Day 1

List of Programs:

1The intervals and corresponding frequencies are as follows. age frequency

1-5. 200

5-15 450

15-20 300

20-50 1500

50-80 700

80-110 44

Compute an approximate median value for the data

Program

intervals <- c(1, 5, 15, 20, 50, 80, 110)

frequencies <- c(200, 450, 300, 1500, 700, 44)

cumulative\_freq <- cumsum(frequencies)

median\_index <- which(cumulative\_freq >= sum(frequencies) / 2)[1]

lower\_bound <- intervals[median\_index]

upper\_bound <- intervals[median\_index + 1]

midpoint <- lower\_bound + (upper\_bound - lower\_bound) \* (0.5 - cumulative\_freq[median\_index - 1]) / frequencies[median\_index]

cat("Approximate median value:", midpoint, "\n")

2.Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) What is the mean of the data? What is the median?

(b) What is the mode of the data? Comment on the data’s modality (i.e., bimodal, trimodal, etc.).

(c) What is the midrange of the data?

(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

Program

# Given data

age\_data <- c(13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70)

# Mean

mean\_age <- mean(age\_data)

cat("Mean:", mean\_age, "\n")

# Median

median\_age <- median(age\_data)

cat("Median:", median\_age, "\n")

# Mode

mode\_age <- as.numeric(names(table(age\_data)[table(age\_data) == max(table(age\_data))]))

cat("Mode:", mode\_age, "\n")

# Modality

modality <- length(table(age\_data))

if (modality == 1) {

cat("Data is unimodal (one mode)\n")

} else {

cat("Data is", modality, "modal\n")

}

# Midrange

midrange\_age <- (max(age\_data) + min(age\_data)) / 2

cat("Midrange:", midrange\_age, "\n")

# Quartiles

q1 <- quantile(age\_data, 0.25)

q3 <- quantile(age\_data, 0.75)

cat("First Quartile (Q1):", q1, "\n")

cat("Third Quartile (Q3):", q3, "\n")

3.Data Preprocessing :Reduction and Transformation

Use the two methods below to normalize the following group of data: 200, 300, 400, 600, 1000 (a) min-max normalization by setting min = 0 and max = 1 (b) z-score normalization

Program

data <- c(200, 300, 400, 600, 1000)

min\_max\_normalize <- function(x) {

(x - min(x)) / (max(x) - min(x))

}

min\_max\_normalized <- min\_max\_normalize(data)

cat("Min-Max Normalization:")

print(min\_max\_normalized)

z\_score\_normalize <- function(x) {

(x - mean(x)) / sd(x)

}

z\_score\_normalized <- z\_score\_normalize(data)

cat("Z-Score Normalization:")

print(z\_score\_normalized)

4.Data:11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,

72,73,75

a) Smoothing by bin mean

b) Smoothing by bin median

c) Smoothing by bin boundaries

program

# Given data

data <- c(11, 13, 13, 15, 15, 16, 19, 20, 20, 20, 21, 21, 22, 23, 24, 30, 40, 45, 45, 45, 71, 72, 73, 75)

# Function to perform bin smoothing using mean

bin\_mean\_smooth <- function(data, bin\_size) {

smoothed\_data <- numeric(length = length(data))

for (i in seq\_len(length(data))) {

lower <- max(1, i - bin\_size + 1)

upper <- min(length(data), i + bin\_size - 1)

smoothed\_data[i] <- mean(data[lower:upper])

}

return(smoothed\_data)

}

# Function to perform bin smoothing using median

bin\_median\_smooth <- function(data, bin\_size) {

smoothed\_data <- numeric(length = length(data))

for (i in seq\_len(length(data))) {

lower <- max(1, i - bin\_size + 1)

upper <- min(length(data), i + bin\_size - 1)

smoothed\_data[i] <- median(data[lower:upper])

}

return(smoothed\_data)

}

# Function to perform bin smoothing using boundaries

bin\_boundaries\_smooth <- function(data, bin\_size) {

smoothed\_data <- numeric(length = length(data))

for (i in seq\_len(length(data))) {

lower <- max(1, i - bin\_size + 1)

upper <- min(length(data), i + bin\_size - 1)

smoothed\_data[i] <- (data[lower] + data[upper]) / 2

}

return(smoothed\_data)

}

# Bin size for smoothing

bin\_size <- 3

# Perform smoothing using different methods

smoothed\_mean <- bin\_mean\_smooth(data, bin\_size)

smoothed\_median <- bin\_median\_smooth(data, bin\_size)

smoothed\_boundaries <- bin\_boundaries\_smooth(data, bin\_size)

# Print the smoothed data

cat("Smoothing by bin mean:\n")

print(smoothed\_mean)

cat("Smoothing by bin median:\n")

print(smoothed\_median)

cat("Smoothing by bin boundaries:\n")

print(smoothed\_boundaries)

5. Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:



(a) Calculate the mean, median, and standard deviation of age and %fat. (b) Draw the boxplots for age and %fat.  
(c) Draw a scatter plot and a q-q plot based on these two variables.

6.Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

(i) Use min-max normalization to transform the value 35 for age onto the range [0.0, 1.0].  
(ii) Use z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years.  
(iii) Use normalization by decimal scaling to transform the value 35 for age. Perform the above functions using R – tool

Program

# Given value

age\_value <- 35

# (i) Min-Max Normalization

min\_age <- 0

max\_age <- 1

min\_max\_normalized <- (age\_value - min\_age) / (max\_age - min\_age)

cat("Min-Max Normalization:", min\_max\_normalized, "\n")

# (ii) Z-Score Normalization

mean\_age <- 0 # Assuming mean age is 0 for simplicity

std\_dev\_age <- 12.94

z\_score\_normalized <- (age\_value - mean\_age) / std\_dev\_age

cat("Z-Score Normalization:", z\_score\_normalized, "\n")

# (iii) Normalization by Decimal Scaling

scale\_factor <- 10

decimal\_scaled <- age\_value / scale\_factor

cat("Normalization by Decimal Scaling:", decimal\_scaled, "\n")

7.The following values are the number of pencils available in the different boxes. Create a vector and find out the mean, median and mode values of set of pencils in the given data.

Box1 Box2 Box3 Box4 Box5 Box6 Box7 Box8 Box9 Box 10

9 25 23 12 11 6 7 8 9 10

Program

# Create a vector with pencil values

pencil\_values <- c(9, 25, 23, 12, 11, 6, 7, 8, 9, 10)

# Calculate mean

mean\_value <- mean(pencil\_values)

cat("Mean:", mean\_value, "\n")

# Calculate median

median\_value <- median(pencil\_values)

cat("Median:", median\_value, "\n")

# Calculate mode

mode\_value <- as.numeric(names(sort(table(pencil\_values), decreasing = TRUE)[1]))

cat("Mode:", mode\_value, "\n")

8. the following table would be plotted as (x,y) points, with the first column being the x values as number of mobile phones sold and the second column being the y values as money. To use the scatter plot for how many mobile phones sold.

x :4 1 5 7 10 2 50 25 90 36

y :12 5 13 19 31 7 153 72 275 110

program

# Given data

x\_values <- c(4, 1, 5, 7, 10, 2, 50, 25, 90, 36)

y\_values <- c(12, 5, 13, 19, 31, 7, 153, 72, 275, 110)

# Load the necessary library for plotting

library(ggplot2)

# Create a data frame from the data

data\_df <- data.frame(x = x\_values, y = y\_values)

# Create a scatter plot

ggplot(data\_df, aes(x, y)) +

geom\_point(color = "blue") +

labs(title = "Scatter Plot of Mobile Phones Sold",

x = "Number of Mobile Phones Sold",

y = "Money") +

theme\_minimal()

9. Implement of the R script using marks scored by a student in his model exam has been sorted as follows: 55, 60, 71, 63, 55, 65, 50, 55,58,59,61,63,65,67,71,72,75. Partition them into three bins by each of the following methods. Plot the data points using histogram.

(a) equal-frequency (equi-depth) partitioning (b) equal-width partitioning

PROGRAM

# Given data

marks <- c(55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75)

# Number of bins for partitioning

num\_bins\_ef <- 3

num\_bins\_ew <- 3

# (a) Equal-Frequency (Equi-Depth) Partitioning

marks\_ef\_bins <- cut(marks, breaks = num\_bins\_ef, labels = FALSE)

# (b) Equal-Width Partitioning

marks\_ew\_bins <- cut(marks, breaks = num\_bins\_ew, labels = FALSE)

# Load the necessary library for plotting

library(ggplot2)

# Create a data frame for the histograms

data\_ef <- data.frame(Marks = marks, Bins = factor(marks\_ef\_bins))

data\_ew <- data.frame(Marks = marks, Bins = factor(marks\_ew\_bins))

# Plot histograms for both methods

histogram\_ef <- ggplot(data\_ef, aes(x = Marks, fill = Bins)) +

geom\_histogram(binwidth = 10, position = "identity") +

labs(title = "Histogram (Equal-Frequency Partitioning)",

x = "Marks",

y = "Frequency") +

theme\_minimal()

histogram\_ew <- ggplot(data\_ew, aes(x = Marks, fill = Bins)) +

geom\_histogram(binwidth = 10, position = "identity") +

labs(title = "Histogram (Equal-Width Partitioning)",

x = "Marks",

y = "Frequency") +

theme\_minimal()

# Display histograms

print(histogram\_ef)

print(histogram\_ew)

10. Suppose that the speed car is mentioned in different driving style.

Regular 78.3 81.8 82 74.2 83.4 84.5 82.9 77.5 80.9 70.6 Speed

Calculate the Inter quantile and standard deviation of the given data.

Program

# Given data

speed\_data <- c(78.3, 81.8, 82, 74.2, 83.4, 84.5, 82.9, 77.5, 80.9, 70.6)

# Calculate Interquartile Range (IQR)

q1 <- quantile(speed\_data, 0.25)

q3 <- quantile(speed\_data, 0.75)

iqr <- q3 - q1

cat("Interquartile Range (IQR):", iqr, "\n")

# Calculate Standard Deviation

std\_dev <- sd(speed\_data)

cat("Standard Deviation:", std\_dev, "\n")

11.Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

Program

# Given age data

age\_data <- c(13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70)

# Calculate first quartile (Q1)

q1 <- quantile(age\_data, 0.25, type = 1)

# Calculate third quartile (Q3)

q3 <- quantile(age\_data, 0.75, type = 1)

cat("First Quartile (Q1):", q1, "\n")

cat("Third Quartile (Q3):", q3, "\n")