**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**SAVEETHA SCHOOL OF ENGINEERING**

**LABORATORY MANUAL**

**ITA04 Statistics with R Programming**

**T.Abishek sam**

**192111069**

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

**Exercises:**

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

**CODE:**

**v1=c(1,2,3,4,5)**

**v2=c(1,2)**

**print(v1+v2)**

**OUTPUT:**

[1] 2 4 4 6 6

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

**CODE:**

**print("matrix :")**

**r=c(1,2,3,4)**

**mat=matrix(r1,ncol = 2)**

**print(mat)**

**print("Applying sum into the columns of matrix :")**

**print(apply(arr,2,sum))**

**OUTPUT:**

[1] "matrix :"

[,1] [,2]

[1,] 1 3

[2,] 2 4

[1] "Applying sum into the columns of matrix :"

[1] 3 7

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

**CODE:**

**data <- data.frame(a = c(1, 2, 3),b = c(1, 2, 3),c = c(1, 2, 3))**

**print(data)**

**print(lapply(data,sum))**

**OUTPUT:**

a b c

1 1 1 1

2 2 2 2

3 3 3 3

$a

[1] 6

$b

[1] 6

$c

[1] 6

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

**CODE:  
data <- data.frame(a = c(1, 2, 3),b = c(1, 2, 3),c = c(1, 2, 3))**

**print(data)**

**print(sapply(data,sum))**

**OUTPUT:** a b c

1 1 1 1

2 2 2 2

3 3 3 3

a b c

6 6 6

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

**CODE:  
data <- data.frame(a = c(3, 4, 1),b = c(6, 2, 3),c = c(5,8,10))**

**print(data)**

**print(tapply(data$c,data$a,mean))**

**OUTPUT:**a b c

1 3 6 5

2 4 2 8

3 1 3 10

1 3 4

10 5 8

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

**CODE:**

**v1 <- c(1, 2, 3, 4, 5)**

**v2 <- c(2, 4, 1, 2, 6)**

**print(mapply(max, v1, v2))**

**OUTPUT:**[1] 2 4 3 4 6

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

**CODE:**

**n=as.integer(readline(prompt = "Enter number of natural numbers :"))**

**sum=0**

**for(i in 1:n){**

**sum=sum+i**

**}**

**print(paste("Sum of ",n," natural numbers :",sum))**

**OUTPUT:**

[1] "Sum of 10 natural numbers : 55"

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

**CODE:**

**fibo<- function(n) {**

**if(n<= 1) {**

**return(n)**

**} else {**

**return(fibo(n-1) + fibo(n-2))**

**}**

**}**

**n = as.integer(readline(prompt = "Enter number of digits in series :"))**

**print("Fibonacci sequence:")**

**for(i in 0:(n-1)) {**

**print(fibo(i))**

**}**

**OUTPUT:**

[1] "Fibonacci sequence:"

[1] 0

[1] 1

[1] 1

[1] 2

[1] 3

[1] 5

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

**CODE:**

**fact<- function(n) {**

**if(n <= 1) {**

**return(1)**

**} else {**

**return(n\*fact(n-1))**

**}**

**}**

**x=as.integer(readline(prompt = "Enter a number :"))**

**print(paste("factorial :",fact(x)))**

**OUTPUT:**

Enter a number :4

[1] "factorial : 24"

**5.CREATION AND MANIPULATION OF DATAFRAMES IN R**

**Exercise 1**

Consider two vectors: x=seq(1,43,along.with=Id)

y=seq(-20,0,along.with=Id)

Create a data frame ‘df’ as shown below.

>df

Id Letter x y

1 1 a 1.000000 -20.000000

2 1 b 4.818182 -18.181818

3 1 c 8.636364 -16.363636

4 2 a 12.454545 -14.545455

5 2 b 16.272727 -12.727273

6 2 c 20.090909 -10.909091

7 3 a 23.909091 -9.090909

8 3 b 27.727273 -7.272727

9 3 c 31.545455 -5.454545

10 4 a 35.363636 -3.636364

11 4 b 39.181818 -1.818182

12 4 c 43.000000 0.000000

CODE:

x=1

Id=c()

letter=c()

for(i in 1:12){

Id=append(Id,x,i)

if(i%%3==0){

x=x+1

}

}

y=1

for(i in 1:4){

for(j in 1:3){

letter=append(letter,letters[j],y)

y=y+1

}

}

x=seq(1,43,along.with=Id)

y=seq(-20,0,along.with=Id)

df=data.frame(Id,letter,x,y)

print(df)

**Exercise 2**

Using the data frame ‘df’ in Exercise1, Construct the following data frame. Id x.ay.ax.by.bx.cy.c 1 1 1.00000 -20.000000 4.818182 -18.181818 8.636364 -16.363636 4 2 12.45455 -14.545455 16.272727 -12.727273 20.090909 -10.909091 7 3 23.90909 -9.090909 27.727273 -7.272727 31.545455 -5.454545 10 4 35.36364 -3.636364 39.181818 -1.818182 43.000000 0.000000

**Exercise 3**

Create two data frame df1 and df2:

> df1

Id Age

1. 1 14
2. 2 12
3. 3 15
4. 4 10

> df2

Id Sex Code

1. 1 F a
2. 2 M b
3. 3 M c
4. 4 F d

From df1 and df2 create M:

>M

Id Age Sex Code

1. 1 14 F a
2. 2 12 M b
3. 3 15 M c 4 4 10 F d

CODE:

id=c(1:4)

age=c(14,12,15,10)

sex=c("F","M","M","F")

code=c("a","b","c","d")

df1=data.frame(id,age)

df2=data.frame(id,sex,code)

df3=merge(df1,df2)

print(df3)

## **Exercise 4**

Create a data frame df3:

> df3 id2 score 1 4 100

1. 3 98
2. 2 94
3. 1 99

From M (used in Exercise-3) and df3 create N:

Id Age Sex Code score

1. 1 14 F a 99
2. 2 12 M b 94
3. 3 15 M c 98 4 4 10 F d 100

CODE:

id=c(1:4)

age=c(14,12,15,10)

sex=c("F","M","M","F")

code=c("a","b","c","d")

df1=data.frame(id,age)

df2=data.frame(id,sex,code)

df2=merge(df1,df2)

score=c(100,98,94,99)

id2=c(4:1)

df3=data.frame(id,score)

df3=merge(df2,df3)

print(df3)

## **Exercise 5**

Consider the previous one data frame N:

1) Remove the variables Sex and Code

2) From N, create a data frame:

values ind

1. 1 Id
2. 2 Id
3. 3 Id
4. 4 Id
5. 14 Age
6. 12 Age
7. 15 Age
8. 10 Age
9. 99 score
10. 94 score
11. 98 score
12. 100 score

CODE:

library(reshape)

id=c(1:4)

age=c(14,12,15,10)

sex=c("F","M","M","F")

code=c("a","b","c","d")

df1=data.frame(id,age)

df2=data.frame(id,sex,code)

df2=merge(df1,df2)

score=c(100,98,94,99)

id=c(4:1)

df3=data.frame(id,score)

df3=merge(df2,df3)

df3$sex=NULL

df3$code=NULL

print(melt.data.frame(df3))

## **Exercise 6**

For this exercise, we’ll use the (built-in) dataset trees.

1. Make sure the object is a data frame, if not change it to a data frame.
2. Create a new data frame A:

>A

Girth Height Volume

mean\_tree 13.24839 76 30.17097

min\_tree 8.30000 63 10.20000

max\_tree 20.60000 87 77.00000

sum\_tree 410.70000 2356 935.30000

CODE:

df=as.data.frame(summary(as.data.frame(trees)))

print(df)

## **Exercise 7**

Consider the data frame A:

1)Order the entire data frame by the first column.

2)Rename the row names as follows: mean, min, max, tree

CODE:

mean\_tree=apply(trees,2,mean)

max\_tree=apply(trees,2,max)

min\_tree=apply(trees,2,min)

sum\_tree=apply(trees,2,sum)

a=data.frame(mean\_tree,min\_tree,max\_tree,sum\_tree)

a=t(a)

print(a[order(a[,1]),])

row.names(a)=c('mean','min','max','sum')

## **Exercise 8**

Create an empty data frame with column types:

>df

IntsLogicals Doubles Characters

(or 0-length row.names)

## CODE:

df1 <- data.frame(Doubles=double(),

Integers=integer(),

Factors=factor(),

Logicals=logical(),

Characters=character(),

stringsAsFactors=FALSE)

print(df1)

## **Exercise 9**

Create a data frame XY

X=c(1,2,3,1,4,5,2)

Y=c(0,3,2,0,5,9,3)

> XY

X Y

1. 1 0
2. 2 3
3. 3 2
4. 1 0
5. 4 5
6. 5 9
7. 2 3
8. look at duplicated elements using a provided R function.
9. keep only the unique lines on XY using a provided R function.

CODE:  
x=c(1,2,3,1,4,5,2)

y=c(0,3,2,0,5,9,3)

df=data.frame(x,y)

print("dataframe :")

print(df)

print("unique :")

print(unique(df))

print("Duplicated :")

print(duplicated(df))

## **Exercise 10**

Use the (built-in) dataset Titanic.

1. Make sure the object is a data frame, if not change it to a data frame.
2. Define a data frame with value 1st in Class variable, and value NO in Survived variable and variables Sex, Age and Freq.

Sex Age Freq

1 Male Child 0

5 Female Child 0

9 Male Adult 118

13 Female Adult 4

CODE:  
a=subset(data.frame(Titanic), subset = Class=='1st' & Survived=='No',select=c(Sex,Age,Freq))

print(a)

MERGING DATAFRAMES

## **Exercise 11 a)**

Create the following dataframes to merge:

buildings<- data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data <- data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

The dataframes, *buildings*and *data*have a common key variable called, “location”.

Use the merge() function to merge the two dataframes by “location”, into a new dataframe,“buildingStats”.

CODE:  
buildings= data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data=data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

buildingstats=merge.data.frame(buildings,data)

print(buildingstats)

## **Exercise 11 b)**

Give the dataframes different key variable names:

buildings<- data.frame(location=c(1, 2, 3), name=c("building1","building2", "building3"))

data <- data.frame(survey=c(1,1,1,2,2,2), LocationID=c(1,2,3,2,3,1), efficiency=c(51,64,70,71,80,58))

The dataframes, buildings and data have corresponding variables called, location, and LocationID. Use the merge() function to merge the columns of the two dataframes by the corresponding variables.

CODE:  
buildings= data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data=data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

buildingstats=merge.data.frame(buildings,data)

print(buildingstats)

DIFFERENT TYPES OF MERGE IN R

**Exercise 12a)InnerJoin:**

The R merge() function automatically joins the frames by common variable names. In that case, demonstrate how you would perform the merge in **Exercise 11a** without specifying the key variable.

CODE:

buildings= data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data=data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

buildingstats=merge.data.frame(buildings,data)

print(buildingstats)

**Exercise 12b)OuterJoin:**

Merge the two dataframes from **Exercise 11a**. Use the “all=” parameter in the merge() function to return all records from both tables. Also, merge with the key variable, “location”.

CODE:  
buildings= data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data=data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

buildingstats=merge.data.frame(buildings,data,all = TRUE,by="location")

print(buildingstats)

**Exercise 12c)Left Join:**

Merge the two dataframes from **Exercise 11a**, and return all rows from the left table. Specify the matching key from **Exercise 11a.**

**CODE:**buildings= data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data=data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

buildingstat=merge.data.frame(buildings,data)

buildings= data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data=data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

buildingstats=merge.data.frame(buildings,data,all = TRUE,by="location")

print(merge.data.frame(buildingstat,buildingstats))

**Exercise 12d)Right Join:**

Merge the two dataframes from **Exercise 11a,** and return all rows from the right table. Use the matching key from **Exercise 11a** to return matching rows from the left table.

**Exercise 12e)Cross Join:**

Merge the two dataframes from **Exercise 11a**, into a “Cross Join” with each row of “buildings” matched to each row of “data”. What new column names are created in “buildingStats”?

### **Exercise 13MergingDataframe rows:**

To join two data frames (datasets) vertically, use the rbind function. The two data frames must have the same variables, but they do not have to be in the same order.

Merge the rows of the following two dataframes:

buildings<- data.frame(location=c(1, 2, 3), name=c("building1",

"building2", "building3"))

buildings2 <- data.frame(location=c(5, 4, 6), name=c("building5", "building4", "building6"))

Also, specify the new dataframe as, “allBuidings”.

### **Exercise 14**

Create a new dataframe, buildings3, that has variables not found in the previous dataframes.

buildings3 <- data.frame(location=c(7, 8, 9), name=c("building7", "building8", "building9"), startEfficiency=c(75,87,91))

Create a new buildings3 without the extra variables.

### **Exercise 15**

Instead of deleting the extra variables from buildings3. append the buildings, and buildings2 with the new variable in buildings3, **(from Exercise 14).** Set the new data in buildings and buildings2 , (**from Exercise 13)**, to NA.