**DAY -1**

|  |  |
| --- | --- |
| **Ex.1**  # Given data  age\_intervals <- c("1-5", "5-15", "15-20", "20-50", "50-80", "80-110")  frequencies <- c(200, 450, 300, 1500, 700, 44)  # Calculate cumulative frequencies  cumulative\_freq <- cumsum(frequencies)  # Find the class interval containing the median  N <- sum(frequencies)  median\_class\_index <- which(cumulative\_freq >= N / 2)[1]  median\_class <- age\_intervals[median\_class\_index]  # Extract lower and upper bounds of the median class  median\_class\_bounds <- as.numeric(strsplit(median\_class, "-")[[1]])  L <- median\_class\_bounds[1]  U <- median\_class\_bounds[2]  # Calculate the approximate median  F <- cumulative\_freq[median\_class\_index - 1]  f <- frequencies[median\_class\_index]  w <- U - L  median <- L + ((N / 2 - F) / f) \* w  median | **Ex. 2.**  # Given data  age <- 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)  # (a) Mean and Median  mean\_age <- mean(age)  median\_age <- median(age)  # (b) Mode  mode\_age <- names(table(age))[which.max(table(age))]  # (c) Midrange  midrange\_age <- (max(age) + min(age)) / 2  # (d) Quartiles  Q1 <- quantile(age, 0.25)  Q3 <- quantile(age, 0.75)  # Print the results  print(paste("Mean:", mean\_age))  print(paste("Median:", median\_age))  print(paste("Mode:", mode\_age))  print(paste("Midrange:", midrange\_age))  print(paste("Q1:", Q1))  print(paste("Q3:", Q3)) |
| **Ex. 3**  # Given data  data <- c(200, 300, 400, 600, 1000)  # Min-max normalization  min\_max <- function(x) {  (x - min(x)) / (max(x) - min(x))  }  min\_max\_normalized <- min\_max(data)  min\_max\_normalized  # Z-score normalization  z\_score <- function(x) {  (x - mean(x)) / sd(x)  }  z\_score\_normalized <- z\_score(data)  z\_score\_normalized | **Ex. 4.**  # 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)  # Number of bins  num\_bins <- 5  # Bin width  bin\_width <- ceiling((max(data) - min(data)) / num\_bins)  # Bin boundaries  bin\_boundaries <- seq(min(data), max(data), by = bin\_width)  # Bin indices  bin\_indices <- cut(data, breaks = bin\_boundaries, labels = FALSE)  # Smoothing by bin mean  bin\_means <- tapply(data, bin\_indices, mean)  smoothed\_mean <- sapply(bin\_indices, function(i) bin\_means[i])  # Smoothing by bin median  bin\_medians <- tapply(data, bin\_indices, median)  smoothed\_median <- sapply(bin\_indices, function(i) bin\_medians[i])  # Smoothing by bin boundaries  smoothed\_boundaries <- sapply(bin\_indices, function(i) bin\_boundaries[i])  # Print the results  print("Smoothing by bin mean:")  print(smoothed\_mean)  print("Smoothing by bin median:")  print(smoothed\_median)  print("Smoothing by bin boundaries:")  print(smoothed\_boundaries) |
| **Ex. 5.**  # Age and body fat data  age <- c(23, 23, 27, 27, 39, 41, 47, 49, 50, 52, 54, 54, 56, 57, 58, 58, 60, 61)  body\_fat <- c(9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2, 31.2, 34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7)  # (a) Calculate mean, median, and standard deviation  mean\_age <- mean(age)  median\_age <- median(age)  sd\_age <- sd(age)  mean\_fat <- mean(body\_fat)  median\_fat <- median(body\_fat)  sd\_fat <- sd(body\_fat)  # (b) Boxplots  par(mfrow = c(1, 2))  boxplot(age, main = "Age", ylab = "Age", col = "lightblue")  boxplot(body\_fat, main = "Body Fat %", ylab = "Body Fat %", col = "lightgreen")  # (c) Scatter plot and q-q plot  par(mfrow = c(1, 2))  plot(age, body\_fat, xlab = "Age", ylab = "Body Fat %", main = "Scatter Plot")  qqplot\_age <- qqplot(age, main = "Q-Q Plot: Age")  qqline(age)  qqplot\_fat <- qqplot(body\_fat, main = "Q-Q Plot: Body Fat %")  qqline(body\_fat)  # Print the results  cat("Age: Mean =", mean\_age, ", Median =", median\_age, ", SD =", sd\_age, "\n")  cat("Body Fat %: Mean =", mean\_fat, ", Median =", median\_fat, ", SD =", sd\_fat, "\n") | **Ex. 6.**  # Age and body fat data  age <- c(23, 23, 27, 27, 39, 41, 47, 49, 50, 52, 54, 54, 56, 57, 58, 58, 60, 61)  body\_fat <- c(9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2, 31.2, 34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7)  # (a) Calculate mean, median, and standard deviation  mean\_age <- mean(age)  median\_age <- median(age)  sd\_age <- sd(age)  mean\_fat <- mean(body\_fat)  median\_fat <- median(body\_fat)  sd\_fat <- sd(body\_fat)  # (b) Boxplots  par(mfrow = c(1, 2))  boxplot(age, main = "Age", ylab = "Age", col = "lightblue")  boxplot(body\_fat, main = "Body Fat %", ylab = "Body Fat %", col = "lightgreen")  # (c) Scatter plot and q-q plot  par(mfrow = c(1, 2))  plot(age, body\_fat, xlab = "Age", ylab = "Body Fat %", main = "Scatter Plot")  qqplot\_age <- qqplot(age, main = "Q-Q Plot: Age")  qqline(age)  qqplot\_fat <- qqplot(body\_fat, main = "Q-Q Plot: Body Fat %")  qqline(body\_fat)  # Print the results  cat("Age: Mean =", mean\_age, ", Median =", median\_age, ", SD =", sd\_age, "\n")  cat("Body Fat %: Mean =", mean\_fat, ", Median =", median\_fat, ", SD =", sd\_fat, "\n") |
| **Ex. 7.**  # Given value for age  age\_value <- 35  # (i) Min-max normalization  min\_age <- min(age)  max\_age <- max(age)  min\_max\_age <- (age\_value - min\_age) / (max\_age - min\_age)  # (ii) Z-score normalization  mean\_age <- mean(age)  sd\_age <- 12.94  z\_score\_age <- (age\_value - mean\_age) / sd\_age  # (iii) Normalization by decimal scaling  scale\_factor <- 10 ^ ceiling(log10(max(age)))  decimal\_scaled\_age <- age\_value / scale\_factor  # Print the results  cat("Min-max normalization for age:", min\_max\_age, "\n")  cat("Z-score normalization for age:", z\_score\_age, "\n")  cat("Normalization by decimal scaling for age:", decimal\_scaled\_age, "\n") | **Ex. 8.**  # Given data  pencils <- c(9, 25, 23, 12, 11, 6, 7, 8, 9, 10)  # Mean  mean\_pencils <- mean(pencils)  # Median  median\_pencils <- median(pencils)  # Mode (using the 'Mode' function)  Mode <- function(x) {  ux <- unique(x)  ux[which.max(tabulate(match(x, ux)))]  }  mode\_pencils <- Mode(pencils)  # Print the results  cat("Mean:", mean\_pencils, "\n")  cat("Median:", median\_pencils, "\n")  cat("Mode:", mode\_pencils, "\n") |
| **Ex. 9.**  # 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)  # Scatter plot  plot(x, y, xlab = "Number of Mobile Phones Sold", ylab = "Money", main = "Scatter Plot of Mobile Phones Sold vs Money") | **Ex. 10.**  # Given data  marks <- c(55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75)  # (a) Equal-frequency (equi-depth) partitioning  equal\_freq\_bins <- cut(marks, breaks = 3, labels = FALSE)  # (b) Equal-width partitioning  min\_mark <- min(marks)  max\_mark <- max(marks)  width <- (max\_mark - min\_mark) / 3  equal\_width\_bins <- cut(marks, breaks = seq(min\_mark, max\_mark + width, by = width), labels = FALSE)  # Plotting histogram  par(mfrow = c(1, 2))  hist(marks, breaks = 3, main = "Equal-frequency Partitioning", xlab = "Marks", ylab = "Frequency", col = "lightblue", border = "black")  abline(v = breaks, col = "red", lty = 2)  hist(marks, breaks = seq(min\_mark, max\_mark + width, by = width), main = "Equal-width Partitioning", xlab = "Marks", ylab = "Frequency", col = "lightgreen", border = "black")  abline(v = breaks, col = "red", lty = 2)  # Reset plot layout  par(mfrow = c(1, 1)) |
| **Ex.11.**  # Given data  speed <- c(78.3, 81.8, 82, 74.2, 83.4, 84.5, 82.9, 77.5, 80.9, 70.6)  # Interquartile range (IQR)  q1 <- quantile(speed, 0.25)  q3 <- quantile(speed, 0.75)  iqr <- q3 - q1  # Standard deviation  sd\_speed <- sd(speed)  # Print the results  cat("Interquartile range (IQR):", iqr, "\n")  cat("Standard deviation:", sd\_speed, "\n") | **Ex.12.**  # Given data  age <- 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)  # First quartile (Q1)  q1 <- quantile(age, 0.25)  # Third quartile (Q3)  q3 <- quantile(age, 0.75)  # Print the results  cat("First quartile (Q1):", q1, "\n")  cat("Third quartile (Q3):", q3, "\n") |