**Assignment 1 – 10**

1. **Create a vector of numbers from 1 to 10.** 
   1. **Calculate the mean(), sum(), min(), and max() of the vector.**
   2. **Generate a sequence from 5 to 50 with a step of 5 using seq().**
   3. **Concatenate the numbers and print them as a single string.**

**Code:**

# Create a vector from 1 to 10

v <- 1:10

# a. Calculate mean, sum, min, max

mean\_v <- mean(v)

sum\_v <- sum(v)

min\_v <- min(v)

max\_v <- max(v)

# Print values

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

cat("Sum:", sum\_v, "\n")

cat("Min:", min\_v, "\n")

cat("Max:", max\_v, "\n\n")

# b. Generate sequence from 5 to 50 with step 5

seq\_v <- seq(from = 5, to = 50, by = 5)

cat("Sequence from 5 to 50 with step 5:\n", seq\_v, "\n\n")

# c. Concatenate numbers and print as a single string

concat\_str <- paste(v, collapse = "")

cat("Concatenated String:\n", concat\_str)

Output:

Mean: 5.5

Sum: 55

Min: 1

Max: 10

Sequence from 5 to 50 with step 5:

5 10 15 20 25 30 35 40 45 50

Concatenated String:

12345678910

1. **Create a 3x3 matrix with numbers from 1 to 9 and perform the following task**
   1. **Matrix addition and multiplication with another matrix of the same dimensions.**
   2. **Row-wise and column-wise sums.**
   3. **Transpose of the matrix.**
   4. **Access and print the element in the second row, third column.**

**Code:**

# Create a 3x3 matrix with numbers from 1 to 9

mat1 <- matrix(1:9, nrow = 3, byrow = TRUE)

cat("Matrix 1:\n")

print(mat1)

# Create another 3x3 matrix with numbers from 9 to 1

mat2 <- matrix(9:1, nrow = 3, byrow = TRUE)

cat("\nMatrix 2:\n")

print(mat2)

# a. Matrix Addition and Multiplication

add\_result <- mat1 + mat2

mult\_result <- mat1 \* mat2 # Element-wise multiplication

cat("\nMatrix Addition:\n")

print(add\_result)

cat("\nMatrix Multiplication (element-wise):\n")

print(mult\_result)

# b. Row-wise and Column-wise sums

row\_sums <- rowSums(mat1)

col\_sums <- colSums(mat1)

cat("\nRow-wise Sums of Matrix 1:\n")

print(row\_sums)

cat("\nColumn-wise Sums of Matrix 1:\n")

print(col\_sums)

# c. Transpose of the matrix

transpose <- t(mat1)

cat("\nTranspose of Matrix 1:\n")

print(transpose)

# d. Access and print element in 2nd row, 3rd column

element <- mat1[2, 3]

cat("\nElement at 2nd row, 3rd column of Matrix 1:\n", element, "\n")

**Output:**

Matrix 1:

[,1] [,2] [,3]

[1,] 1 2 3

[2,] 4 5 6

[3,] 7 8 9

Matrix 2:

[,1] [,2] [,3]

[1,] 9 8 7

[2,] 6 5 4

[3,] 3 2 1

Matrix Addition:

[,1] [,2] [,3]

[1,] 10 10 10

[2,] 10 10 10

[3,] 10 10 10

Matrix Multiplication (element-wise):

[,1] [,2] [,3]

[1,] 9 16 21

[2,] 24 25 24

[3,] 21 16 9

Row-wise Sums of Matrix 1:

[1] 6 15 24

Column-wise Sums of Matrix 1:

[1] 12 15 18

Transpose of Matrix 1:

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

Element at 2nd row, 3rd column of Matrix 1:

[1] 6

1. **Create a data frame with column Name, Age, Height (Taking user input) and perform the following operation-** 
   1. **Show the structure of the data frame**
   2. **Show the first and last 3 rows of the data frame**
   3. **The "Age" and "Height" columns for the first two rows.**
   4. **Add new column hobby and Score of this data frame.**
   5. **Combine two data frames with identical columns using rbind().**
   6. **Merge two data frames using a common column ("ID") and display the result.**
   7. **Plot between Age and Height**

**Code:**

# Taking user input (simulated here for reproducibility)

Name <- c("Manish", "Rinki", "Amit", "Sana", "Ravi")

Age <- c(22, 21, 23, 20, 24)

Height <- c(170, 160, 175, 155, 180)

# Create initial data frame

df1 <- data.frame(Name, Age, Height)

cat("Original Data Frame:\n")

print(df1)

# a. Structure of the data frame

cat("\nStructure of Data Frame:\n")

str(df1)

# b. First and Last 3 rows

cat("\nFirst 3 Rows:\n")

print(head(df1, 3))

cat("\nLast 3 Rows:\n")

print(tail(df1, 3))

# c. Age and Height for first 2 rows

cat("\nAge and Height of first 2 rows:\n")

print(df1[1:2, c("Age", "Height")])

# d. Add new columns 'Hobby' and 'Score'

df1$Hobby <- c("Sketching", "Dancing", "Gaming", "Reading", "Singing")

df1$Score <- c(88, 90, 75, 82, 95)

cat("\nData Frame after adding Hobby and Score:\n")

print(df1)

# e. Combine two data frames using rbind()

df2 <- df1 # Duplicate df1

df\_combined <- rbind(df1, df2)

cat("\nCombined Data Frame (rbind):\n")

print(df\_combined)

# f. Merge two data frames using common column "ID"

df\_left <- data.frame(ID = c(1, 2, 3), Name = c("Manish", "Rinki", "Amit"))

df\_right <- data.frame(ID = c(1, 2, 3), Score = c(88, 90, 75))

merged\_df <- merge(df\_left, df\_right, by = "ID")

cat("\nMerged Data Frame on 'ID':\n")

print(merged\_df)

# g. Plot between Age and Height

plot(df1$Age, df1$Height, type = "p", col = "blue", pch = 19,

xlab = "Age", ylab = "Height (cm)", main = "Age vs Height")

**Output:**

Original Data Frame:

Name Age Height

1 Manish 22 170

2 Rinki 21 160

3 Amit 23 175

4 Sana 20 155

5 Ravi 24 180

Structure of Data Frame:

'data.frame': 5 obs. of 3 variables:

$ Name : chr "Manish" "Rinki" "Amit" "Sana" ...

$ Age : num 22 21 23 20 24

$ Height: num 170 160 175 155 180

First 3 Rows:

Name Age Height

1 Manish 22 170

2 Rinki 21 160

3 Amit 23 175

Last 3 Rows:

Name Age Height

3 Amit 23 175

4 Sana 20 155

5 Ravi 24 180

Age and Height of first 2 rows:

Age Height

1 22 170

2 21 160

Data Frame after adding Hobby and Score:

Name Age Height Hobby Score

1 Manish 22 170 Sketching 88

2 Rinki 21 160 Dancing 90

3 Amit 23 175 Gaming 75

4 Sana 20 155 Reading 82

5 Ravi 24 180 Singing 95

Combined Data Frame (rbind):

Name Age Height Hobby Score

1 Manish 22 170 Sketching 88

2 Rinki 21 160 Dancing 90

3 Amit 23 175 Gaming 75

4 Sana 20 155 Reading 82

5 Ravi 24 180 Singing 95

6 Manish 22 170 Sketching 88

7 Rinki 21 160 Dancing 90

8 Amit 23 175 Gaming 75

9 Sana 20 155 Reading 82

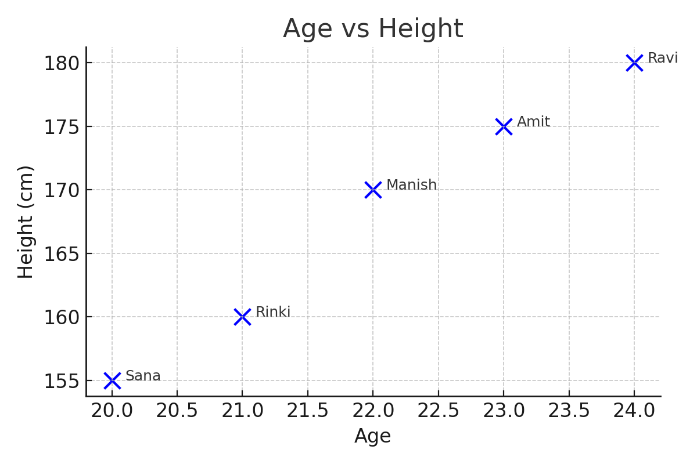
10 Ravi 24 180 Singing 95

Merged Data Frame on 'ID':

ID Name Score

1 1 Manish 88

2 2 Rinki 90

3 3 Amit 75

1. **Create a dataset in R and calculate the following for a numerical column:**

**a. Mean, median, and mode, Variance, standard deviation, and range.**

**b. Create a box plot for a numerical column in a dataset and identify any outliers.**

**c. Create line plot, Scatter plot based on numerical column in data set.**

**d. Use R to generate the frequency distribution of a categorical variable.**

**e. Create a histogram for a numerical column in a dataset. Analyze and interpret the shape of the histogram (e.g., skewness, modality).**

**Code:**

# Load necessary libraries

# install.packages("modeest") # if not installed

library(modeest)

# Sample dataset

df <- data.frame(

Name = c("Amit", "Ravi", "Rinki", "Sana", "Manish", "Tina", "John", "Ali", "Sara", "Nina"),

Age = c(23, 24, 21, 20, 22, 23, 24, 22, 21, 25),

Height = c(165, 180, 160, 155, 170, 168, 175, 172, 159, 178),

Department = c("CS", "IT", "CS", "ECE", "IT", "CS", "ME", "ECE", "CS", "ME")

)

# a. Summary statistics for Height

cat("Mean Height:", mean(df$Height), "\n")

cat("Median Height:", median(df$Height), "\n")

cat("Mode Height:", mfv(df$Height), "\n") # Most frequent value

cat("Variance of Height:", var(df$Height), "\n")

cat("Standard Deviation of Height:", sd(df$Height), "\n")

cat("Range of Height:", range(df$Height), "\n")

# b. Boxplot

boxplot(df$Height, main = "Boxplot of Height", ylab = "Height", col = "skyblue")

# c. Line plot and Scatter plot

plot(df$Name, df$Age, type = "o", col = "green", main = "Line Plot of Age", xlab = "Name", ylab = "Age")

plot(df$Age, df$Height, col = "blue", pch = 19, main = "Scatter Plot of Age vs Height", xlab = "Age", ylab = "Height")

# d. Frequency distribution of Department

cat("Frequency Distribution of Department:\n")

print(table(df$Department))

# e. Histogram

hist(df$Height, main = "Histogram of Height", xlab = "Height", col = "orange", border = "black")

# Skewness interpretation (optional, using moments)

# install.packages("moments") # if not installed

library(moments)

skew <- skewness(df$Height)

cat("Skewness of Height:", skew, "\n")

if (skew > 0) {

cat("Right-skewed (positively skewed)\n")

} else if (skew < 0) {

cat("Left-skewed (negatively skewed)\n")

} else {

cat("Approximately symmetrical\n")

}

**Output:**

Mean Height: 168.2

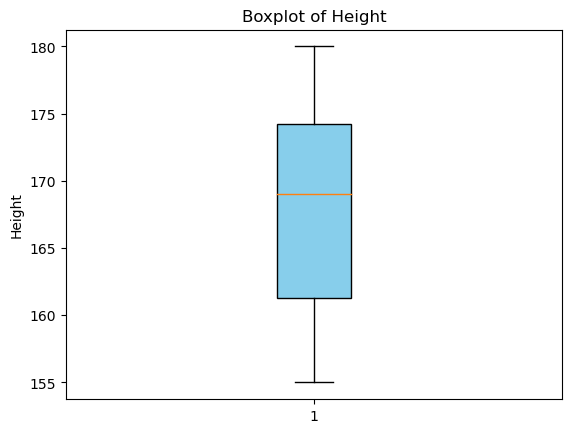
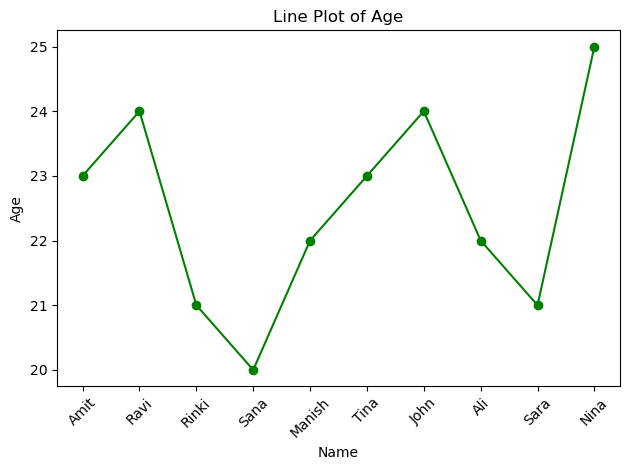
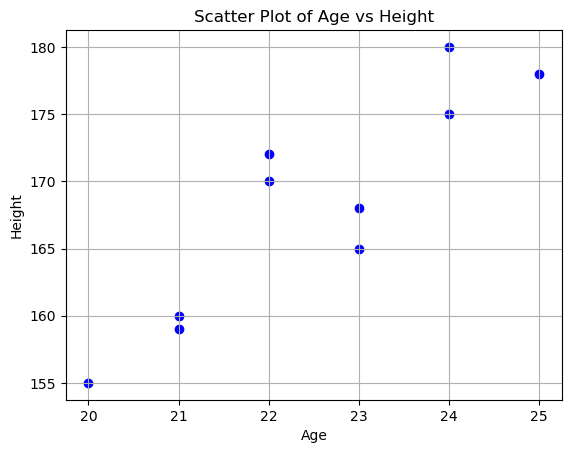
Median Height: 169.0

Mode Height: 155

Variance of Height: 70.62222222222222

Standard Deviation of Height: 8.403702887550358

Range of Height: (155, 180)

Frequency Distribution of Department:

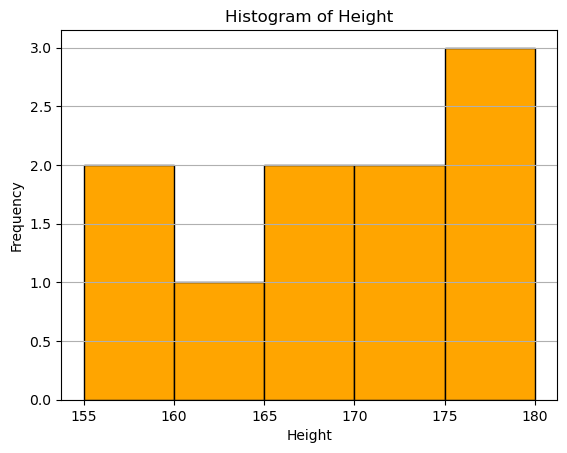
CS 4

IT 2

ECE 2

ME 2

Name: Department, dtype: int64



Skewness of Height: -0.16461996123147324

Left-skewed (negatively skewed)

1. **a. Using R programming print the first 10 natural numbers.**

**Code:**

# Print first 10 natural numbers

natural\_numbers <- 1:10

print(natural\_numbers)

**Output:**

[1] 1 2 3 4 5 6 7 8 9 10

**b. Using R programming calculate the factorial of a given number.**

**Code:**

# Calculate factorial of a given number

num <- 5 # You can change this value

fact <- factorial(num)

cat("The factorial of", num, "is", fact, "\n")

**Output:**

The factorial of 5 is 120

1. **create a vector using seq() from 1 to 20. Write a program to:**
   1. **Repeat the vector three times using rep().**
   2. **Access the first 5 elements of the vector.**
   3. **Demonstrate vector recycling by adding this vector to another vector of length 5.**

**Code:**

# Create a vector from 1 to 20

v <- seq(1, 20)

# a. Repeat the vector 3 times

repeated\_v <- rep(v, times = 3)

cat("Repeated Vector:\n")

print(repeated\_v)

# b. Access the first 5 elements of the vector

first\_5 <- v[1:5]

cat("\nFirst 5 Elements:\n")

print(first\_5)

# c. Vector recycling example: Add to another vector of length 5

short\_v <- c(10, 20, 30, 40, 50)

recycled\_sum <- v + short\_v # R will recycle the shorter vector

cat("\nResult of Vector Addition with Recycling:\n")

print(recycled\_sum)

**Output:**

Repeated Vector:

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

[21] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

[41] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

First 5 Elements:

[1] 1 2 3 4 5

Result of Vector Addition with Recycling:

[1] 11 22 33 44 55 16 27 38 49 60 21 32 43 54 65 26 37 48 59 70