Introduction to R programming and Data acquisition Install packages , Loading packages Data types,  checking type of variable, printing variable and objects (Vector, Matrix, List, Factor, Data frame, Table)  cbind-ing and rbind-ing, **Reading and Writing data.** setwd(), getwd(), data(), rm(), Attaching and Detaching  data. Reading data from the consol. Loading data from different data sources. (CSV, Excel).

**Q.1 Installation and setting up R**

\*To explain the process of installation and screen shots to be put.

**Q.2 Data types in R Programming.**

**1. R Program to illustrate Numeric data type**

# Assign a decimal value to variable x

x = 5.6

# print the class name of variable x

print(class(x))

# print the type of variable x

print(typeof(x))

# print the value of variable x

print (x)

**Output:**

[1] "numeric"

[1] "double"

[1] 5.6

• **Even if an integer is assigned to a variable y, it is still being saved as a numeric value.** # Assign an integer value to variable y

y = 5

# print the class name of variable y

print(class(y))

# print the type of variable y

print(typeof(y))

**Output:**

[1] "numeric"

[1] "double"

**2. R program to illustrate integer data type** # Create an integer variable

x = as.integer(5)

# print the class name of variable x

print(class(x))

# print the type of variable x

print(typeof(x))

# Declare an integer by appending ‘L’ as suffix. y = 5L

# print the class name of y

print(class(y))

# print the type of y

print(type of(y))

Output:

[1] "integer"

[1] "integer"

[1] "integer"

[1] "integer"

**3. R program to illustrate logical data type** # Two variables

x = 4

y = 3

# Comparing two values

z = x > y

# print the logical value

print(z)

# print the class name of z

print(class(z))

# print the type of z

print(type of(z))

**Output:**

[1] TRUE

[1] "logical"

[1] "logical"

**4. R program to illustrate complex data type** # Assign a complex value to variable x x = 4 + 3i

# print the class name of variable x

print(class(x))

# print the type of variable x

print(type of(x))

**Output:**

[1] "complex"

[1] "complex"

**5. R program to illustrate character data type**

# Assign a character value to char

char = "Mumbai University"

# print the class name of char

print(class(char))

# print the type of char

print(type of(char))

**Output:**

[1] "character"

[1] "character"

**Q.3 Reading and Writing data to and from R.**

**Create a file myrecord.txt and put data of your choice**

Welcome to my text file

Extracted data for text file

record\_data <- read.table("D:/myrfile/myrecord.txt")

head(record\_data)

Similarly, **read.csv()** function can be used to read data from csv files.

record\_data <- read.csv("D:/myrfile/data.csv")

head(record\_data)

Similarly, **read\_excel ()** function can be used to read data from xls and xlsx files(Excel files). install.packages("readxl") #install the package for excel file

library(readxl)

data <- read\_excel("D:/myrfile/data-1.xls", sheet = 1)

print(data)

• **Writing Data to a File**

the setwd() function to assign working directory.

setwd("D:/myrfile ")

To check your current working directory, type

getwd()

**In R, we can write data easily to a file, using the write.table() command.**

x <- data.frame (name = c("John", "Alice", "Bob"), department = c("Sales", "Marketing",  "Finance"))

write.table(x, file ="data.csv", sep = ",")

z <- data.frame(a = 10, b = 40, c = pi)

write.csv(z, file = "sample.csv")

**Q.4 Packages in R programming.**

**Check Available R Packages**

Get library locations containing R packages.

.libPaths()

When we execute the above code, it will produce the following result.

**Get the list of all the packages installed**

library()

**R provides search() function to get all packages currently loaded in the R environment.** search()

• **install.packages("Package Name")**

• **Install the package named "XML".**

**install.packages("XML")**

**Q.5 Demonstrate Rstudio to insert data from a data frame into a mysql table. Open mysql command line**

**Create database prac5;**

**Create table emp1(name varchar(50),age int,city varchar(50));**

**Installing and Loading Required Package:**

install.packages("RMySQL")

library(RMySQL)

**Establishing a Database Connection**

con <- dbConnect(MySQL(), user='root', password='howareyou', dbname='prac5', host='localhost',port=3306)

**Creating a Data Frame**

data <- data.frame( name = c("John", "Alice", "Bob"),age = c(30, 25, 35),city = c("New York",  "London", "Paris"))

**Building the INSERT Query**

insert\_query <- paste0("INSERT INTO emp1 (name, age, city) VALUES ")

# Generate values for the INSERT statement

values <- paste0("('", data$name, "', ", data$age, ", '", data$city, "')")

# Combine the INSERT statement and values

insert\_query <- paste0(insert\_query, paste(values, collapse = ", "))

**Executing the INSERT Query**

This line sends the constructed SQL INSERT query to the MySQL database using the  'dbSendQuery' function. The data from the 'data' data frame is inserted into the 'emp1' table in the  'prac5' database.

# Execute the INSERT query

dbSendQuery(con, insert\_query)

**Closing the Database Connection**

The code closes the database connection to free up resources after the data has been inserted. # Close the database connection

dbDisconnect(con)