) {
  set.seed(123)
  km <- kmeans(zz, centers = k)
  totwss[k - 1] <- km$tot.withinss
}

# Plot TOTWSS vs. k
plot(2:20, totwss, type = "b", pch = 19,
    xlab = "Number of Clusters (k)",
    ylab = "Total Within Sum of Squares (TOTWSS)",
    main = "Elbow Method to Determine Optimal k")
```

```
#Q5.2# Load required package
install.packages("kernlab")  # Run only once if not installed
library(kernlab)

# Load the data (if not already loaded)
zz <- read.table("lettersdata.txt")
zz <- as.matrix(zz)

# ---- Spectral Clustering with 4 clusters ----
sc <- specc(zz, centers = 4)  # Spectral clustering

# Plot spectral clustering results
plot(zz, col = sc, pch = 19,
    main = "Spectral Clustering (k = 4)",
    xlab = "X1", ylab = "X2")

# ---- K-Means Clustering for comparison (k = 4) ----
set.seed(123)
km4 <- kmeans(zz, centers = 4)

# Plot K-means clustering results for comparison
plot(zz, col = km4$cluster, pch = 19,
    main = "K-Means Clustering (k = 4)",
    xlab = "X1", ylab = "X2")
points(km4$centers, col = 1:4, pch = 4, cex = 2, lwd = 3)
```