**Experiment Name: To perform the basic mathematical operations in R programming.**

**R Script:-**

#Assignment

s <- 10

print(s)

#case sensitive

a <- 5

b <- 2

c <- 1

print(a+B+c)

#Arithmetic Operations

#creation of vector

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

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

print(a)

print(b)

#Addition of two vector

Result <- a+b

print(Result)

#Subtraction of two vector

Result <- a-b

print(Result)

#Multiplication of two vector

Result <- a\*b

print(Result)

#Division of two vector

Result <- a/b

print(Result)

#options()

1/7#by default it will show 7 digits output.

options(digits = 3)#by using this it will show only 3 digits after decimal point

1/7

#Miscellaneous Mathematical functions

x<-20

abs(x) #Absolute Value

sqrt(x) #square root

exp(x) #exponential transformation

log(x) #logarithmic transformation

cos(x) #cosine and other trigonometric transformation

#infinite and Nan Number

y<-5

z<-6

ls() #List all object

exists("y") #identify R object with 'y' name

rm(y) #remove object.

rm(y,z) #remove multiple object.

rm(list=ls()) #remove everything on working environment.

**Write program for creating and manipulating R objects in R-Vectors, Matrices, Array, Dataframes, List.**

**Atomic Vector:-**

**Numeric vector: -**

num\_vec<-c(10.1, 10.2, 33.2)

num\_vec

**Integer vector: -**

num<-c(2L,6L,4L,9L)

num

**Character vector: -**

fruits<-c("Mango","apple","papaya")

print(fruits)

**Logical vector: -**

a<-as.integer(20)

b<-as.integer(10)

log\_vec<-c(a<b,b<a,a>b,b>a)

log\_vec

**Operations on Vector: -**

1)combining vectors:

data\_vec<-c(names,num)

data\_vec

**2)Arithmetic operations:**

a<-c(1,3,5,7)

b<-c(2,4,6,8)

a+b

a-b

a\*b

a/b

**3)Logical Index vector:**

z<-c(1,2,3,4,5,6)

z[c(TRUE,FALSE,TRUE,TRUE,FALSE,TRUE)]

**4)Numeric Index: -**

q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")

q[2]

q[-4]

q[15]

**5)Duplicate Index: -**

q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")

q[c(2,4,4,3)]

**6)Range Indexes: -**

q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")

b<-q[2:5]

b

**7)out-of-order Indexes: -**

q<-c("shubham","arpita","nishka","gunjan","vaishali","sumit")

q[c(2,1,3,4,5,6)]

**8)Named vectors members: -**

z=c("Roshani","Kawale")

z

names(z)=c("FirstName","LastName")

z

z["FirstName"]

**#creation of matrix:-**

P <- matrix(c(5:16), nrow = 4, byrow = TRUE)

print(P)

Q <- matrix(c(3:14), nrow = 4, byrow = FALSE)

print(Q)

**#operations on Matrix:-**

**#1)Addition:-**

sum<-P+Q

print(sum)

**#2)Subtraction:-**

sub<-P-Q

print(sub)

**#3)Multiplication(\*):-**

mult<-P\*Q

print(mult)

**#4)Multiplication(by constant):-**

mult<-P\*5

print(mult)

**#5)Division:-**

div<-P/Q

div

**#creation of Arrays:-**

vec1 <-c(1,3,5)

vec2 <-c(10,11,12,13,14,15)

res <- array(c(vec1,vec2),dim=c(3,3,2))

print(res)

**#Naming Of Arrays**

col\_names <- c("Col1","Col2","Col3")

row\_names <- c("Row1","Row2","Row3")

matrix\_names <- c("Matrix1","Matrix2")

res <- array(c(vec1,vec2),dim=c(3,3,2),dimnames=list(row\_names,col\_names,matrix\_names))

print(res)

**#Creation Of DataFrame**

stud.data<- data.frame(

student\_id = c (1:5),

student\_name = c("Shubham","Arpita","Nishka","Gunjan","Sumit"),

class = c("MBA","MCA","MBA","IMCA","MCA"),

roll\_no=c(20,45,78,12,50)

)

print(stud.data)

**#Operations on DataFrame:-**

**1)Extracting specific columns from a data frame**

final <- data.frame(stud.data$student\_id,stud.data$class)

print(final)

**2)Modification of DataFrame:**

**#Adding row in the data frame**

x <- list(6,"Vaishali","IMCA",15)

rbind(stud.data,x)

**#Adding column in the data frame**

y <- c("Moradabad","Lucknow","Etah","Sambhal","Khurja")

cbind(stud.data,city=y)

**#Delete rows from data frame**

stud.data<-stud.data[-1,]

print(stud.data)

**#Delete column from the data frame**

stud.data$roll\_no<-NULL

print(stud.data)

**#creation of List:-**

list\_1<-list("Shubham","Arpita","Vaishali")

list\_1

list\_data<-list("Shubham","Arpita",c(1,2,3,4,5),TRUE,FALSE,22.5,12L)

print(list\_data)

**#Operation on lists:-**

**1)Giving name to list:-**

list\_data <- list(c("Shubham","Nishka","Gunjan"), matrix(c(40,80,60,70,90,80), nrow = 2),

list("BCA","MCA","B.tech"))

names(list\_data) <- c("Students", "Marks", "Course")

list\_data

**2)Accessing elements using index:-**

print(list\_data[1])

**3)Accessing elements using names:-**

print(list\_data["Students"])

print(list\_data$Marks)

**4)Merging Lists:-**

Even\_list <- list(2,4,6)

Odd\_list <- list(1,3,5)

# Merging the two lists.

merged.list <- list(Even\_list,Odd\_list)

print(merged.list)

**Write program to demonstrate Loops & Vectorization of Missing Values.**

**#using for loop**

week<-c('Sunday',

'Monday',

'Tuesday',

'Wednesday',

'Thursday',

'Friday',

'saturday')

for(day in week)

{

print(day)

}

**#using while loop**

val=1

while(val<=5)

{

print(val)

val=val+1

}

**#using repeat loop**

val=1

repeat

{

print(val)

val=val+1

if(val>5)

{

break

}

}

**Demonstrate Importing and exporting data.**

**#IMPORT**

getwd()

#Importing csv file.

path<-"C:/Users/Leena/OneDrive/Documents/candidate-elimination.csv"

content<-read.csv(path)

print(content)

#Importing Text file.

x<-read.table("C:/Users/Leena/OneDrive/Documents/file.txt",header=FALSE)

print(x)

#Importing CSV file using csv2.

x<-read.csv2("C:/Users/Leena/OneDrive/Documents/candidate-elimination.csv")

print(x)

**# EXPORT**

**1).Export a data frame to a text file using write.table().**

df=data.frame(

"Name"=c("Leena","Roshani","Komal"),

"Language"=c("R","Python","Java"),

"Age"=c(22,25,24)

)

write.table(df,

file="Demo.txt",

sep = "\t",

row.names = TRUE,

col.names = NA)

2).**Exporting Data to a csv file.**

df=data.frame(

"Name"=c("Ankit","Manthan","Pranav"),

"Language"=c("C Programming","Java","HTML"),

"Age"=c(28,27,29)

)

write.table(df,

file="myFile.csv",

sep = "\t",

row.names = FALSE)

**3)Exporting data to a csv2 file**

library(readr)

df2=data.frame(

"Name"=c("Swati","Anushka","Ashish","Kalpesh"),

"Language"=c("R","Python","Java","PHP"),

"Age"=c(22,25,45,23),

"class"=c("SYMCA","SYMCA","FYMCA","FYMCA")

)

write\_csv(df2,path="Demo2.csv")

write\_csv2(df2,path="Demo3.csv")

**4)Exporting data using write\_tsv()function**

getwd()

#library(readr)

df2=data.frame(

"Name"=c("Swati","Anushka","Ashish","Kalpesh"),

"Language"=c("R","Python","Java","PHP"),

"Age"=c(22,25,45,23),

"class"=c("SYMCA","SYMCA","FYMCA","FYMCA")

)

write\_tsv(df2,path="pract5.txt")

**Write program for validating and exploring data manipulation (Summarizing, Sorting, Subsetting, Merging, Joining).**

**1)Summarizing:-**

#create a data frame

data1<-data.frame(player=c('A','B','c','D','E'),

runs=c(100,200,105,50,90),

wickets=c(15,20,8,5,8)

)

data1

#summarize method

summarize(data1,sum(runs),mean(runs),mode(wickets))

//summarize(data1)

------------------------------------------------------------------------------------------------------------------------------

**2)Sorting:-**

#creating data frame

dataBook=data.frame(Customers=c("Ruhi","James","Heera","Shubham","Joe","Priya"),

Products=c("ProdA","ProdB","ProdC","ProdD","ProdE","prodF"),

Salary=c(500,600,450,700,300,400))

dataBook

#sorting the data frame in ascending order

arrange(dataBook,Salary)

#sorting the data frame in descending order

dataBook%>%arrange(desc(Salary))

------------------------------------------------------------------------------------------------------------------------------

**3)Subsetting:-**

#Subsetting in R using []operator:

#create vector

x<-1:15

cat("Original vector:",x,"\n")

#subsetting vector:

cat("First 5 values of vector:",x[1:5],"\n")

cat("Without values present at index 1,2and 3",x[-c(1,2,3),"\n"])

#Subsetting in R using [[]]operator:

#create list:

ls<-list(a=1,b=2,c=10,d=20)

cat("Original List:\n")

print(ls)

#select first element of list:

cat("Element of list:",ls[[3]],"\n")

#Subsetting using c() function:

ls2<-list(a=list(x=1,y="students"),b=1:10)

ls2

cat("Using c() function:\n")

//print(ls2[[c(1,2)]])

//print(ls2[[1]][[2]])

#Subsetting Using $ operator:

ls3<-list(a="Roshani",b=1,c="Hello")

ls3

cat("Using $ operator:\n")

print(ls3$a)

---------------------------------------------------------------------------------------------------------------

**4)Merging: -**

#Merge DataFrames by Row Names:-

data\_frame1<-data.frame(No=c(1:5),

Name=letters[1:5],

Salary=c(200,200,300,NA,300)

)

data\_frame1

data\_frame2<-data.frame(No=c(6:8),

Name=letters[8:10],

Salary=c(400,350,NA)

)

data\_frame2

data\_frame\_merge<-merge(data\_frame1,data\_frame2,by='row.names',all=TRUE)

print("Merged DataFrame")

print(data\_frame\_merge)

----------------------------------------------------------------------------------------------------------------

**5)Joining:-**

#Using Inner join:-

data1<-data.frame(ID=c(1:5))

data2<-data.frame(ID=c(4:8))

inner\_join(data1,data2,by="ID")

#Using Left join:-

data1<-data.frame(ID=c(1:5),

Name=c("Rutuja","Lokesh","Ram","Purvi","Nita"))

data2<-data.frame(ID=c(4:8),

Marks=c(70,85,80,90,75))

left\_join(data1,data2,by="ID")

**#Validating data:-**

data(cars)

head(cars, 3)

library(validate)

rules <- validator(speed >= 0,

dist >= 0,

speed/dist <= 1.5,

cor(speed, dist)>=0.2)

out <- confront(cars, rules)

summary(out)

**Write program to implement the following analysis techniques using R.**

**1.Statistical hypothesis generation and testing**

**2. Chi-Square test**

**3. t-Test**

**4. Correlation analysis**

**1)Stastical hypothesis testing:-**

#One-sample T-testing:

x<-rnorm(100)#sample vector

t.test(x,mu=5)#one-sample t-test

#two-sample T-testing:

x<-rnorm(100)

y<-rnorm(100)

t.test(x,y)

#Directional Hypothesis:-

t.test(x,mu=2,alternative = 'greater')

#one sample u-test:-

wilcox.test(y,exact = FALSE)

#Two sample u-test:-

wilcox.test(x,y)

**2)Correlation Test:-**

cor.test(matcars$mpg,matcars$hp)

**3)Chi-Square Test:-**

library(MASS)

#create DataFrame:

print(str(survey))

# Create a data frame from the main data set.

stu\_data = data.frame(survey$Smoke,survey$Exer)

# Create a contingency table with the needed variables.

stu\_data = table(survey$Smoke,survey$Exer)

print(stu\_data)

**Write program to implement the following analysis techniques using R.**

**Regression analysis**

**# R program to illustrate**

**# Linear Regression**

**# Height vector**

x <- c(153, 169, 140, 186, 128,

       136, 178, 163, 152, 133)

**# Weight vector**

y <- c(64, 81, 58, 91, 47, 57,

       75, 72, 62, 49)

**# Create a linear regression model**

model <- lm(formula = y~x)

**# Print regression model**

print(model)

**# Find the weight of a person With height 182**

df <- data.frame(x = 182)

res <-  predict(model, df)

cat("\nPredicted value of a person

               with height = 182")

print(res)

**# Output to be present as PNG file**

png(file = "linearRegGFG.png")

**# Plot**

plot(x, y, main = "Height vs Weight Regression model")

abline(lm(y~x))

**# Save the file.**

dev.off()

**# R program to illustrate**

**# Multiple Linear Regression**

**# Using airquality dataset**

input <- airquality[1:50,c("Ozone", "Wind", "Temp")]

**# Create regression model**

model <- lm(Ozone~Wind + Temp,data = input)

**# Print the regression model**

cat("Regression model:\n")

print(model)

**# Output to be present as PNG file**

png(file = "multipleRegGFG.png")

**# Plot**

plot(model)

**# Save the file.**

dev.off()

**# R program to illustrate**

**# Logistic Regression**

**# Using mtcars dataset**

**# To create the logistic model**

model <- glm(formula = vs ~ wt,family = binomial,data = mtcars)

**# Creating a range of wt values**

x <- seq(min(mtcars$wt),max(mtcars$wt),0.01)

**# Predict using weight**

y <- predict(model, list(wt = x), type = "response")

**# Print model**

print(model)

**# Output to be present as PNG file**

png(file = "LogRegGFG.png")

**# Plot**

plot(mtcars$wt, mtcars$vs, pch = 16, xlab = "Weight", ylab = "VS")

lines(x, y)

**# Saving the file**

dev.off()

**Regression Analysis**

library(ggplot2)

library(broom)

library(dplyr)

library(ggpubr)

#Linear Regression

x<-c(153,169,140,186,128,136,178,163,152,133)

y<-c(64,81,58,91,47,57,75,72,62,49)

model<-lm(formula = y~x)

print(model)

df<-data.frame(x=182)

res<-predict(model,df)

cat("\n Predicted value of a person with Height = 182")

print(res)

png(file = "LinearRegression.png")

plot(x,y,main = "Height vs Weight Regression model")

abline(lm(y~x))

dev.off()

#Multiple Linear Regression

input<-airquality[1:50,c("Ozone","Wind","Temp")]

model<-lm(Ozone~Wind + Temp,data = input)

cat("Regression Model:\n")

print(model)

png(file="multipleRegGFG.png")

plot(model)

dev.off()

#Logistic Regression

model<-glm(formula = vs~wt,family = binomial,data=mtcars)

x<-seq(min(mtcars$wt),max(mtcars$wt),0.01)

y<-predict(model,list(wt=x),type = "response")

print(model)

png(file="Rplot%03d.png")

plot(mtcars$wt,mtcars$vs,pch=16,xlab="Weight",ylab="VS")

lines(x,y)

dev.off()