Day 1

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

**# Create a dataframe with the given data**

**data <- data.frame(**

**age = c("1-5", "5-15", "15-20", "20-50", "50-80", "80-110"),**

**frequency = c(200, 450, 300, 1500, 700, 44)**

**)**

**# Calculate cumulative frequencies**

**data$cumulative\_frequency <- cumsum(data$frequency)**

**# Find the interval containing the median**

**total\_frequency <- sum(data$frequency)**

**median\_cf <- total\_frequency / 2**

**median\_interval <- data$age[which(data$cumulative\_frequency >= median\_cf)[1]]**

**# Calculate the median value within the median interval**

**median\_interval\_values <- as.numeric(strsplit(median\_interval, "-")[[1]])**

**median\_lower\_limit <- median\_interval\_values[1]**

**median\_upper\_limit <- median\_interval\_values[2]**

**median\_interval\_width <- median\_upper\_limit - median\_lower\_limit**

**median\_cumulative\_frequency\_before <- data$cumulative\_frequency[which(data$age == median\_interval) - 1]**

**median\_frequency\_within\_interval <- data$frequency[which(data$age == median\_interval)]**

**median <- median\_lower\_limit + ((median\_cf - median\_cumulative\_frequency\_before) / median\_frequency\_within\_interval) \* median\_interval\_width**

**print(paste("Approximate median value:", median))**

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?

**A.**

**# Age values**

ages <- 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 mean

mean\_age <- mean(ages)

# Calculate median

median\_age <- median(ages)

print(paste("Mean:", mean\_age))

print(paste("Median:", median\_age))

**B.# Calculate mode**

mode\_age <- as.numeric(names(sort(table(ages), decreasing = TRUE)[1]))

# Determine modality

num\_modes <- length(table(ages))

modality <- ifelse(num\_modes == 1, "Unimodal",

ifelse(num\_modes == 2, "Bimodal",

ifelse(num\_modes == 3, "Trimodal",

"Multimodal")))

print(paste("Mode:", mode\_age))

print(paste("Modality:", modality))

**C.**

**# Calculate midrange**

**midrange\_age <- (max(ages) + min(ages)) / 2**

**print(paste("Midrange:", midrange\_age))**

**D.# Calculate quartiles**

**q1 <- quantile(ages, 0.25)**

**q3 <- quantile(ages, 0.75)**

**print(paste("First Quartile (Q1):", q1))**

**print(paste("Third Quartile (Q3):", q3))**

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

**A)# Given data**

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

**# Min-max normalization**

**min\_value <- min(data)**

**max\_value <- max(data)**

**normalized\_data\_minmax <- (data - min\_value) / (max\_value - min\_value)**

**print("Min-Max Normalized Data:")**

**print(normalized\_data\_minmax)**

**B)# Z-score normalization**

**mean\_value <- mean(data)**

**std\_deviation <- sd(data)**

**normalized\_data\_zscore <- (data - mean\_value) / std\_deviation**

**print("Z-Score Normalized Data:")**

**print(normalized\_data\_zscore)**

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

**A)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)**

**bin\_size <- 5**

**# Create bins**

**bins <- seq(min(data), max(data), by = bin\_size)**

**# Calculate bin means**

**bin\_means <- tapply(data, cut(data, breaks = bins), mean)**

**print("Smoothing by Bin Mean:")**

**print(bin\_means)**

**B.# Calculate bin medians**

**bin\_medians <- tapply(data, cut(data, breaks = bins), median)**

**print("Smoothing by Bin Median:")**

**print(bin\_medians)**

**C.# Calculate bin boundaries**

**bin\_boundaries <- seq(min(data), max(data) + bin\_size, by = bin\_size)**

**print("Smoothing by Bin Boundaries:")**

**print(bin\_boundaries)**

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



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

**# Age and %fat data (replace with actual data)**

**age <- c(...)**

**percent\_fat <- c(...)**

**# Calculate mean**

**mean\_age <- mean(age)**

**mean\_percent\_fat <- mean(percent\_fat)**

**# Calculate median**

**median\_age <- median(age)**

**median\_percent\_fat <- median(percent\_fat)**

**# Calculate standard deviation**

**sd\_age <- sd(age)**

**sd\_percent\_fat <- sd(percent\_fat)**

**print("Mean Age:", mean\_age)**

**print("Median Age:", median\_age)**

**print("Standard Deviation Age:", sd\_age)**

**print("Mean %Fat:", mean\_percent\_fat)**

**print("Median %Fat:", median\_percent\_fat)**

**print("Standard Deviation %Fat:", sd\_percent\_fat)**

**# Create boxplot for age**

**boxplot(age, main="Boxplot of Age")**

**# Create boxplot for %fat**

**boxplot(percent\_fat, main="Boxplot of %Fat")**

**# Create scatter plot**

**plot(age, percent\_fat, main="Scatter Plot", xlab="Age", ylab="%Fat")**

**# Create Q-Q plot for age**

**qqnorm(age)**

**qqline(age, col = 2)**

**# Create Q-Q plot for %fat**

**qqnorm(percent\_fat)**

**qqline(percent\_fat, col = 2)**

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

**# Given value**

**value <- 35**

**# Min-Max Normalization**

**min\_value <- 0**

**max\_value <- 1**

**minmax\_normalized <- (value - min\_value) / (max\_value - min\_value)**

**print(paste("Min-Max Normalized value:", minmax\_normalized))**

**# Z-Score Normalization**

**mean\_age <- 0 # Assume mean of age is 0 for simplicity**

**std\_deviation\_age <- 12.94**

**zscore\_normalized <- (value - mean\_age) / std\_deviation\_age**

**print(paste("Z-Score Normalized value:", zscore\_normalized))**

**# Normalization by Decimal Scaling**

**power <- floor(log10(max(abs(value)))) + 1**

**decimal\_scaled <- value / (10 ^ power)**

**print(paste("Normalization by Decimal Scaling:", decimal\_scaled))**

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

**# Create a vector with the given data**

**pencils <- c(9, 25, 23, 12, 11, 6, 7, 8, 9, 10)**

**# Calculate mean**

**mean\_pencils <- mean(pencils)**

**# Calculate median**

**median\_pencils <- median(pencils)**

**# Calculate mode (using a custom function since base R doesn't have a built-in mode function)**

**get\_mode <- function(x) {**

**ux <- unique(x)**

**ux[which.max(tabulate(match(x, ux)))]**

**}**

**mode\_pencils <- get\_mode(pencils)**

**print(paste("Mean:", mean\_pencils))**

**print(paste("Median:", median\_pencils))**

**print(paste("Mode:", mode\_pencils))**

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

**# Given data**

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

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

**# Create a scatter plot**

**plot(x, y, main="Scatter Plot of Mobile Phones Sold",**

**xlab="Number of Mobile Phones Sold", ylab="Money")**

**# Add a grid to the plot (optional)**

**grid()**

**# Show the plot**

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

**# Given data**

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

**# Calculate number of bins**

**num\_bins <- 3**

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

**bin\_breaks\_eq\_freq <- quantile(marks, probs = seq(0, 1, length.out = num\_bins + 1))**

**marks\_binned\_eq\_freq <- cut(marks, breaks = bin\_breaks\_eq\_freq, include.lowest = TRUE)**

**# (b) Equal-Width Partitioning**

**bin\_width <- (max(marks) - min(marks)) / num\_bins**

**bin\_breaks\_eq\_width <- seq(min(marks), max(marks) + bin\_width, by = bin\_width)**

**marks\_binned\_eq\_width <- cut(marks, breaks = bin\_breaks\_eq\_width, include.lowest = TRUE)**

**# Plot histograms**

**par(mfrow = c(1, 2)) # Set up a 1x2 grid for two histograms**

**hist(marks, breaks = bin\_breaks\_eq\_freq, main = "Equal-Frequency Partitioning",**

**xlab = "Marks", ylab = "Frequency", col = "lightblue")**

**abline(v = bin\_breaks\_eq\_freq, col = "red", lwd = 2)**

**hist(marks, breaks = bin\_breaks\_eq\_width, main = "Equal-Width Partitioning",**

**xlab = "Marks", ylab = "Frequency", col = "lightgreen")**

**abline(v = bin\_breaks\_eq\_width, col = "red", lwd = 2)**

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.

**# 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**

**# Calculate Standard Deviation**

**sd\_speed <- sd(speed\_data)**

**print(paste("Interquartile Range (IQR):", iqr))**

**print(paste("Standard Deviation:", sd\_speed))**

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?

**# 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)**

**# Calculate the positions for Q1 and Q3**

**total\_data\_points <- length(age\_data)**

**q1\_position <- (total\_data\_points + 1) / 4 # 25% of data**

**q3\_position <- 3 \* q1\_position # 75% of data**

**# Find the values at the approximate positions**

**q1 <- age\_data[ceiling(q1\_position)]**

**q3 <- age\_data[ceiling(q3\_position)]**

**print(paste("Approximate First Quartile (Q1):", q1))**

**print(paste("Approximate Third Quartile (Q3):", q3))**