SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

ITA 0443 - STATISTICS WITH R PROGRAMMING FOR REAL TIME PROBLEM

DAY 2 – LAB EXERCISES

IMPLEMENTATION OF VECTOR RECYCLING, APPLY FAMILY &amp; RECURSION

1. Demonstrate Vector Recycling in R.

PROGRAM:

vec1=1:6

vec2=1:2

print(vec1+vec2)

OUTPUT:

[1] 2 4 4 6 6 8

2. Demonstrate the usage of apply function in R

PROGRAM:

m1 <- matrix(C<-(1:10),nrow=5, ncol=6)

m1

a\_m1 <- apply(m1, 2, sum)

a\_m1

OUTPUT:

[1] 15 40 15 40 15 40

3. Demonstrate the usage of lapply function in R

PROGRAM:

movies <- c("SPYDERMAN","BATMAN","VERTIGO","CHINATOWN")

movies\_lower <-lapply(movies, tolower)

str(movies\_lower)

OUTPUT:

List of 4

$ : chr "spyderman"

$ : chr "batman"

$ : chr "vertigo"

$ : chr “chinatown"

4.Demonstrate the usage of sapply function in R.

PROGRAM:

dt <- cars

lmn\_cars <- lapply(dt, min)

smn\_cars <- sapply(dt, min)

lmn\_cars

OUTPPUT:

$speed

[1] 4

$dist

[1] 2

5.Demonstrate the usage of tapply function in R.

PROGRAM :

set.seed(2)

data\_set <- data.frame(price = round(rnorm(25, sd = 10, mean = 30)),

type = sample(1:4, size = 25, replace = TRUE),

store = sample(paste("Store", 1:4),

size = 25, replace = TRUE))

head(data\_set)

price <- data\_set$price

store <- data\_set$store

type <- factor(data\_set$type,

labels = c("toy", "food", "electronics", "drinks"))

mean\_prices <- tapply(price, type, mean)

mean\_prices

OUTPUT:

toy food electronics drinks

39.50000 30.33333 32.20000 29.33333

6. Demonstrate the usage of mapply function in R

PROGRAM:

mapply(rep, 1:3, times=5)

OUTPUT:

[,1] [,2] [,3]

[1,] 1 2 3

[2,] 1 2 3

[3,] 1 2 3

[4,] 1 2 3

[5,] 1 2 3

7. Sum of Natural Numbers using Recursion

PROGRAM:

sum<-function(n){

if (n<=1){

return(n)

}else{

return(n+sum(n-1))

}

}

sum(7)

OUTPUT:

[1] 28

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

PROGRAM:

nterms = as.integer(readline(prompt="How many terms? "))

n1 = 0

n2 = 1

count = 2

if(nterms <= 0) {

print("Plese enter a positive integer")

} else {

if(nterms == 1) {

print("Fibonacci sequence:")

print(n1)

} else {

print("Fibonacci sequence:")

print(n1)

print(n2)

while(count < nterms) {

nth = n1 + n2

print(nth)

# update values

n1 = n2

n2 = nth

count = count + 1

}

}

}

OUTPUT:

How many terms? 10

[1] "Fibonacci sequence:"

[1] 0

[1] 1

[1] 1

[1] 2

[1] 3

[1] 5

[1] 8

[1] 13

[1] 21

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

PROGRAM:

recur\_fact <- function(n) {

if(n <= 1) {

return(1)

} else {

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

}

}

recur\_fact(8)

num<-8

print(factorial(num))

fact <- 1

if (num < 0) {

print("Factorial for negative numbers not allowed!")

} else if (num == 0) {

print("The factorial of 0 is 1")

} else {

for(i in 1:num){

fact=fact\*i

}

print(fact)

}

OUTPUT:

[1] 40320