JECRC University, Jaipur

**Assignment Unit III**

**MCA - III Semester**

**Subject: Statistical Computing using R**

**Subject Code: MCA206A**

**Date of Release:**

**Last Date of Submission:**

**Part-A 10X1=10**

Q.1 Point out the correct statement?

A. Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.  
B. it tests the condition at the end of the loop body.  
C. Both A and B  
D. None of the above

Q.2 In which statement terminates the loop statement and transfers execution to the statement immediately following the loop?

A. goto  
B. switch  
C. break  
D. label

Q.3 How do you start writing a while loop in R?

A) while x < y:

B) x < y while

C) while x < y

D) while (x < y)

Q.4 What will be the output of the following R code? install.packages(c("devtools", "roxygen2"))

a) Develops the tools  
b) Exits R studio  
c) Installs the given packages  
d) Nothing happens

Q.5 Function can return values--

a) Single

b) Multiple

c) Both A and B

d) None of Above

Q.6 What will be the output of the following—

S<- “Veg”

Switch (S, Veg=”onion”, fruit= “Mango”, “Neither”)

a) onion  
b) Mango  
c) Neither  
d) Error

Q.7 Which function can be used to return the square root of a number?

a) sq  
b) sqr  
c) sqrt  
d) sqroot

Q.8 How do you create a Function?

a) my\_function <- function

b) my\_function <- function()  
c) my\_function <- function[]  
d) None of the above

Q.10 What will be the output of the following R function?

paste("Everybody", "is", “a” , "warrior")

a) Everybody is a warrior

b) “Everybody is a warrior”

c) Everybody”, “is”, “a” , “warrior

d) “Everybody”, “is”, “a” , “warrior”

Q.11 What will be the output of the following R function?

**Sys.Date()**

a) some date

b) today

c) tomorrow

d) yesterday

Q.12 Which of the following should be preferred for evaluation from list of alternatives?

a) subsett

b) eval

c) switch

d) set

Q.13  \_\_\_\_\_\_\_\_\_\_ function is used to watch for all available packages in library.

a) lib()

b) fun.lib()

c) libr()

d) library()

Q.14 Which apply function is used for simplify list.

a) apply

b) sapply

c) mapply

d) tapply

Q.15 A single element of a character vector is referred as \_\_\_\_\_\_\_

a) Character string

b)string

c) Data String

d)Raw data

Q.16 Dt = “01-12-2020” is in the form of character. What is the option to convert date into “MM-DD-YYYY

a) To\_date (dt, ”MM – DD – YYYY”)

b) date( x = dt, format = “%m / %d / %Y”)

c) Date ( x = dt, format = “%m / %d / %Y”)

1. none of the above

Q.17 Scripts will run on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a) Script Editors

b) Console

c) Terminal

d) GCC Compiler

Q.18 Which layer is not involved in ggplot2?

A) data

B) facets

C) list

D) theme

Q.19 How many packages approx. in R?

a) 100+

b) 1000+

c) 10000+

d) 100000+

Q.20 Which function is used for input operation?

a) scan

b) print

c) paste

d) cat

**Part-B 5X2=10**

Q.1What is Function? Write any 3 math functions in R.

a function is an object so the R interpreter is able to pass control to the function, along with arguments that may be necessary for the function to accomplish the actions.

The function in turn performs its task and returns control to the interpreter as well as any result which may be stored in other objects.

function\_name <- function(arg\_1, arg\_2, ...) {

Function body

}

max(5, 10, 15), min(5, 10, 15), sqrt(16)

Q.2 Distinguish between for and while loop in R.

The primary difference between a for loop and a while loop is: a for loop is used when the number of iterations a code should be run is known where a while loop is used when the number of iterations is not known.

Q.3 Explain about Data Mining Packages.

R is the most popular language for Data Science. There are many packages and libraries provided for doing different tasks. For example, there is dplyr and data.table for data manipulation, whereas libraries like ggplot2 for data visualization and data cleaning library like tidyr. Also, there is a library like 'Shiny' to create a Web application and knitr for the Report generation where finally mlr3, xgboost, and caret are used in Machine Learning.

Q.4 How can we plot bar chart?

A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. R uses the function **barplot()** to create bar charts. R can draw both vertical and Horizontal bars in the bar chart. In bar chart each of the bars can be given different colors.

Syntax

The basic syntax to create a bar-chart in R is −

barplot(H,xlab,ylab,main, names.arg,col)

Following is the description of the parameters used −

* **H** is a vector or matrix containing numeric values used in bar chart.
* **xlab** is the label for x axis.
* **ylab** is the label for y axis.
* **main** is the title of the bar chart.
* **names.arg** is a vector of names appearing under each bar.
* **col** is used to give colors to the bars in the graph.

Q.5 What are the Packages? Name some functions available in “dplyr” packages.

Packages in [R Programming language](https://www.geeksforgeeks.org/introduction-to-r-programming-language/) are a set of R functions, compiled code, and sample data. These are stored under a directory called “library” within the R environment. By default, R installs a group of packages during installation. Once we start the R console, only the default packages are available by default. Other packages that are already installed need to be loaded explicitly to be utilized by the R program that’s getting to use them.

|  |  |  |
| --- | --- | --- |
| **dplyr Function** | **Description** | **Equivalent SQL** |
| select() | Selecting columns (variables) | SELECT |
| filter() | Filter (subset) rows. | WHERE |
| group\_by() | Group the data | GROUP BY |
| summarise() | Summarise (or aggregate) data | - |
| arrange() | Sort the data | ORDER BY |
| join() | Joining data frames (tables) | JOIN |
| mutate() | Creating New Variables | COLUMN ALIAS |

**Part-C 5X6=30**

Q.1Write R code to return a complex object.

res2<-**function**()

{

v<-c(1,2,5,3,8)

m<-matrix(1:8,ncol=4)

v1<-mean(v)

m1<-min(m)

L<-list(vec=v1,mat=m1)

**return**(L)

}

res2()

## $vec

## [1] 3.8

##

## $mat

## [1] 1

Q.2 Explain various apply methods in R.

## **The apply() Function in R Programming**

 The apply function helps us to apply a function on rows or columns (margins) of a matrix or a data frame. It has syntax as shown below:

apply**(**x, MARGIN, FUN, ...**)**

 x - Stands for the matrix or the data frame on which we want to apply the function.

MARGIN - is a vector that defines, which part of the matrix/data frame the function should be applied on.

FUN - specifies the function that will be applied on the MARGIN.

## **The lapply() Function in R Programming**

The lapply takes a list as an argument and applies a function to each element of the list by looping. It has syntax as shown below:

lapply**(**x, FUN, ...**)**

X - Specifies a list on which functions should be replicated.

FUN - Is a function that needed to be looped on each element of the list.

## **The sapply() Function in R Programming**

The sapply is a more generalized version of lapply(). It works the same as lapply(), the only difference is in output generalization.The syntax for the sapply() function is as shown below:

sapply(X, FUN, ..., simplify **= TRUE,**USE.NAMES**= TRUE)**

X - specifies the list on which we want to apply the function.

FUN - specifies the function to be applied.

… - arguments that can be added.

simplify - argument that specifies if we want to simplify the results or not.

USE.NAME - specifies the argument names to be used or not.

## **The tapply() Function in R Programming**

The tapply() function can be applied on a subset of a vector where the vector is divided into different levels that are also known as factors. In such cases, where we want to break the data into different subgroups and apply a specific function on each of the subgroups, we can use the tapply() function.The syntax for tapply() function is as shown below:

tapply**(**x, INDEX, FUN, ..., simplify **= TRUE)**

x - is a vector on which the function is to be applied.

INDEX - is a vector of the factors.

FUN - is a function to be applied to each subgroup.

Simplify - is an argument which specifies if we want a simplified result or not. If we want a simplified result, we should use TRUE otherwise FALSE.

## **The mapply() Function in R Programming**

The mapply() is a multivariate version of sapply() function in R. The mapply() applies a function in parallel to the given set of arguments. It applies the same function to each argument passed. Meaning, if we want to sum, the sum() function will be applied over the first argument, second argument, third argument, and so on.The syntax for the mapply() function is as shown below:

mapply(FUN, ..., MoreArgs**= NULL,**SIMPLIFY**= TRUE,**USE.NAMES**= TRUE)**

FUN - is the function to be applied over the objects.

MoreArgs - specifies the other arguments for the FUN

SIMPLIFY - specifies whether we want simplified results or not.

USE.NAMES - specifies if we want the names for arguments or not.

Q.3 Write a for loop that iterates over the numbers 1 to 7 and prints the cube of each number using print()

for (j in 1:7)

{

print(j\*j\*j)

}

Q.4. Distinguish between cat(), Paste() and Print().

* The **cat()** function will output the concatenated string to the console, but it won’t store the results in a variable.
* The **paste()** function will output the concatenated string to the console and **it will store** the results in a character variable.

The **print()** function returns a data object to the R (or RStudio) console

Q.5. Write a R script to find out factorial of given number using recursion.

recur\_factorial <- function(n) {

if(n <= 1) {

return(1)

} else {

return(n \* recur\_factorial(n-1))

}

}

**Output**

> recur\_factorial(5)

[1] 120

**Part-D 2X10=20**

Q.1 a script that will print 'Is a Matrix' if the variable x is a matrix, otherwise "Not a

Matrix

Q.1 (a) Use a nested for loop that produces the following matrix, pre-allocate the matrix with NA values.

0 1 2 3 4

1 0 1 2 3

2 1 0 1 2

3 2 1 0 1

4 3 2 1 0

mat <- matrix(NA\_integer\_, nrow = 5, ncol = 5)

for (i in 1:5) {

for (j in 1:5) {

mat[i, j] <- abs(i - j)

}

}

mat

(b) Write a program to implement Looping over list and vector.

List:

# Create a list with three vectors

fruit <- list(Basket = c('Apple', 'Orange', 'Passion fruit', 'Banana'),

Money = c(10, 12, 15), purchase = FALSE)

for (p in fruit)

{

print(p)

}

#### Output:

## [1] "Apple" "Orange" "Passion fruit" "Banana"

## [1] 10 12 15

## [1] FALSE

Vector:

# Create fruit vector

fruit <- c('Apple', 'Orange', 'Passion fruit', 'Banana')

# Create the for statement

for ( i in fruit){

print(i)

}

#### Output:

## [1] "Apple"

## [1] "Orange"

## [1] "Passion fruit"

## [1] "Banana"

Q.2. (a) What do you mean by Function Scoping. Distinguish Lexical and Dynamic scoping with example.

Function scope  
  
This scope means that **the variables are only accessible in the function in which they are declared**.

Table

Description automatically generated

(b) Explain structure of ggplot2 code with example.

**ggplot2 package in R Programming Language** also termed as **Grammar of Graphics** is a free, open-source, and easy-to-use visualization package widely used in [R](https://www.geeksforgeeks.org/introduction-to-r-programming-language/). It is the most powerful visualization package written by Hadley Wickham.

It includes several layers on which it is governed. The layers are as follows:

## Building Blocks of layers with the grammar of graphics

* **Data:** The element is the data set itself
* **Aesthetics:** The data is to map onto the Aesthetics attributes such as x-axis, y-axis, color, fill, size, labels, alpha, shape, line width, line type
* **Geometrics:** How our data being displayed using point, line, histogram, bar, boxplot
* **Facets:** It displays the subset of the data using Columns and rows
* **Statistics:** Binning, smoothing, descriptive, intermediate
* **Coordinates:** the space between data and display using Cartesian, fixed, polar, limits
* **Themes:** Non-data link

Chart, funnel chart, surface chart

Description automatically generated

Example:

ggplot(mydata100) + geom\_bar( aes(workshop) )

[Chart, bar chart

Description automatically generated](https://i0.wp.com/r4stats.com/wp-content/uploads/2022/08/GGbarWorkshop-1.png?ssl=1)