**Program 1: Basic Arithmetic Operations in R**

**Aim**:  
To perform basic arithmetic operations (Addition, Subtraction, Multiplication, and Division) in R.

**Procedure**:

1. Define two numeric variables.
2. Use arithmetic operators +, -, \*, /.
3. Print the results using print() or cat().

**Program**:

a <- 10

b <- 7

sum <- a + b

sub <- a - b

mul <- a \* b

div <- a / b

print(sum)

print(sub)

print(mul)

print(div)

**Output**:

[1] 17

[1] 3

[1] 70

[1] 1.428571

**Result**:  
Basic arithmetic operations were successfully performed.

**Program 2: Check Even or Odd**

**Aim**:  
To check whether a given number is even or odd.

**Procedure**:

1. Assign a value to a variable.
2. Use modulus operator (%%) to check divisibility by 2.
3. Use if...else to print result.

**Program**:

num <- 7

if (num %% 2 == 0) {

cat(num, "is Even")

} else {

cat(num, "is Odd")

}

**Output**:

7 is Odd

**Result**:  
The program correctly identified the number as odd.

**Program 3: Fibonacci Sequence Using Loop**

**Aim**:  
To generate the Fibonacci sequence using a loop.

**Procedure**:

1. Initialize two variables a and b.
2. Use for loop to print the first n Fibonacci numbers.
3. Use temporary variable to update values.

**Program**:

n <- 10

a <- 0

b <- 1

cat("Fibonacci Sequence:\n")

for (i in 1:n){

cat(a, "")

temp <- a + b

a <- b

b <- temp

}

**Output**:

Fibonacci Sequence:

0 1 1 2 3 5 8 13 21 34

**Result**:  
The Fibonacci sequence was generated successfully.

**Program 4: Factorial Using Loop**

**Aim**:  
To calculate the factorial of a number using a for loop.

**Procedure**:

1. Set a number.
2. Use for loop to multiply numbers from 1 to n.
3. Print the factorial.

**Program**:

num <- 5

factorial <- 1

for(i in 1:num){

factorial <- factorial \* i

}

cat("Factorial of", num, "is", factorial)

**Output**:

Factorial of 5 is 120

**Result**:  
The factorial of a number was calculated using a loop.

**Program 5: Create and Print a Vector**

**Aim**:  
To create a vector and perform basic operations.

**Procedure**:

1. Use c() to create a vector.
2. Access and modify elements using indexing.
3. Use sort() and length() functions.

**Program**:

techstack <- c("HTML", "CSS", "JavaScript", "PHP")

print(techstack)

# Access elements

print(techstack[2])

# Modify an element

techstack[4] <- "React"

print(techstack)

# Length

print(length(techstack))

**Output**:

[1] "HTML" "CSS" "JavaScript" "PHP"

[1] "CSS"

[1] "HTML" "CSS" "JavaScript" "React"

[1] 4

**Result**:  
Vector operations were successfully performed.

**Program 6: Create and Access a Matrix**

**Aim**:  
To create and access elements of a matrix.

**Procedure**:

1. Use matrix() to create a matrix.
2. Access elements using row and column indices.
3. Loop through matrix using for loops.

**Program**:

m <- matrix(1:9, nrow = 3, ncol = 3)

print(m)

for (i in 1:nrow(m)) {

for (j in 1:ncol(m)) {

cat("Element at [", i, ",", j, "] is:", m[i, j], "\n")

}

}

**Output**:

Element at [ 1 , 1 ] is: 1

Element at [ 1 , 2 ] is: 4

...

Element at [ 3 , 3 ] is: 9

**Result**:  
Matrix was created and elements were accessed using nested loops.

**Program 7: Data Frame and Grade Calculation**

**Aim**:  
To create a data frame and assign grades based on scores.

**Procedure**:

1. Create a data frame using data.frame().
2. Loop through each row to assign grades.
3. Use if...else for conditions.

**Program**:

df <- data.frame(

Name = c("Alice", "Bob", "Charlie", "David"),

Score = c(85, 72, 90, 65)

)

df$Grade <- ""

for (i in 1:nrow(df)) {

if (df$Score[i] >= 90) {

df$Grade[i] <- "A"

} else if (df$Score[i] >= 80) {

df$Grade[i] <- "B"

} else if (df$Score[i] >= 70) {

df$Grade[i] <- "C"

} else {

df$Grade[i] <- "D"

}

}

print(df)

**Output**:

Name Score Grade

1 Alice 85 B

2 Bob 72 C

3 Charlie 90 A

4 David 65 D

**Result**:  
Data frame was created and grades were calculated based on scores.

**Program 8: Survey Responses using Factor**

**Aim**:  
To analyze categorical data using factors.

**Procedure**:

1. Create a character vector.
2. Convert to factor using factor().
3. Summarize using summary().

**Program**:

responses <- c("Phone", "Laptop", "Tablet", "Phone", "Laptop", "Phone")

device\_factor <- factor(responses)

summary(device\_factor)

**Output**:

Laptop Phone Tablet

2 3 1

**Result**:  
Categorical data was successfully summarized using factors.

**Program 9: Sorting and Searching in Vectors**

**Aim**:  
To sort and search elements in a vector.

**Procedure**:

1. Create a numeric vector.
2. Sort using sort().
3. Use loop and if to search value.

**Program**:

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

sorted\_vec <- sort(vec)

print(sorted\_vec)

target <- 30

i <- 1

found <- FALSE

while (i <= length(vec)) {

if (vec[i] == target) {

cat("Value found at index:", i, "\n")

found <- TRUE

}

i <- i + 1

}

if (!found) {

cat("Value not found\n")

}

**Output**:

[1] 10 20 30 30 40 50

Value found at index: 3

Value found at index: 5

**Result**:  
Sorting and searching operations on a vector were successful.

**Program 10: Bar Plot of Monthly Expenses**

**Aim**:  
To create a bar chart to visualize monthly expenses.

**Procedure**:

1. Define a numeric vector of expenses.
2. Use barplot() to visualize.
3. Set names.arg, col, and labels.

**Program**:

expenses <- c(1200, 1100, 980, 1800, 180, 1400)

months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun")

**Output**: *(Graphical Output - Bar Plot)*

**Result**:  
Bar chart was created to display monthly expenses visually.

**Program 11: Importing and Analyzing mtcars dataset**

**Aim:** To import mtcars dataset and perform various operations using R language

**Procedure**

Step1) Install R and R Studio IDE in the System

Step2) Create a new script file using the command File🡪New File🡪R Script or Ctrl +Shift+N.

Step3) Install ggplot2 package and load mtcars dataset .

Step4) Perform various descriptive statistical operations using functions.

Step5) Save this file using File🡪Save or Ctrl+S

Step5) Select ALL and run the script file using Ctrl+Enter

Step6) Close the R Script file.

**Program coding**

**install.packages("ggplot2")**

**library("ggplot2")**

**data(mtcars)**

**mtcars**

#To summarize each variable in dataset

**summary(mtcars)**

#Structure of dataset

**str(mtcars)**

#dimensions in terms of number of rows and columns

**dim(mtcars)**

#get all quartiles of mpg variable

**res<-quantile(mtcars$mpg)**

**print(res)**

#get maximum mpg

**print(max(mtcars$mpg))**

#count number of observations

**print(nrow(mtcars))**

#count number of variables

**print(ncol(mtcars))**

#first 6 observations

**head(mtcars)**

#last 6 observations

**tail(mtcars)**

#cars with cylinder>6 and mpg>15.5

**data<-subset(mtcars,cyl>6 & mpg>15.5)**

**print(data)**

**write.csv(data,"P:\\mtcarsdata.csv",row.names=TRUE)**

**print(“The observations with cylinder greater than 6 and mileage greater than 15.5 is”)**

**read.csv("P:\\mtcarsdata.csv")**

**Result :**

Thus mtcars dataset has been loaded and analysed using R language successfully

**Program 12: Analyzing and Visualizing mtcars dataset**

**Aim:** To import mtcars dataset and plot the values in histogram using R language

**Procedure**

Step1) Install R and R Studio IDE in the System

Step2) Create a new script file using the command File🡪New File🡪R Script or Ctrl +Shift+N.

Step3) Install ggplot2 package and load mtcars dataset .

Step4) Perform various descriptive statistical operations using functions.

Step5) Plot the variable values in histogram

Step6) Save this file using File🡪Save or Ctrl+S

Step7) Select ALL and run the script file using Ctrl+Enter

Step8) Close the R Script file.

**Program coding**

**install.packages("ggplot2")**

**library("ggplot2")**

**data(mtcars)**

**mtcars**

#To summarize each variable in dataset

**summary(mtcars)**

#Structure of dataset

**str(mtcars)**

#dimensions in terms of number of rows and columns

**dim(mtcars)**

#get all quartiles of mpg variable

**res<-quantile(mtcars$mpg)**

**print(res)**

#get maximum mpg

**print(max(mtcars$mpg))**

#count number of observations

**print(nrow(mtcars))**

#count number of variables

**print(ncol(mtcars))**

#first 6 observations

**head(mtcars)**

#last 6 observations

**tail(mtcars)**

**#create histogram of mpg values**

**hist(mtcars$mpg,**

**col=”steelblue”,**

**main=”Histogram of Mileage vaues”,**

**xlab=”mpg”,**

**ylab=”Frequency”)**

**write.csv(data,"P:\\mtcarsdata.csv",row.names=TRUE)**

**print(“The observations with cylinder greater than 6 and mileage greater than 15.5 is”)**

**read.csv("P:\\mtcarsdata.csv")**

**Result :**

Thus mtcars dataset has been loaded, analysed and visualized in histogram using R language successfully

**Ex.12 Importing and Analyzing IRIS dataset**

**Aim:** To import IRIS dataset and perform various operations using R language

**Procedure**

Step1) Install R and R Studio IDE in the System

Step2) Create a new script file using the command File🡪New File🡪R Script or Ctrl +Shift+N.

Step3) Install ggplot2 package and load IRIS dataset .

Step4) Perform various descriptive statistical operations using functions.

Step5) Save this file using File🡪Save or Ctrl+S

Step5) Select ALL and run the script file using Ctrl+Enter

Step6) Close the R Script file.

**Program coding**

**#install.packages("ggplot2")**

**#library("ggplot2")**

**data(iris)**

**iris**

**summary(iris)**

**str(iris)**

**print(nrow(iris))**

**#count number of variables**

**print(ncol(iris))**

**#first 6 observations**

**head(iris)**

**#last 6 observations**

**tail(iris)**

**#extract only setosa species**

**res<-subset(iris,Species=="setosa")**

**print(res)**

**#extract only sepal length and find sum,mean,median,min and max values**

**mysepal = iris$Sepal.Length**

**sum(mysepal)**

**mean(mysepal)**

**median(mysepal)**

**min(mysepal)**

**max(mysepal)**

**attach(iris)**

**plot(Sepal.Length, Sepal.Width)**

**plot(Sepal.Length, Sepal.Width, ylim=c(0,5), xlim=c(3,9))**

**Result :**

Thus IRIS dataset has been loaded and various operations performed using R language successfully

**Ex.11 Performing Statistical testing t-test and Correlation on two sample variables of mtcars dataset**

**Aim:** To perform students t-test and finding correlation between two sample variables using R language

**Procedure**

Step1) Install R and R Studio IDE in the System

Step2) Create a new script file using the command File🡪New File🡪R Script or Ctrl +Shift+N.

Step3) Install ggplot2 package and load mtcars dataset .

Step4) Perform statistical t-test and finding correlation using functions.

Step5) Plot the result in boxplot.

Step6) Save this file using File🡪Save or Ctrl+S

Step7) Select ALL and run the script file using Ctrl+Enter

Step8) Close the R Script file.

**Program coding**

# View the first few rows of the dataset

**head(mtcars)**

# Separate the data into two groups (e.g., automatic vs. manual transmission)

**group\_auto <- mtcars$mpg[mtcars$am==0]**

**group\_manual <- mtcars$mpg[mtcars$am==1]**

# Perform a two-sample t-test

**result <- t.test(group\_auto, group\_manual)**

# Print the result

**print(result)**

#finding correlation between mileage per gallon and weight

**cor.test(mtcars$mpg,mtcars$wt)**

**means <- c(mean(group\_auto), mean(group\_manual))**

**names <- c("Automatic", "Manual")**

# Create a boxplot

**boxplot(group\_auto, group\_manual,**

**names = c("Automatic", "Manual"),**

**main = "Miles per Gallon (mpg) by Transmission Type",**

**ylab = "Miles per Gallon (mpg)")**

**Result :**

Thus mtcars dataset has been loaded and performed statistical t-test ,correlation using R language successfully