1. **Mean:**

data <- c(12, 15, 18, 22, 25, 28, 31, 35, 38, 42)

mean\_value <- mean(data)

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

1. **Mode:**

get\_mode <- function(x) {

ux <- unique(x)

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

}

data <- c(12, 15, 18, 22, 22, 25, 28, 31, 35, 35)

mode\_value <- get\_mode(data)

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

1. **Median:**

data <- c(12, 15, 18, 22, 25, 28, 31, 35, 38, 42)

median\_value <- median(data)

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

1. **Quartile:**

data <- c(12, 15, 18, 22, 25, 28, 31, 35, 38, 42)

q1 <- quantile(data, 0.25)

q2 <- quantile(data, 0.50) # Also known as the median

q3 <- quantile(data, 0.75)

print(paste("Q1 (25th percentile):", q1))

print(paste("Q2 (50th percentile - Median):", q2))

print(paste("Q3 (75th percentile):", q3))

1. **Odd or Even :**

is\_odd\_or\_even <- function(number) {

if (number %% 2 == 0) {

return("Even")

} else {

return("Odd")

}

}

sample\_numbers <- c(12, 15, 18, 22, 25, 28, 31, 35, 38, 42)

for (number in sample\_numbers) {

print(paste("Number:", number, "|", "Type:", is\_odd\_or\_even(number)))

}

1. **Addition :**

num1 <- 25

num2 <- 37

result <- num1 + num2

print(paste("The sum of", num1, "and", num2, "is:", result))

1. **Subtraction**

num1 <- 55

num2 <- 27

result <- num1 - num2

print(paste("The result of", num1, "minus", num2, "is:", result))

1. **Multiplication**

num1 <- 6

num2 <- 8

result <- num1 \* num2

print(paste("The product of", num1, "and", num2, "is:", result))

1. **Division**

num1 <- 100

num2 <- 5

result <- num1 / num2

print(paste("The result of", num1, "divided by", num2, "is:", result))

**10.Normal Distribution**

set.seed(123) # For reproducibility

data <- rnorm(1000, mean = 0, sd = 1) # Mean = 0, Standard Deviation = 1

hist(data, main = "Normal Distribution", xlab = "Values", ylab = "Frequency", col = "lightblue")

**11.Min-Max**

data <- c(12, 15, 18, 22, 25, 28, 31, 35, 38, 42)

min\_value <- min(data)

max\_value <- max(data)

print(paste("Min value:", min\_value))

print(paste("Max value:", max\_value))

**12.K-Means**

set.seed(123) # For reproducibility

x <- rnorm(100)

y <- rnorm(100)

data <- cbind(x, y)

kmeans\_result <- kmeans(data, centers = 3)

print(kmeans\_result)

**13.Array**

arr <- array(data = 1:24, dim = c(2, 3, 4))

print(arr)

**14.Range**

data <- c(12, 15, 18, 22, 25, 28, 31, 35, 38, 42)

range\_value <- max(data) - min(data)

print(paste("Range:", range\_value))

**15.Square Root**

data <- c(4, 9, 16, 25, 36)

square\_root <- sqrt(data)

print(paste("Square roots of data:", paste(square\_root, collapse = ", ")))

**16.Pie chart**

labels <- c("A", "B", "C", "D", "E")

sizes <- c(20, 30, 10, 25, 15)

pie(sizes, labels = labels, main = "Pie Chart")

**17.Bar Chart**

categories <- c("A", "B", "C", "D", "E")

values <- c(20, 30, 10, 25, 15)

barplot(values, names.arg = categories, main = "Bar Chart", xlab = "Categories", ylab = "Values")

**18.Line Chart**

x <- c(1, 2, 3, 4, 5)

y <- c(2, 4, 6, 8, 10)

plot(x, y, type = "l", col = "blue", main = "Line Chart", xlab = "X", ylab = "Y")

**19.Box Plot**

data <- c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

boxplot(data, main = "Box Plot")

**20.Decision Tree**

install.packages("rpart")

install.packages("rpart.plot")

library(rpart)

library(rpart.plot)

data <- data.frame(x1 = c(1, 2, 3, 4, 5), x2 = c(2, 3, 4, 5, 6), y = c("A", "A", "B", "B", "B"))

tree\_model <- rpart(y ~ x1 + x2, data = data)

rpart.plot(tree\_model)

**21.Scatter plot**

x <- c(1, 2, 3, 4, 5)

y <- c(2, 4, 6, 8, 10)

plot(x, y, main = "Scatter Plot", xlab = "X", ylab = "Y")

**22.Linear Regression :**

x <- c(1, 2, 3, 4, 5)

y <- c(2, 4, 6, 8, 10)

lm\_model <- lm(y ~ x)

summary(lm\_model)

**23.Multiple regression:**

x1 <- c(1, 2, 3, 4, 5)

x2 <- c(2, 3, 4, 5, 6)

y <- c(3, 5, 7, 9, 11)

lm\_model <- lm(y ~ x1 + x2)

summary(lm\_model)

**24.Random Forest :**

install.packages("randomForest")

library(randomForest)

data <- iris[, 1:4] # Using iris dataset for demonstration

rf\_model <- randomForest(Species ~ ., data = data)

print(rf\_model)

**25.Histogram :**

data <- c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

hist(data, main = "Histogram", xlab = "Values", ylab = "Frequency")

**26.Confusion Matrix:**

actual <- c("A", "B", "A", "A", "B")

predicted <- c("A", "A", "B", "A", "B")

confusion\_matrix <- table(actual, predicted)

print(confusion\_matrix)

**27.Chi Square:**

observed <- c(20, 30, 10, 25, 15)

expected <- c(25, 25, 25, 25, 25) # Expected frequencies

chi\_square\_test <- chisq.test(observed, p = expected)

print(chi\_square\_test)

**28.Z- Score :**

data <- c(10, 20, 30, 40, 50)

mean\_value <- mean(data)

sd\_value <- sd(data)

z\_scores <- (data - mean\_value) / sd\_value

# Print z-scores

print(paste("Z-Scores:", paste(z\_scores, collapse = ", ")))

**29.Decimal Scaling :**

data <- c(100, 200, 300, 400, 500)

max\_value <- max(data)

decimal\_scaled <- data / max\_value

print(paste("Decimal Scaled Data:", paste(decimal\_scaled, collapse = ", ")))

**30.Apriori Algorithm:**

install.packages("arules")

library(arules)

transactions <- read.transactions(file = "transaction\_data.csv", format = "basket")

rules <- apriori(transactions, parameter = list(support = 0.1, confidence = 0.5))

inspect(rules)