**ITA 0443 - STATISTICS WITH R PROGRAMMING**

**DAY 2 – LAB EXERCISES**

**Reg No : 192121038**

**Name : B.Ramya**

**IMPLEMENTATION OF VECTOR RECYCLING, APPLY FAMILY & RECURSION**

**1. Demonstrate Vector Recycling in R.**

> vec1=1:6

> vec2=1:2

> print(vec1+vec2)

[1] 2 4 4 6 6 8

**2. Demonstrate the usage of apply function in R**

> sample\_matrix <- matrix(C<-(1:10),nrow=3, ncol=10)

>

> print( "sample matrix:")

[1] "sample matrix:"

> sample\_matrix

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]

[1,] 1 4 7 10 3 6 9 2 5 8

[2,] 2 5 8 1 4 7 10 3 6 9

[3,] 3 6 9 2 5 8 1 4 7 10

> print("sum across rows:")

[1] "sum across rows:"

> apply( sample\_matrix, 1, sum)

[1] 55 55 55

> print("mean across columns:")

[1] "mean across columns:"

> apply( sample\_matrix, 2, mean)

[1] 2.000000 5.000000 8.000000 4.333333 4.000000 7.000000 6.666667 3.000000

[9] 6.000000 9.000000

**3. Demonstrate the usage of lapply function in R**

> names <- c("priya", "raj","pawan","sudha","devraj")

> print( "original data:")

[1] "original data:"

> names

[1] "priya" "raj" "pawan" "sudha" "devraj"

> print("data after lapply():")

[1] "data after lapply():"

> lapply(names, toupper)

[[1]]

[1] "PRIYA"

[[2]]

[1] "RAJ"

[[3]]

[1] "PAWAN

[[4]]

[1] "SUDHA"

[[5]]

[1] "DEVRAJ"

**4. Demonstrate the usage of sapply function in R**

> sample\_data<- data.frame( x=c(1,2,3,4,5,6),y=c(3,2,4,2,34,5))

> print( "original data:")

[1] "original data:"

> sample\_data

x y

1 1 3

2 2 2

3 3 4

4 4 2

5 5 34

6 6 5

> print("data after sapply():")

[1] "data after sapply():"

> sapply(sample\_data, max)

x y

6 34

**5. Demonstrate the usage of tapply function in R**

library(tidyverse)

print(" Head of data:")

head(diamonds)

print("Average price for each cut of diamond:")

tapply(diamonds$price, diamonds$cut, mean)

**6. Demonstrate the usage of mapply function in R**

> A = list(c(1, 2, 3, 4))

> B = list(c(2, 5, 1, 6))

> result = mapply(sum, A, B)

> print(result)

[1] 24

**7. Sum of Natural Numbers using Recursion**

> sum<-function(n){

> if (n<=1){

> return(n)

> }else{

> return(n+sum(n-1))

> }

> }

> sum(7)

[1] 28

**8. Write a program to generate Fibonacci sequence using Recursion in R**

> Fibonacci <- numeric(10)

> Fibonacci[1] <- Fibonacci[2] <- 1

> for (i in 3:10) Fibonacci[i] <- Fibonacci[i - 2] + Fibonacci[i - 1]

> print("First 10 Fibonacci numbers:")

[1] "First 10 Fibonacci numbers:"

> print(Fibonacci)

[1] 1 1 2 3 5 8 13 21 34 55

**9. Write a program to find factorial of a number in R using recursion.**

rec\_fac <- function(x){

if(x==0 || x==1)

{

return(1)

}

else

{

return(x\*rec\_fac(x-1))

}

}