1. Write R program to find the given number is Prime or not?

Input

# Function to check if a number is prime

isPrime <- function(n) {

if (n <= 1) {

return(FALSE) # Numbers less than or equal to 1 are not prime

}

if (n == 2) {

return(TRUE) # 2 is a prime number

}

if (n %% 2 == 0) {

return(FALSE) # Even numbers (except 2) are not prime

}

# Check divisibility by odd numbers starting from 3 up to sqrt(n)

for (i in 3:sqrt(n)) {

if (n %% i == 0) {

return(FALSE) # Found a divisor, number is not prime

}

}

return(TRUE) # Number is prime

}

# Test the function

number <- 17 # Change the number to test different values

if (isPrime(number)) {

print(paste(number, "is a prime number."))

} else {

print(paste(number, "is not a prime number."))

}

2. Write a program for creating a box plot using the vectors H=c(7,12,28,3,41) and M=c(“mar”, “apr”, “may”, “jun”, “jul”). Add a title to the chart as “Revenue chart”.

Input

H <- c(7, 12, 28, 3, 41)

M <- c("mar", "apr", "may", "jun", "jul")

# Create the box plot

boxplot(H, main = "Revenue chart", xlab = "Month", ylab = "Revenue")

# Set the x-axis labels

axis(1, at = 1:length(M), labels = M)

# Add a legend

legend("topright", legend = "Revenue", fill = "white", border = "white")

# Add a title

title(main = "Revenue chart")

3. Write a program for creating a pie-chart in r using the input vector(21,62,10,53). Provide labels for the chart as ‘london’, ‘new york’, ‘singapore’, ‘mumbai’. Add a title to the chart as ‘city piechart’ and add a legend at the top right corner of the chart.

Input

# Install the required package

install.packages("plotrix")

# Load the package

library(plotrix)

# Input vector and labels

values <- c(21, 62, 10, 53)

labels <- c('london', 'new york', 'singapore', 'mumbai')

# Create the pie chart

pie(values, labels = labels, main = "City Pie Chart")

# Add a legend at the top right corner

legend("topright", legend = labels, fill = rainbow(length(labels)), cex = 0.8)

4. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix.

Input:

# Create the vectors

vector1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)

vector2 <- c(10, 11, 12, 13, 14, 15, 16, 17, 18)

# Create the array of two matrices

array <- array(c(vector1, vector2), dim = c(3, 3, 2))

# Print the second row of the second matrix

second\_matrix <- array[, , 2]

second\_row <- second\_matrix[2, ]

print("Second row of the second matrix:")

print(second\_row)

# Print the element in the 3rd row and 3rd column of the first matrix

first\_matrix <- array[, , 1]

element\_3rd\_row\_3rd\_column <- first\_matrix[3, 3]

print("Element in the 3rd row and 3rd column of the first matrix:")

print(element\_3rd\_row\_3rd\_column)

5. Write R Program To Find The Given Number Is Positive Or Negative

Input:

# Function to check if a number is positive or negative

checkNumber <- function(number) {

if (number > 0) {

print("The number is positive.")

} else if (number < 0) {

print("The number is negative.")

} else {

print("The number is zero.")

}

}

# Example usage

num <- -7

checkNumber(num)

6. Write a R code for histogram charts for the below given age Attribute. Age :

9,13,21,8,36,22,12,41,31,33,19. Add a title to the chart as ‘Age’ and add color, border, x and y

limits.

# Age data

age <- c(9, 13, 21, 8, 36, 22, 12, 41, 31, 33, 19)

# Histogram chart

hist(age,

main = "Age", # Title of the chart

col = "skyblue", # Color of bars

border = "black", # Border color of bars

xlim = c(0, 45), # X-axis limits

ylim = c(0, 4), # Y-axis limits

xlab = "Age", # X-axis label

ylab = "Frequency", # Y-axis label

breaks = seq(0, 45, by = 5), # Breaks for bins

freq = TRUE # Show frequencies on y-axis

)

7. a. Melt ‘airquality’ data set which inbuild dataset in ‘R’ and display as a long – format data?

b. Melt air quality data and specify month and day to be “ID variables”?

c. Cast the molten ‘airquality’ data set.

d. Use cast function appropriately and compute the average of Ozone, Solar, Wind and temperature per

month?

e. Create a boxplot for ozone reading of ‘airquality’ dataset. Add title, label and color.

Input:

# Load the "airquality" dataset

data(airquality)

# (a) Melt 'airquality' dataset into long-format data

melted\_df <- melt(airquality)

print("(a) Melted 'airquality' dataset:")

print(melted\_df)

# (b) Melt air quality data and specify month and day as ID variables

melted\_id\_df <- melt(airquality, id.vars = c("Month", "Day"))

print("(b) Melted 'airquality' dataset with Month and Day as ID variables:")

print(melted\_id\_df)

# (c) Cast the molten 'airquality' dataset

cast\_df <- dcast(melted\_df, formula = variable ~ Month + Day)

print("(c) Casted 'airquality' dataset:")

print(cast\_df)

# (d) Compute the average of Ozone, Solar.R, Wind, and Temp per month using cast function

cast\_avg\_df <- dcast(melted\_id\_df, formula = Month ~ variable, fun.aggregate = mean)

print("(d) Average of Ozone, Solar.R, Wind, and Temp per month:")

print(cast\_avg\_df)

# (e) Create a boxplot for ozone reading of 'airquality' dataset

boxplot(airquality$Ozone, main = "Boxplot of Ozone Reading",

xlab = "Ozone", ylab = "Reading", col = "skyblue")

8. Write a R program:

i) Sort a vector in ascending and descending order

ii) To find sum, mean and product of vectors

iii) To create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60

and sum of numbers from 51 to 91.

iv) To calculate the area of Rectangle

input:

# i) Sort a vector in ascending and descending order

vector <- c(9, 2, 5, 1, 7, 3, 6, 4, 8)

ascending\_order <- sort(vector)

descending\_order <- sort(vector, decreasing = TRUE)

print("Sorted vector in ascending order:")

print(ascending\_order)

print("Sorted vector in descending order:")

print(descending\_order)

# ii) To find sum, mean and product of vectors

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

vector\_sum <- sum(vector)

vector\_mean <- mean(vector)

vector\_product <- prod(vector)

print("Sum of the vector:")

print(vector\_sum)

print("Mean of the vector:")

print(vector\_mean)

print("Product of the vector:")

print(vector\_product)

# iii) To create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

sequence\_20\_50 <- seq(20, 50)

mean\_20\_60 <- mean(seq(20, 60))

sum\_51\_91 <- sum(seq(51, 91))

print("Sequence of numbers from 20 to 50:")

print(sequence\_20\_50)

print("Mean of numbers from 20 to 60:")

print(mean\_20\_60)

print("Sum of numbers from 51 to 91:")

print(sum\_51\_91)

# iv) To calculate the area of a Rectangle

calculate\_rectangle\_area <- function(length, width) {

area <- length \* width

return(area)

}

rectangle\_length <- 5

rectangle\_width <- 10

rectangle\_area <- calculate\_rectangle\_area(rectangle\_length, rectangle\_width)

print("Area of the Rectangle:")

print(rectangle\_area)

9. Generate the following matrix 1 6 21 26 2 7 22 27 3 8 23 28 4 9 24 29 5 10 25 30 (i)Find the mean of each row of the above matrix. (ii)Compute the median (iii)Compute the sum of first 2 columns.

Input:

# Generate the matrix

matrix <- matrix(c(1, 6, 21, 26,

2, 7, 22, 27,

3, 8, 23, 28,

4, 9, 24, 29,

5, 10, 25, 30), nrow = 5, byrow = TRUE)

# (i) Find the mean of each row

row\_means <- rowMeans(matrix)

row\_means

# (ii) Compute the median

medians <- apply(matrix, 1, median)

medians

# (iii) Compute the sum of the first 2 columns

column\_sums <- colSums(matrix[, 1:2])

column\_sums

10. Write a program for creating a pie-chart in R using the input vector(21,62,10,53). Provide labels for the chart as ‘London’, ‘New York’, ‘Singapore’, ‘Mumbai’. Add a title to the chart as ‘city pie-chart’ and add a legend at the top right corner of the chart.

Input:

# Input vector and labels

values <- c(21, 62, 10, 53)

labels <- c("London", "New York", "Singapore", "Mumbai")

# Create the pie chart

pie(values, labels = labels, main = "City Pie Chart")

# Add a legend at the top right corner

legend("topright", legend = labels, fill = rainbow(length(labels)))

11. (i) Write a function in R programming to find a factorial of a given number. (ii) Find sum of natural numbers up-to 10, without formula using loop statement.

i) factorial <- function(n) {

if (n == 0) {

return(1)

} else {

return(n \* factorial(n - 1))

}

}

# Call the factorial function

result <- factorial(5)

result

ii) sum <- 0

for (i in 1:10) {

sum <- sum + i

}

Sum

12. For this exercise, use the (built-in) dataset Titanic. a. Draw a Bar chart to show details of “Survived” on the Titanic based on passenger Class b. Modify the above plot based on gender of people who survived c. Draw histogram plot to show distribution of feature “Age”

Input:

# Load the Titanic dataset (assuming it's available in the workspace)

data(Titanic)

# a. Bar chart for "Survived" based on passenger class

barplot(table(Titanic$Survived, Titanic$Class), beside = TRUE,

legend.text = TRUE, main = "Survived on the Titanic by Passenger Class")

# b. Bar chart for "Survived" based on passenger class and gender

barplot(table(Titanic$Survived, Titanic$Class, Titanic$Sex), beside = TRUE,

legend.text = TRUE, main = "Survived on the Titanic by Passenger Class and Gender")

# c. Histogram plot for the distribution of "Age"

hist(Titanic$Age, main = "Distribution of Age on the Titanic")

13. . A student recorded his/her scores on weekly R programming quizzes that were marked out of a

possible 10 points. His/Herscores were as follows:

8, 5, 8, 5, 7, 6, 7, 7, 5, 7, 5, 5, 6, 6, 9, 8, 9, 7, 9, 9, 6, 8, 6, 6, 7

What is the mean, median and mode of his/her scores on the weekly R programming quizzes?

Input:

# Student scores on weekly R programming quizzes

scores <- c(8, 5, 8, 5, 7, 6, 7, 7, 5, 7, 5, 5, 6, 6, 9, 8, 9, 7, 9, 9, 6, 8, 6, 6, 7)

# Calculate the mean score

mean\_score <- mean(scores)

# Calculate the median score

median\_score <- median(scores)

# Calculate the mode (most frequent score)

mode\_score <- names(table(scores))[table(scores) == max(table(scores))]

# Print the results

print(paste("Mean Score:", mean\_score))

print(paste("Median Score:", median\_score))

print(paste("Mode Score:", mode\_score))

14. The merchant sale a products such as tee-shirt, pant, jeans etc., if he sale the above the product

quantities between 20 and 30, then the print “Average day”. If quantity is > 30 then print “What a

great day!”, otherwise “Not enough for today”. Write the R code using if-else-if control

statements.

Input:

# Quantity of products sold

quantity <- 25

# Check the quantity and print the corresponding message

if (quantity > 30) {

print("What a great day!")

} else if (quantity >= 20 && quantity <= 30) {

print("Average day")

} else {

print("Not enough for today")

}

15. User

Create R matrix x=3,2,4,5 and y= 6,7,5,4. Find the following and check ur answers in R

a. x<-matrix (3 2 4 5, 2:2)

b. x%\*%y

c. y\*y

d. 2\*x

input:

# Define the matrices

x <- matrix(c(3, 2, 4, 5), nrow = 2, ncol = 2)

y <- matrix(c(6, 7, 5, 4), nrow = 2, ncol = 2)

# a. Selecting elements from x

selected\_elements <- x[2, 2]

# b. Matrix multiplication of x and y

matrix\_multiplication <- x %\*% y

# c. Element-wise multiplication of y with itself

elementwise\_multiplication <- y \* y

# d. Scalar multiplication of 2 with x

scalar\_multiplication <- 2 \* x

# Print the results

print("Selected Elements from x:")

print(selected\_elements)

print("Matrix Multiplication of x and y:")

print(matrix\_multiplication)

print("Element-wise Multiplication of y with itself:")

print(elementwise\_multiplication)

print("Scalar Multiplication of 2 with x:")

print(scalar\_multiplication)

16. Describe how histogram charts are created in R. Create a histogram chart for the below given age

attribute.

Age : 5,45,23,30,33,32,34,35,42,41,28,29

ii) Create a 3D Pie Chart for the dataset “political Knowledge” with suitable labels and

colour

input:

i)# Age attribute

age <- c(5, 45, 23, 30, 33, 32, 34, 35, 42, 41, 28, 29)

# Create a histogram chart

hist(age,

main = "Histogram of Age",

xlab = "Age",

ylab = "Frequency",

col = "lightblue", # Set color of bars

border = "black", # Set border color of bars

xlim = c(0, 50), # Set x-axis limits

ylim = c(0, 4), # Set y-axis limits

breaks = 5 # Set number of bins

)

ii)Load the required package

library(plotrix)

# Political Knowledge dataset

political\_knowledge <- c(25, 35, 15, 10, 5, 10)

# Create labels for the chart

labels <- c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree", "Not Sure")

# Set colors for the chart

colors <- c("red", "orange", "yellow", "green", "blue", "purple")

# Create a 3D Pie Chart

pie3D(political\_knowledge,

labels = labels,

explode = 0.1, # Set explode value for a 3D effect

main = "Political Knowledge",

col = colors # Set colors for the slices

)

17. Write R Program To Print The Sequence Of Numbers (1,2,….,10) Using Repeat loop.

Input:

# Initialize the counter

i <- 1

# Use the repeat loop to print the sequence of numbers

repeat {

# Print the current number

print(i)

# Increment the counter

i <- i + 1

# Check the exit condition

if (i > 10) {

break # Exit the loop if the counter exceeds 10

}

}

18. Write a R program to create inner, outer, left, right join(merge) from given two data frames.

df1 = data.frame(numid = c(12, 14, 10, 11))

df2 = data.frame(numid = c(13, 15, 11, 12))

input:

# Create the data frames

df1 <- data.frame(numid = c(12, 14, 10, 11))

df2 <- data.frame(numid = c(13, 15, 11, 12))

# Inner join (merge)

inner\_join <- merge(df1, df2, by = "numid", all = FALSE)

print("Inner Join:")

print(inner\_join)

# Outer join (merge)

outer\_join <- merge(df1, df2, by = "numid", all = TRUE)

print("Outer Join:")

print(outer\_join)

# Left join (merge)

left\_join <- merge(df1, df2, by = "numid", all.x = TRUE)

print("Left Join:")

print(left\_join)

# Right join (merge)

right\_join <- merge(df1, df2, by = "numid", all.y = TRUE)

print("Right Join:")

print(right\_join)

19. The table below shows one year of marketing spend and company sales by month.

Month 1 2 3 4 5 6 7 8 9 10 11 12

Spends 1000 4000 5000 4500 3000 4000 9000 11000 15000 12000 7000 3000

Sales 9914 40487 54324 50044 34719 42551 94871 118914 158484 131348 78504 36284

Create a regression model to show the amount of sales(Sales) based on the how much the company spends

(Spends) in advertising. Predict the Sales if Spend=13500

Input:

# Create the data frame

data <- data.frame(

Month = 1:12,

Spends = c(1000, 4000, 5000, 4500, 3000, 4000, 9000, 11000, 15000, 12000, 7000, 3000),

Sales = c(9914, 40487, 54324, 50044, 34719, 42551, 94871, 118914, 158484, 131348, 78504, 36284)

)

# Create the regression model

model <- lm(Sales ~ Spends, data = data)

# Print the model summary

summary(model)

# Predict sales for Spend = 13500

new\_data <- data.frame(Spends = 13500)

predicted\_sales <- predict(model, newdata = new\_data)

print("Predicted Sales:")

print(predicted\_sales)

20. For this exercise, use the (built-in) dataset Titanic.

a. Draw a Bar chart to show details of “Survived” on the Titanic based on passenger Class

b. Modify the above plot based on gender of people who survived

c. Draw histogram plot to show distribution of feature “Age”

input:

i)# Load the Titanic dataset

data(Titanic)

# Create a bar chart of "Survived" based on passenger class

barplot(Titanic$Survived, beside = TRUE, legend.text = TRUE, names.arg = colnames(Titanic), xlab = "Passenger Class", ylab = "Count", main = "Survived on Titanic by Passenger Class")

ii)# Create a bar chart of "Survived" based on passenger class and gender

barplot(Titanic$Survived, beside = TRUE, legend.text = TRUE, names.arg = colnames(Titanic), xlab = "Passenger Class", ylab = "Count", main = "Survived on Titanic by Passenger Class and Gender", col = c("blue", "pink"), args.legend = list(x = "topleft"))

iii)# Load the Titanic dataset

data(Titanic)

# Extract the "Age" column

age <- Titanic$Age

# Create a histogram plot of the "Age" distribution

hist(age, breaks = 10, col = "lightblue", xlab = "Age", ylab = "Frequency", main = "Distribution of Age on Titanic")

21. Write a R program to create a matrix - m(c(1:16), nrow=4,byrow = TRUE) taking a given vector

of numbers as input and define the column and row names. Access the element at 3rd column

and 2nd row, only the 3rd row and only the 4th column of a given matrix

input:

# Create a matrix with given vector and define column and row names

vector <- c(1:16)

matrix <- matrix(vector, nrow = 4, byrow = TRUE)

colnames(matrix) <- c("A", "B", "C", "D")

rownames(matrix) <- c("X", "Y", "Z", "W")

# Access the element at 3rd column and 2nd row

element1 <- matrix[2, 3]

# Access only the 3rd row

row3 <- matrix[3, ]

# Access only the 4th column

column4 <- matrix[, 4]

# Print the results

print("Element at 3rd column and 2nd row:")

print(element1)

print("3rd row:")

print(row3)

print("4th column:")

print(column4)

22. Using linear regression analysis establish a relationship between height and weight of a person

using the input vector given below.

# values of height -> 151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# values of weight. -> 63, 81, 56, 91, 47, 57, 76, 72, 62, 48

Input:

# Define the input vectors

height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Create a data frame with height and weight

data <- data.frame(height, weight)

# Perform linear regression analysis

model <- lm(weight ~ height, data = data)

# Print the regression coefficients and summary

print("Regression Coefficients:")

print(coef(model))

print("Regression Summary:")

summary(model)

23. each of the 50 states in 1973. It also contains the percentage of people in the state who live in an

urban area.

(i) a. Explore the summary of Data set, like number of Features and its type. Find the number of

records for each feature. Print the statistical feature of data

b. Print the state which saw the largest total number of rape

c. Print the states with the max & min crime rates for murder

input:

# Summary of the dataset

summary(state\_data)

# Number of features and their types

num\_features <- ncol(state\_data)

feature\_types <- sapply(state\_data, class)

cat("Number of features:", num\_features, "\n")

cat("Feature types:", paste(feature\_types, collapse = ", "), "\n")

# Number of records for each feature

feature\_records <- sapply(state\_data, length)

cat("Number of records for each feature:\n")

print(feature\_records)

# Statistical features of the data

statistical\_features <- sapply(state\_data, function(x) c(min(x), max(x), mean(x), median(x)))

cat("Statistical features of the data:\n")

print(statistical\_features)

# State with the largest total number of rape

state\_with\_max\_rape <- state\_data$State[which.max(state\_data$Rape)]

cat("State with the largest total number of rape:", state\_with\_max\_rape, "\n")

# States with the max and min crime rates for murder

state\_with\_max\_murder <- state\_data$State[which.max(state\_data$Murder)]

state\_with\_min\_murder <- state\_data$State[which.min(state\_data$Murder)]

cat("State with the max crime rate for murder:", state\_with\_max\_murder, "\n")

cat("State with the min crime rate for murder:", state\_with\_min\_murder, "\n")

24. (i) Write a R program to extract the five of the levels of factor created from a random sample

from the LETTERS (Part of the base R distribution.)

(ii)Write R function to find the range of given vector. Range=Max-Min

Sample input, C<-(9,8,7,6,5,4,3,2,1),

output=8

input:

i)set.seed(123) # Setting seed for reproducibility

random\_sample <- sample(LETTERS, size = 10) # Creating a random sample from LETTERS

factor\_levels <- factor(random\_sample) # Creating a factor from the random sample

five\_levels <- levels(factor\_levels)[1:5] # Extracting the first five levels

print(five\_levels)

ii)find\_range <- function(vec) {

range <- max(vec) - min(vec)

return(range)

}

# Example usage

C <- c(9, 8, 7, 6, 5, 4, 3, 2, 1)

range\_value <- find\_range(C)

print(range\_value)

25. Write the R CODE for the loop next and break statements.

Input:

# Example using the 'next' statement

for (i in 1:10) {

if (i == 5) {

next # Skip the current iteration and move to the next iteration

}

print(i)

}

# Example using the 'break' statement

for (i in 1:10) {

if (i == 5) {

break # Exit the loop when i reaches 5

}

print(i)

}

26. Create a data frame for the given table.

Name Score Attempts Qualify

Shan 12.5 1 Yes

Ash 9.0 NA No

Malu 16.5 2 Yes

Viji 12.5 NA No

Write a R program to extract 2

rd and 4

th rows with 1

st and 3

rd columns from a given data frame.

Input:

# Create the data frame

df <- data.frame(

Name = c("Shan", "Ash", "Malu", "Viji"),

Score = c(12.5, 9.0, 16.5, 12.5),

Attempts = c(1, NA, 2, NA),

Qualify = c("Yes", "No", "Yes", "No"),

stringsAsFactors = FALSE

)

# Print the data frame

print(df)

# Extract the desired rows and columns

subset\_df <- df[c(2, 4), c(1, 3)]

# Print the subset data frame

print(subset\_df)

27. For this exercise, use the (built-in) dataset Titanic.

a. Draw a Bar chart to show details of “Survived” on the Titanic based on passenger Class

b. Modify the above plot based on gender of people who survived

c. Draw histogram plot to show distribution of feature “Age”

input:

i)# Load the Titanic dataset

data(Titanic)

# Create a bar chart

barplot(Titanic$Survived, beside = TRUE, legend.text = rownames(Titanic),

col = c("red", "blue"), main = "Survival on the Titanic by Passenger Class",

xlab = "Passenger Class", ylab = "Count")

ii)# Create a subset of the dataset for survivors

survivors <- Titanic[, , "Survived"]

row.names(survivors) <- rownames(Titanic)

# Create a stacked bar chart by gender

barplot(survivors, beside = TRUE, legend.text = rownames(survivors),

col = c("red", "blue"), main = "Survival on the Titanic by Passenger Class and Gender",

xlab = "Passenger Class", ylab = "Count")

iii)# Extract the "Age" variable from the Titanic dataset

age <- Titanic[,, "Age"]

# Create a histogram

hist(age, main = "Distribution of Age on the Titanic", xlab = "Age", ylab = "Frequency")

28. a. Create a 6 × 10 matrix of random integers chosen in the range of from 1:10

b. Find the number of entries in each row which are greater than 4.

c. Which rows contain exactly two occurrences of the number 7?

Input:

set.seed(75)

matrix <- matrix(sample(1:10, 6\*10, replace = TRUE), nrow = 6)

print(matrix)

ii)greater\_than\_4 <- apply(matrix, 1, function(row) sum(row > 4))

print(greater\_than\_4)

iii)two\_occurrences\_7 <- which(apply(matrix == 7, 1, function(row) sum(row) == 2))

print(two\_occurrences\_7)

29. Write R code to create a bar chart where Bike is assigned red , car is assigned yellow , bus

is assigned blue , auto is assigned black , and train is assigned white. Use respective

parameters such xlab, ylab, main, labels, color etc.,

input:

# Create a vector of categories

categories <- c("Bike", "Car", "Bus", "Auto", "Train")

# Create a vector of corresponding colors

colors <- c("red", "yellow", "blue", "black", "white")

# Create a vector of corresponding values (e.g., number of occurrences)

values <- c(10, 15, 8, 12, 20)

# Create a bar plot with custom colors

barplot(values, names.arg = categories, col = colors,

xlab = "Categories", ylab = "Count",

main = "Distribution of Transportation Modes")

30. Create the vector with following numbers. v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)

Apply the mean, median, and mode in the above vector using R

Input:

# Create the vector

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

# Calculate the mean

mean\_value <- mean(v)

print(mean\_value)

# Calculate the median

median\_value <- median(v)

print(median\_value)

# Calculate the mode

mode\_value <- names(table(v))[table(v) == max(table(v))]

print(mode\_value)

31.. Write a program for creating a pie-chart in R using the input vector (21,62,10,53).

Provide labels for the chart as ‘London’, ‘New York’, ‘Singapore’, ‘Mumbai’. Add a

title to the chart as ‘city pie-chart’ and add a legend at the top right corner of the chart.

b. Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and

M=c(“mar”, “apr”, “may”, “jun”, “jul”). Add a title to the chart as “Revenue chart”

Input:

# Input vector

data <- c(21, 62, 10, 53)

# Labels for the chart

labels <- c('London', 'New York', 'Singapore', 'Mumbai')

# Create the pie chart

pie(data, labels = labels, main = 'City Pie-Chart')

# Add legend

legend('topright', legend = labels, title = 'Cities', fill = c('red', 'green', 'blue', 'orange'))

b)# Input vectors

H <- c(7, 12, 28, 3, 41)

M <- c("mar", "apr", "may", "jun", "jul")

# Create the bar chart

barplot(H, names.arg = M, main = "Revenue Chart")

32. a. Write suitable R code to compute the mean, median, mode of the following values

c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20,75,70,10,60,70,85,95,55,15)

b. Write R code to find 2nd highest and 4th Lowest value of above problem

input:

# Input vector

values <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20, 75, 70, 10, 60, 70, 85, 95, 55, 15)

# Mean

mean\_value <- mean(values)

# Median

median\_value <- median(values)

# Mode

mode\_value <- names(table(values))[table(values) == max(table(values))]

# Print the results

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

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

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

b)

# Input vector

values <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20, 75, 70, 10, 60, 70, 85, 95, 55, 15)

# Sort the vector in ascending order

sorted\_values <- sort(values)

# Find the 2nd highest value

second\_highest <- sorted\_values[length(sorted\_values) - 1]

# Find the 4th lowest value

fourth\_lowest <- sorted\_values[4]

# Print the results

print(paste("2nd Highest:", second\_highest))

print(paste("4th Lowest:", fourth\_lowest))

33. Create First Dataset with variables -> Surname, nationality

Create Second Dataset with variables -> Surname, movies

The common key variable is surname. How to merge both data and check if the dimensionality is

7x3.

Input:

# Create the first dataset

df1 <- data.frame(

Surname = c("Smith", "Johnson", "Williams", "Jones", "Brown", "Davis", "Miller"),

Nationality = c("USA", "UK", "USA", "Australia", "USA", "Canada", "USA")

)

# Create the second dataset

df2 <- data.frame(

Surname = c("Smith", "Johnson", "Williams", "Jones", "Brown", "Davis", "Miller"),

Movies = c("Action", "Drama", "Comedy", "Thriller", "Sci-Fi", "Romance", "Horror")

)

# Merge the datasets based on the common key variable "Surname"

merged\_df <- merge(df1, df2, by = "Surname")

# Check the dimensionality of the merged dataset

dim(merged\_df)

34.Write a R program to call the (built-in) dataset airquality. Check whether it is a data frame

or not? Order the entire data frame by the first and second column.

Input:

# Call the built-in dataset airquality

data(airquality)

# Check if it is a data frame

if (is.data.frame(airquality)) {

print("airquality is a data frame")

} else {

print("airquality is not a data frame")

}

# Order the entire data frame by the first and second column

ordered\_df <- airquality[order(airquality[, 1], airquality[, 2]), ]

# Print the ordered data frame

print(ordered\_df)

35. Load dataset named ChickWeight,

i.Order the data frame, in ascending order by feature name “weight” grouped by feature “diet”

and Extract the last 6 records from order data frame.

(ii).a Perform melting function based on “Chick", "Time", "Diet" features as ID variables

b. Perform cast function to display the mean value of weight grouped by Diet

c. Perform cast function to display the mode of weight grouped by Diet

(iii)a. Create Box plot for “weight” grouped by “Diet”

b. Create a Histogram for “weight” features belong to Diet- 1 category

c. Create Scatter plot for “ weight” vs “Time” grouped by Diet

input:

# Load the ChickWeight dataset

data(ChickWeight)

# Order the data frame by "weight" grouped by "diet"

ordered\_df <- ChickWeight[order(ChickWeight$weight), ]

# Extract the last 6 records

last\_six <- tail(ordered\_df, 6)

# Print the last 6 records

print(last\_six)

b)# Load the reshape2 library

library(reshape2)

# Melting the data frame based on "Chick", "Time", "Diet" as ID variables

melted\_df <- melt(ChickWeight, id.vars = c("Chick", "Time", "Diet"))

# Perform cast function to display the mean value of weight grouped by Diet

mean\_weight <- dcast(melted\_df, Diet ~ variable, fun.aggregate = mean)

# Perform cast function to display the mode of weight grouped by Diet

mode\_weight <- dcast(melted\_df, Diet ~ variable, fun.aggregate = function(x) {names(table(x))[which.max(table(x))]})

c)Load the ggplot2 library

library(ggplot2)

# (a) Create a boxplot for "weight" grouped by "Diet"

boxplot\_plot <- ggplot(ChickWeight, aes(x = factor(Diet), y = weight)) +

geom\_boxplot() +

xlab("Diet") +

ylab("Weight") +

ggtitle("Boxplot of Weight Grouped by Diet")

# (b) Create a histogram for "weight" in the Diet-1 category

histogram\_plot <- ggplot(ChickWeight[ChickWeight$Diet == 1, ], aes(x = weight)) +

geom\_histogram(binwidth = 5) +

xlab("Weight") +

ylab("Frequency") +

ggtitle("Histogram of Weight in Diet-1")

# (c) Create a scatter plot for "weight" vs "Time" grouped by "Diet"

scatter\_plot <- ggplot(ChickWeight, aes(x = Time, y = weight, color = factor(Diet))) +

geom\_point() +

xlab("Time") +

ylab("Weight") +

ggtitle("Scatter Plot of Weight vs Time Grouped by Diet")

# Print the plots

print(boxplot\_plot)

print(histogram\_plot)

print(scatter\_plot)

36. (i)Write R function to find the range of given vector. Range=Max-Min

Sample input, C<-(9,8,7,6,5,4,3,2,1), output=8

(ii)Wirte the R function to find the number of vowels in given string

Sample input c<- “matrix”, output<-2

Input:

find\_range <- function(vector) {

range <- max(vector) - min(vector)

return(range)

}

# Test the function

C <- c(9, 8, 7, 6, 5, 4, 3, 2, 1)

output <- find\_range(C)

print(output)

37. Write a R program to add new row(s) to an existing data frame

Input:

# Existing data frame

existing\_df <- data.frame(Name = c("John", "Emily", "David"),

Age = c(25, 30, 35),

Gender = c("Male", "Female", "Male"))

# New row(s) to be added

new\_rows <- data.frame(Name = c("Sarah", "Michael"),

Age = c(28, 32),

Gender = c("Female", "Male"))

# Add new row(s) to the existing data frame

updated\_df <- rbind(existing\_df, new\_rows)

# Print the updated data frame

print(updated\_df)

38. Write an R program to create an array with three columns, three rows, and two "tables", taking

two vectors as input to the array. Print the array and perform the following.

a) A+B b) A-B c) t(A)

input:

# Input vectors

vector\_A <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)

vector\_B <- c(9, 8, 7, 6, 5, 4, 3, 2, 1)

# Create the array

my\_array <- array(c(vector\_A, vector\_B), dim = c(3, 3, 2))

# Print the array

print(my\_array)

# Perform operations

# a) A + B

result\_sum <- my\_array[, , 1] + my\_array[, , 2]

print("A + B:")

print(result\_sum)

# b) A - B

result\_diff <- my\_array[, , 1] - my\_array[, , 2]

print("A - B:")

print(result\_diff)

# c) Transpose of A

result\_transpose <- t(my\_array[, , 1])

print("Transpose of A:")

print(result\_transpose)

39. Randomly Sample the iris dataset such as 80% data for training and 20% for test and create

Logistics regression with train data,use species as target and petals width and

length as feature variables ,Predict the probability of the model using test data, Create Confusion

matrix for above test model

input:

# Load the iris dataset

data(iris)

# Set the random seed for reproducibility

set.seed(123)

# Randomly sample the iris dataset

sampled\_data <- iris[sample(nrow(iris)), ]

# Calculate the index to split the data into training and test sets

train\_size <- round(0.8 \* nrow(sampled\_data))

# Split the data into training and test sets

train\_data <- sampled\_data[1:train\_size, ]

test\_data <- sampled\_data[(train\_size + 1):nrow(sampled\_data), ]

# Create the logistic regression model using the training data

model <- glm(Species ~ Petal.Width + Petal.Length, data = train\_data, family = "binomial")

# Predict probabilities using the test data

predicted\_probs <- predict(model, newdata = test\_data, type = "response")

# Convert probabilities to predicted classes

predicted\_classes <- ifelse(predicted\_probs > 0.5, "versicolor", "setosa")

# Create the confusion matrix

confusion\_matrix <- table(predicted\_classes, test\_data$Species)

print(confusion\_matrix)

40. a. Create a data frame based on below table. Month

Month 1 2 3 4 5 6 7 8 9 10 11 12

Spends 1000 4000 5000 4500 3000 4000 9000 11000 15000 12000 7000 3000

Sales 9914 40487 54324 50044 34719 42551 94871 118914 158484 131348 78504 36284

b. Create a regression model for that data frame table to show the amount of sales(Sales) based

on the how much the company spends (Spends) in advertising

c. Predict the Sales if Spend=13500

input:

# Create the data frame

df <- data.frame(Month = 1:12,

Spends = c(1000, 4000, 5000, 4500, 3000, 4000, 9000, 11000, 15000, 12000, 7000, 3000),

Sales = c(9914, 40487, 54324, 50044, 34719, 42551, 94871, 118914, 158484, 131348, 78504, 36284))

b)# Create the regression model

model <- lm(Sales ~ Spends, data = df)

c) # Predict the sales for Spend = 13,500

new\_data <- data.frame(Spends = 13500)

predicted\_sales <- predict(model, newdata = new\_data)

41. A student recorded his/her scores on weekly R programming quizzes that were marked out of

a possible 10 points. His/Herscores were as follows:

8, 5, 8, 5, 7, 6, 7, 7, 5, 7, 5, 5, 6, 6, 9, 8, 9, 7, 9, 9, 6, 8, 6, 6, 7

What is the mean, median, mode of his/her scores on the weekly R programming quizzes?

Input:

# Student's scores

scores <- c(8, 5, 8, 5, 7, 6, 7, 7, 5, 7, 5, 5, 6, 6, 9, 8, 9, 7, 9, 9, 6, 8, 6, 6, 7)

# Mean

mean\_score <- mean(scores)

# Median

median\_score <- median(scores)

# Mode

mode\_score <- names(table(scores))[table(scores) == max(table(scores))]

# Print the results

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

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

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

42. Following table contains name and score of the cricket players.

name score

1 Raina 67

2 Bravo 56

3 Dhoni 87

4 Virat 91

a) Create a data frame for the above table1.

b) Write a R program to add a new column in a given data frame

c) Find the dimensions and structure of the data frame.

Input:

i)# Creating the data frame

data <- data.frame(

name = c("Raina", "Bravo", "Dhoni", "Virat"),

score = c(67, 56, 87, 91)

)

# Printing the data frame

print(data)

ii)# Adding a new column

data$new\_column <- c(10, 20, 30, 40)

# Printing the updated data frame

print(data)

iii)# Dimensions

dim(data)

# Structure

str(data)

43. Create a 6 × 10 matrix of random integers chosen in the range of from 1:10

b. Find the number of entries in each row which are greater than 4.

c. Which rows contain exactly two occurrences of the number 7?

Input:

a)#Set the seed for reproducibility

set.seed(1)

# Create the matrix

matrix\_data <- matrix(sample(1:10, 6\*10, replace = TRUE), nrow = 6, ncol = 10)

# Print the matrix

print(matrix\_data)

b)# Count the number of entries greater than 4 in each row

row\_counts <- apply(matrix\_data, 1, function(row) sum(row > 4))

# Print the row counts

print(row\_counts)

c)# Find the rows with exactly two occurrences of the number 7

rows\_with\_two\_sevens <- which(apply(matrix\_data, 1, function(row) sum(row == 7) == 2))

# Print the rows with two sevens

print(rows\_with\_two\_sevens)

44. a. Write a program for creating a pie-chart in R using the input vector (21,62,10,53). Provide

labels for the chart as ‘London’, ‘New York’, ‘Singapore’, ‘Mumbai’. Add a

title to the chart as ‘city pie-chart’ and add a legend at the top right corner of the chart.

b. Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and M=c(“mar”,

“apr”, “may”, “jun”, “jul”). Add a title to the chart as “Revenue chart”

Input:

# Input vector

values <- c(21, 62, 10, 53)

# Labels for the chart

labels <- c("London", "New York", "Singapore", "Mumbai")

# Title of the chart

title <- "City Pie Chart"

# Create the pie chart

pie(values, labels = labels, main = title)

# Add a legend at the top right corner

legend("topright", legend = labels, title = "Cities", fill = rainbow(length(values)))

# Display the chart

b) # Vectors for the chart

H <- c(7, 12, 28, 3, 41)

M <- c("mar", "apr", "may", "jun", "jul")

# Title of the chart

title <- "Revenue Chart"

# Create the bar chart

barplot(H, names.arg = M, main = title, xlab = "Month", ylab = "Revenue")

# Display the chart

45. Generate the following matrix

1 6 21 26

2 7 22 27

3 8 23 28

4 9 24 29

5 10 25 30

(i)find the mean of each row of the above matrix .

(ii)compute the median

(iii)compute the sum of first 2 columns.

Input:

# Generate the matrix

matrix <- matrix(c(1:5, 6:10, 21:25, 26:30), nrow = 5, byrow = TRUE)

# Print the matrix

print(matrix)

# (i) Find the mean of each row

row\_means <- apply(matrix, 1, mean)

print(row\_means)

# (ii) Compute the median

median\_value <- median(matrix)

print(median\_value)

# (iii) Compute the sum of the first 2 columns

sum\_first\_two\_columns <- sum(matrix[, 1:2])

print(sum\_first\_two\_columns)

46. Using linear regression analysis establish a relationship between height and weight of a

person using the input vector given below.

# values of height -> 151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# values of weight. -> 63, 81, 56, 91, 47, 57, 76, 72, 62, 48

Input:

# Input data

height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Create a data frame

data <- data.frame(height, weight)

# Perform linear regression

model <- lm(weight ~ height, data = data)

# Print the summary of the model

summary(model)

# Predict weight based on height

new\_height <- c(160, 170, 180) # New height values for prediction

predicted\_weight <- predict(model, newdata = data.frame(height = new\_height))

print(predicted\_weight)

47. Randomly Sample the iris dataset such as 80% data for training and 20% for test and create

Logistics regression with train data,use species as target and petals width and

length as feature variables ,Predict the p

input:

# Load the iris dataset

data(iris)

# Set the seed for reproducibility

set.seed(123)

# Randomly sample the dataset into training and test sets

train\_index <- sample(1:nrow(iris), 0.8 \* nrow(iris))

train\_data <- iris[train\_index, ]

test\_data <- iris[-train\_index, ]

# Create a logistic regression model

model <- glm(Species ~ Petal.Length + Petal.Width, data = train\_data, family = "binomial")

# Predict the probabilities using the test data

probabilities <- predict(model, newdata = test\_data, type = "response")

# Print the predicted probabilities

print(probabilities)

48. For this exercise, use the (built-in) dataset Titanic.

a. Draw a Bar chart to show details of “Survived” on the Titanic based on passenger Class

b. Modify the above plot based on gender of people who survived

c. Draw histogram plot to show distribution of feature “Age”

input:

49. Create First Dataset with variables -> Surname, nationality

Create Second Dataset with variables -> Surname, movies

The common key variable is surname. How to merge both data and check if the dimensionality is

7x3.

Input:

# Create the first dataset

first\_dataset <- data.frame(

Surname = c("Smith", "Johnson", "Brown", "Davis", "Wilson"),

Nationality = c("USA", "UK", "Australia", "USA", "Canada")

)

# Create the second dataset

second\_dataset <- data.frame(

Surname = c("Smith", "Johnson", "Brown", "Davis", "Wilson"),

Movies = c("Action", "Drama", "Comedy", "Thriller", "Sci-Fi")

)

# Merge the datasets based on the common key variable "Surname"

merged\_data <- merge(first\_dataset, second\_dataset, by = "Surname")

# Check the dimensionality of the merged dataset

dim(merged\_data)

50. Consider the below vector x

x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)

a. Find mean, median for the above “x” vector variable.

b. Apply the trim function to remove the negative values and find the mean value

c. Remove the NA value and find the mean value.

Input:

x <- c(12, 7, 3, 4.2, 18, 2, 54, -21, 8, -5, NA)

# Find the mean of x

mean\_x <- mean(x, na.rm = TRUE)

print(mean\_x)

# Find the median of x

median\_x <- median(x, na.rm = TRUE)

print(median\_x)

b) library(DescTools)

trimmed\_x <- trim(x, lower = 0)

mean\_trimmed\_x <- mean(trimmed\_x, na.rm = TRUE)

print(mean\_trimmed\_x)

c) x\_without\_na <- na.omit(x)

mean\_without\_na <- mean(x\_without\_na)

print(mean\_without\_na)

51. Load dataset named ChickWeight,

i.Order the data frame, in ascending order by feature name “weight” grouped by feature “diet”

and Extract the last 6 records from order data frame.

(ii).a Perform melting function based on “Chick", "Time", "Diet" features as ID variables

b. Perform cast function to display the mean value of weight grouped by Diet

c. Perform cast function to display the mode of weight grouped by Diet

input:

Load the ChickWeight dataset

data(ChickWeight)

# Order the data frame by "weight" grouped by "diet"

ordered\_data <- ChickWeight[order(ChickWeight$weight), ]

# Extract the last 6 records

last\_six\_records <- tail(ordered\_data, 6)

print(last\_six\_records)

b) library(reshape2)

# Perform melting

melted\_data <- melt(ChickWeight, id.vars = c("Chick", "Time", "Diet"))

print(melted\_data)

52. Write suitable R code to compute the mean, median ,mode of the following values

c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20,75,70,10,60,70,85,95,55,15)

b. Write R code to find 2nd highest and 4th Lowest value of above problem

input:

a) # Define the vector of values

values <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20, 75, 70, 10, 60, 70, 85, 95, 55, 15)

# Compute the mean

mean\_value <- mean(values)

# Compute the median

median\_value <- median(values)

# Compute the mode

mode\_value <- names(table(values))[table(values) == max(table(values))]

# Print the results

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

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

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

b) # Sort the values in ascending order

sorted\_values <- sort(values)

# Find the 2nd highest value

second\_highest <- sorted\_values[length(sorted\_values) - 1]

# Find the 4th lowest value

fourth\_lowest <- sorted\_values[4]

# Print the results

print(paste("2nd Highest Value:", second\_highest))

print(paste("4th Lowest Value:", fourth\_lowest))

53. Write the R CODE for the loop next and break statements with neat flowchart

Input:

# Loop Example with Next and Break Statements

# Initializing a vector

numbers <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

# Looping through the vector

for (num in numbers) {

# Checking if the number is odd

if (num %% 2 == 1) {

# Using the next statement to skip to the next iteration

next

}

# Checking if the number is greater than 6

if (num > 6) {

# Using the break statement to exit the loop

break

}

# Printing the number

print(num)

}

54. a. Write suitable R code to compute the mean, median, mode of the following values c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20,75,70,10,60,70,85,95,55,15)

Input:

# Create the vector of values

values <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20, 75, 70, 10, 60, 70, 85, 95, 55, 15)

# Compute the mean

mean\_value <- mean(values)

# Compute the median

median\_value <- median(values)

# Compute the mode

mode\_value <- as.numeric(names(table(values)[table(values) == max(table(values))]))

# Print the mean, median, and mode

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

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

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

55. Write R code to find 2nd highest and 4th Lowest value of above problem.

Input:

# Create the vector of values

values <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20, 75, 70, 10, 60, 70, 85, 95, 55, 15)

# Sort the values in ascending order

sorted\_values <- sort(values)

# Find the 2nd highest value

second\_highest <- sorted\_values[length(sorted\_values) - 1]

# Find the 4th lowest value

fourth\_lowest <- sorted\_values[4]

# Print the 2nd highest and 4th lowest values

print(paste("2nd highest value:", second\_highest))

print(paste("4th lowest value:", fourth\_lowest))

56. Write R Program To Find The Given Number Is Positive Or Negative

Input:

# Function to check if a number is positive or negative

checkPositiveNegative <- function(num) {

if (num > 0) {

print("The number is positive.")

} else if (num < 0) {

print("The number is negative.")

} else {

print("The number is zero.")

}

}

# Example usage

num <- -5 # Replace with any number you want to check

checkpositiiveNegative(num)

57.Write an R program to create an array with three columns, three rows, and two "tables", taking two

vectors as input to the array. Print the array and perform the following.

a) A+B b) A-B c) t(A)

input:

# Create input vectors

A <- c(1, 2, 3)

B <- c(4, 5, 6)

# Create the array

my\_array <- array(c(A, B), dim = c(3, 3, 2))

# Print the array

print(my\_array)

# Perform A + B

addition\_result <- my\_array[, , 1] + my\_array[, , 2]

print("A + B:")

print(addition\_result)

# Perform A - B

subtraction\_result <- my\_array[, , 1] - my\_array[, , 2]

print("A - B:")

print(subtraction\_result)

# Transpose of A

transpose\_result <- t(my\_array[, , 1])

print("Transpose of A:")

print(transpose\_result)

58.Write a program for creating a pie-chart in R using the input vector (21,62,10,53). Provide

labels for the chart as ‘London’, ‘New York’, ‘Singapore’, ‘Mumbai’. Add a

title to the chart as ‘city pie-chart’ and add a legend at the top right corner of the chart.

Input:

# Install and load the required library

install.packages("plotrix")

library(plotrix)

# Create the input vector

values <- c(21, 62, 10, 53)

# Create labels for the chart

labels <- c('London', 'New York', 'Singapore', 'Mumbai')

# Create the pie chart

pie(values, labels = labels, main = 'City Pie Chart')

# Add a legend at the top right corner

legend("topright", legend = labels, cex = 0.8, fill = rainbow(length(values)))

# Note: The rainbow() function is used to generate different colors for each segment

59.Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and M=c(“mar”,

“apr”, “may”, “jun”, “jul”). Add a title to the chart as “Revenue chart”

Input:

# Create the vectors

H <- c(7, 12, 28, 3, 41)

M <- c("mar", "apr", "may", "jun", "jul")

# Create the bar chart

barplot(H, names.arg = M, main = "Revenue Chart", xlab = "Month", ylab = "Revenue")

60.The table above shows one year of marketing spend and company sales by month.

Month 1 2 3 4 5 6 7 8 9 10 11 12

Spends 1000 4000 5000 4500 3000 4000 9000 11000 15000 12000 7000 3000

Sales 9914 40487 54324 50044 34719 42551 94871 118914 158484 131348 78504 36284

Create a regression model to show the amount of sales(Sales) based on the how much the company spends

(Spends) in advertising. Predict the Sales if Spend=13500.

Input:

# Create the data frame

data <- data.frame(

Month = 1:12,

Spends = c(1000, 4000, 5000, 4500, 3000, 4000, 9000, 11000, 15000, 12000, 7000, 3000),

Sales = c(9914, 40487, 54324, 50044, 34719, 42551, 94871, 118914, 158484, 131348, 78504, 36284)

)

# Create the regression model

model <- lm(Sales ~ Spends, data = data)

# Predict sales when Spends = 13500

new\_data <- data.frame(Spends = 13500)

predicted\_sales <- predict(model, newdata = new\_data)

# Print the predicted sales

print(paste("Predicted Sales when Spends = 13500:", predicted\_sales))

61.Write R program to print the sequence of numbers (1,2,….,10) using repeat loop.

Input:

# Initialize the counter variable

counter <- 1

# Start the repeat loop

repeat {

# Print the current number

print(counter)

# Increment the counter

counter <- counter + 1

# Check the exit condition

if (counter > 10) {

break # Exit the loop if the counter exceeds 10

}

}

62. Create a Data frame as shown below

ID Items Store Price

110 book TRUE 2.5

220 pen FALSE 8.0

330 textbook TRUE 10.0

440 Color pen FALSE 25.0

a. Write a R program to extract specific column from a data frame using column name.

b. Write a R program to extract first two rows from a given data frame

input:

# Create the data frame

df <- data.frame(

ID = c(110, 220, 330, 440),

Items = c("book", "pen", "textbook", "Color pen"),

Store = c(TRUE, FALSE, TRUE, FALSE),

Price = c(2.5, 8.0, 10.0, 25.0)

)

# Extract specific column using column name

column\_name <- "Items"

extracted\_column <- df[, column\_name]

print("Extracted column:")

print(extracted\_column)

# Extract first two rows

extracted\_rows <- df[1:2, ]

print("Extracted rows:")

print(extracted\_rows)

63. [A=matrix(c(2,0,1,3),ncol=2) and B=matrix(c(5,2,4,1),ncol=2) and find the following

using R.

a) Addition of two matrix

b) Subtraction of two matrix

c) Multiplication of A and B

d) Transpose of matrix

e) Modulus of A and B

input:

# Define matrices A and B

A <- matrix(c(2, 0, 1, 3), ncol = 2)

B <- matrix(c(5, 2, 4, 1), ncol = 2)

# Addition of two matrices

addition <- A + B

print("Addition of matrices A and B:")

print(addition)

# Subtraction of two matrices

subtraction <- A - B

print("Subtraction of matrices A and B:")

print(subtraction)

# Multiplication of matrices A and B

multiplication <- A %\*% B

print("Multiplication of matrices A and B:")

print(multiplication)

# Transpose of matrix A

transpose\_A <- t(A)

print("Transpose of matrix A:")

print(transpose\_A)

# Transpose of matrix B

transpose\_B <- t(B)

print("Transpose of matrix B:")

print(transpose\_B)

# Modulus of matrices A and B

modulus\_A <- abs(A)

modulus\_B <- abs(B)

print("Modulus of matrix A:")

print(modulus\_A)

print("Modulus of matrix B:")

print(modulus\_B)

64. Write a program for creating a pie-chart in R using the input vector (21, 62, 10, 53).Provide labels

for the chart as ‘London’, ‘New York’,’Singapore’,’Mumbai’.Add a title to the chart as ‘city piechart’ and add a legend at the top right corner of the chart.

Input:

# Input vector

values <- c(21, 62, 10, 53)

# Labels for the chart

labels <- c("London", "New York", "Singapore", "Mumbai")

# Create the pie chart

pie(values, labels = labels, main = "City Pie Chart")

# Add a legend at the top right corner

legend("topright", legend = labels, fill = rainbow(length(labels)))

65. Randomly Sample the iris dataset such as 80% data for training and 20% for test and create

Logistics regression with train data,use species as target and petals width and

length as feature variables ,Predict the probability of the model using test data, Create

Confusion matrix for above test model

Input:

# Load the iris dataset

data(iris)

# Set the seed for reproducibility

set.seed(123)

# Randomly sample the iris dataset

train\_indices <- sample(1:nrow(iris), 0.8 \* nrow(iris))

train\_data <- iris[train\_indices, ]

test\_data <- iris[-train\_indices, ]

# Create the logistic regression model

model <- glm(Species ~ Petal.Width + Petal.Length, data = train\_data, family = binomial)

# Predict probabilities using the test data

probabilities <- predict(model, newdata = test\_data, type = "response")

# Convert probabilities to predicted classes

predicted\_classes <- ifelse(probabilities > 0.5, "versicolor", "setosa")

# Create the confusion matrix

confusion\_matrix <- table(test\_data$Species, predicted\_classes)

# Print the confusion matrix

print("Confusion Matrix:")

print(confusion\_matrix)

66.. Suppose you track your commute times for two weeks (10 days) and you find the following

times in minutes 17 16 20 24 22 15 21 15 17 22 Enter this into R

a. create function “maxi” to find the longest commute time, the function “avger” to find the

average and the function “mini” to find the minimum.

b. Oops, the 24 was a mistake. It should have been 18. How can you fix this? Do so, and

then find the new average.

c. How many times was your commute 20 minutes or more?

Input:

# Create a vector with commute times

commute\_times <- c(17, 16, 20, 24, 22, 15, 21, 15, 17, 22)

# a) Create functions to find the longest commute time, average, and minimum

maxi <- function(times) {

max(times)

}

avger <- function(times) {

mean(times)

}

mini <- function(times) {

min(times)

}

# Call the functions to find the longest commute time, average, and minimum

longest\_commute <- maxi(commute\_times)

average\_commute <- avger(commute\_times)

minimum\_commute <- mini(commute\_times)

# Print the results

print(paste("Longest commute time:", longest\_commute))

print(paste("Average commute time:", average\_commute))

print(paste("Minimum commute time:", minimum\_commute))

# b) Fix the mistake by replacing the incorrect value

commute\_times[commute\_times == 24] <- 18

# Find the new average after fixing the mistake

new\_average\_commute <- avger(commute\_times)

print(paste("New average commute time:", new\_average\_commute))

# c) Count the number of times the commute was 20 minutes or more

commute\_20\_or\_more <- sum(commute\_times >= 20)

print(paste("Number of times commute was 20 minutes or more:", commute\_20\_or\_more))

67. How to expand the data frame by adding rows and columns in data frame for employee data set.

Input:

# Sample employee data frame

employee\_df <- data.frame(

name = c("John", "Jane", "Mark"),

age = c(25, 30, 28),

salary = c(50000, 60000, 55000)

)

# Adding a new row

new\_row <- c("Alex", 32, 65000)

employee\_df <- rbind(employee\_df, new\_row)

# Adding a new column

new\_column <- c(1, 2, 3, 4)

employee\_df <- cbind(employee\_df, employee\_id = new\_column)

# Print the updated data frame

employee\_df

68. Make a histogram for the “AirPassengers “dataset, start at 100 on the x-axis, and from values 200 to 700, make the bins 150 wide

Input:

# Load the AirPassengers dataset (assuming it's available in the workspace)

data(AirPassengers)

# Create the histogram with specific bin widths and starting point

hist(AirPassengers, breaks = seq(100, 700, by = 150))

69. 4. Suppose you track your commute times for two weeks (10 days) and you find the following

times in minutes 17 16 20 24 22 15 21 15 17 22 Enter this into R

a. create function “maxi” to find the longest commute time, the function “avger” to find the

average and the function “mini” to find the minimum.

b. Oops, the 24 was a mistake. It should have been 18. How can you fix this? Do so, and

then find the new average.

c. How many times was your commute 20 minutes or more?

Input:

# Commute times data

commute\_times <- c(17, 16, 20, 24, 22, 15, 21, 15, 17, 22)

# a. Function to find the longest commute time

maxi <- function(times) {

max(times)

}

# Function to find the average commute time

avger <- function(times) {

mean(times)

}

# Function to find the minimum commute time

mini <- function(times) {

min(times)

}

# Call the functions

longest\_commute <- maxi(commute\_times)

average\_commute <- avger(commute\_times)

minimum\_commute <- mini(commute\_times)

# Print the results

longest\_commute

average\_commute

minimum\_commute

# b. Fix the mistake (24 should be 18) and find the new average

commute\_times[which(commute\_times == 24)] <- 18

new\_average <- avger(commute\_times)

new\_average

# c. Count the number of times the commute was 20 minutes or more

num\_greater\_than\_20 <- sum

70. Randomly Sample the iris dataset such as 80% data for training and 20% for test and create

Logistics regression with train data,use species as target and petals width and

length as feature variables ,Predict the probability of the model using test data, Create

Confusion matrix for above test model

Input:

# Load the iris dataset (assuming it's available in the workspace)

data(iris)

# Set the random seed for reproducibility

set.seed(123)

# Randomly sample 80% of the data for training

train\_indices <- sample(1:nrow(iris), 0.8 \* nrow(iris))

train\_data <- iris[train\_indices, ]

test\_data <- iris[-train\_indices, ]

# Perform logistic regression with training data

library(nnet)

logreg\_model <- multinom(Species ~ Petal.Width + Petal.Length, data = train\_data)

# Predict the probabilities using test data

test\_pred <- predict(logreg\_model, newdata = test\_data, type = "probs")

# Create the confusion matrix

library(caret)

confusion\_matrix <- confusionMatrix(test\_pred, test\_data$Species)

confusion\_matrix

71. Following table contains name and score of the cricket players.

name score

1 Raina 67

2 Bravo 56

3 Dhoni 87

4 Virat 91

a) Create a data frame for the above table.

b) Write a R program to add a new column in a given data frame

c) Find the dimensions and structure of the data frame.

Input:

# Create the data frame

cricket\_df <- data.frame(

name = c("Raina", "Bravo", "Dhoni", "Virat"),

score = c(67, 56, 87, 91)

)

# Print the data frame

cricket\_df

b) # Add a new column

cricket\_df$new\_column <- c(100, 85, 92, 78)

# Print the updated data frame

cricket\_df

c) # Dimensions of the data frame

dimensions <- dim(cricket\_df)

nrows <- dimensions[1]

ncols <- dimensions[2]

# Structure of the data frame

structure <- str(cricket\_df)

# Print the results

Nrows

ncols

structure

72. Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and M=c(“mar”,

“apr”, “may”, “jun”, “jul”). Add a title to the chart as “Remainder chart”.

Input:

# Create the vectors

H <- c(7, 12, 28, 3, 41)

M <- c("mar", "apr", "may", "jun", "jul")

# Create the bar chart

barplot(H, names.arg = M, main = "Remainder chart")

73. Using linear regression analysis establish a relationship between height and weight of a

person using the input vector given below.

# Values of height -> 151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# Values of weight -> 63, 81, 56, 91, 47, 57, 76, 72, 62, 48

Predict the weight of a person with height 170 and Visualize the regression graphically

Input:

# Input vectors for height and weight

height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Perform linear regression

reg\_model <- lm(weight ~ height)

# Predict weight for height 170

new\_height <- 170

predicted\_weight <- predict(reg\_model, newdata = data.frame(height = new\_height))

# Visualize the regression graphically

plot(height, weight, main = "Height vs. Weight", xlab = "Height", ylab = "Weight")

abline(reg\_model, col = "red")

points(new\_height, predicted\_weight, col = "blue", pch = 16)

74. a. Create a 3x4 matrix with 12 random numbers between 1-100; have the matrix be filled our

row-by-row, instead of column-by-column. Name the columns of the matrix uno, dos, tres,

cuatro, and the rows x, y, z. Scale the matrix by 10 and save the result.

b. Extract the column called “uno” as a vector from the original matrix and save the result

c. Extract the row called ‘y’ as a vector from the original matrix and print the sum of the VECTOR

input:

# Create the matrix

matrix <- matrix(runif(12, 1, 100), nrow = 3, ncol = 4, byrow = TRUE,

dimnames = list(c("x", "y", "z"), c("uno", "dos", "tres", "cuatro")))

# Scale the matrix by 10

scaled\_matrix <- matrix \* 10

# Print the scaled matrix

scaled\_matrix

b)uno\_column <- matrix[, "uno"]

uno\_column

c)y\_row <- matrix["y", ]

sum\_y\_row <- sum(y\_row)

sum\_y\_row

75. 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. Find the mean, median, mode using R CODE.

Input:

# Age values

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)

# Mean

mean\_age <- mean(age)

mean\_age

# Median

median\_age <- median(age)

median\_age

# Mode

mode\_age <- names(table(age))[table(age) == max(table(age))]

mode\_age

76. Using linear regression analysis establish a relationship between height and weight of a person

using the input vector given below.

# values of height -> 151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# values of weight. -> 63, 81, 56, 91, 47, 57, 76, 72, 62, 48

Predict the weight of a person with height 170. Visualize the regression graphically

Input:

# Input vectors for height and weight

height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Perform linear regression

reg\_model <- lm(weight ~ height)

# Predict weight for height 170

new\_height <- 170

predicted\_weight <- predict(reg\_model, newdata = data.frame(height = new\_height))

# Visualize the regression graphically

plot(height, weight, main = "Height vs. Weight", xlab = "Height", ylab = "Weight")

abline(reg\_model, col = "red")

points(new\_height, predicted\_weight, col = "blue", pch = 16)

77. Explore the USArrests dataset, contains the number of arrests for murder, assault, and rape

for each of the 50 states in 1973. It also contains the percentage of people in the state who

live in an urban area.

(i) a. Explore the summary of Data set, like number of Features and its type. Find the number

of records for each feature. Print the statistical feature of data

b. Print the state which saw the largest total number of rape

c. Print the states with the max & min crime rates for murder

(ii). a. Find the correlation among the features

b. Print the states which have assault arrests more than median of the country

c. Print the states are in the bottom 25% of murder

input:

# Load the USArrests dataset (assuming it's available in the workspace)

data("USArrests")

# (i) a. Summary of dataset

summary(USArrests)

# b. State with the largest total number of rape

state\_max\_rape <- names(USArrests$Rape)[which.max(USArrests$Rape)]

state\_max\_rape

# c. States with the max and min crime rates for murder

state\_max\_murder <- names(USArrests$Murder)[which.max(USArrests$Murder)]

state\_min\_murder <- names(USArrests$Murder)[which.min(USArrests$Murder)]

state\_max\_murder

state\_min\_murder

# (ii) a. Correlation among the features

correlation <- cor(USArrests)

correlation

# b. States with assault arrests more than median

states\_above\_median <- names(USArrests$Assault)[USArrests$Assault > median(USArrests$Assault)]

states\_above\_median

# c. States in the bottom 25% of murder

states\_bottom\_25 <- names(USArrests$Murder)[USArrests$Murder < quantile(USArrests$Murder, 0.25)]

states\_bottom\_25

78. Write a R program to generate the Fibonacci Series

Input:

# Function to generate the Fibonacci Series

fibonacci <- function(n) {

sequence <- c(1, 1)

if (n <= 2) {

return(sequence[1:n])

} else {

for (i in 3:n) {

sequence[i] <- sequence[i-1] + sequence[i-2]

}

return(sequence)

}

}

# Generate Fibonacci Series up to n terms

n <- 10

fib\_series

79. A SIMPLE R CODE FOR How break control statement works. Draw the flowchart and write the code for the below output

using R

[1] "WELCOME" "CSE"

[1] "WELCOME" "CSE"

[1] "WELCOME" "CSE"

[1] "WELCOME" "CSE"

Input:

# Using a for loop with break statement

for (i in 1:10) {

print(c("WELCOME", "CSE"))

if (i == 4) {

break

}

}

80. Write a program for creating a Histogram in R using the input

vector(vc(9,13,21,8,36,22,12,41,31,33,19).

Use respective parameters such as xlim,ylim, xlab,ylab and main

Input:

# Input vector

v <- c(9, 13, 21, 8, 36, 22, 12, 41, 31, 33, 19)

# Create Histogram

hist(v, xlim = c(0, 50), ylim = c(0, 5), xlab = "Values", ylab = "Frequency", main = "Histogram")

81. a. Write a program for creating a pie-chart in R using the input vector (21,62,10,53). Provide labels for

the chart as ‘London’, ‘New York’, ‘Singapore’, ‘Mumbai’. Add a

title to the chart as ‘city pie-chart’ and add a legend at the top right corner of the chart.

b. Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and M=c(“mar”, “apr”,

“may”, “jun”, “jul”). Add a title to the chart as “Revenue chart”

Input:

# Input vector

values <- c(21, 62, 10, 53)

labels <- c("London", "New York", "Singapore", "Mumbai")

# Create pie chart

pie(values, labels = labels, main = "City Pie Chart")

legend("topright", legend = labels, cex = 0.8, fill = 1:length(labels))

82. The table above shows one year of marketing spend and company sales by month.

Month 1 2 3 4 5 6 7 8 9 10 11 12

Spends 1000 4000 5000 4500 3000 4000 9000 11000 15000 12000 7000 3000

Sales 9914 40487 54324 50044 34719 42551 94871 118914 158484 131348 78504 36284

Create a regression model to show the amount of sales(Sales) based on the how much the company spends

(Spends) in advertising. Predict the Sales if Spend=13500

Input:

# Create data frame

data <- data.frame(

Month = 1:12,

Spends = c(1000, 4000, 5000, 4500, 3000, 4000, 9000, 11000, 15000, 12000, 7000, 3000),

Sales = c(9914, 40487, 54324, 50044, 34719, 42551, 94871, 118914, 158484, 131348, 78504, 36284)

)

# Create regression model

reg\_model <- lm(Sales ~ Spends, data = data)

# Predict Sales for Spend = 13500

new\_data <- data.frame(Spends = 13500)

predicted\_sales <- predict(reg\_model, newdata = new\_data)

predicted\_sales