Name: Anshika Jain

Email ID: Pgp20anshikajain@imt.ac.in

Roll number: 200103034

DSA Individual Assignment

Submitted to:

Prof. (Dr.) Sridhar Vaithianathan

Associate Professor, analytics, IMT – Hyderabad

18th December, 2020

_____

R Markdown

```r
#basic functions (mathematic operations)
a=5 #assigning value to variable a
b=7 #assigning value to variable b
a+b
```

```
## [1] 12
```

```r
a-b
```

```
## [1] -2
```

```r
a*b
```

```
## [1] 35
```

```r
a/b
```

```
## [1] 0.7142857
```

```r
a%%b #Modulus function: Remainder of a/b
```

```
## [1] 5
```

```r
#print function

print("I am anshika jain")

## [1] "I am anshika jain"

sprintf("sum of the two numbers=%d", 4+7) #%d used for whole number output

## [1] "sum of the two numbers=11"

sprintf("modulus of 80/3=%f", 80/3) #%f used for fraction

## [1] "modulus of 80/3=26.666667"

sprintf("modulus of 80/3=%0.2f", 80/3) #%0.2f inidcates output till 2 decimal
spaces

## [1] "modulus of 80/3=26.67"

a=80
b=3
round(a/b) #Round off to the nearest whole number

## [1] 27

##vectors concept

A= c(1,2,3,4,5) #A is a variable, c is the combined function
length(A) #Finding the length of the combined function A

## [1] 5

A[c(1,3)] #Finding the number on the 1st and 3rd position

## [1] 1 3

A[5]= 10  #Replacing the number on the 5th position with 10
A

## [1]  1  2  3  4 10

sort(A) #Arranging in ascending order

## [1]  1  2  3  4 10

#Sequence concept

#creating a sequence from 6 to 21 with a difference of 3
AddThree = seq(from=6, to=21, by=3)
AddThree

## [1]  6  9 12 15 18 21
```

```r
#creating a series of 10 even numbers
Even= seq(from=2, by=2, length=10)
Even
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```r
#Value of TRUE = 1 and False= 0

sprintf("is four present in even?=%s", 4%in%Even) #%s is to get output in
terms of true or false
```

```
## [1] "is four present in even?=TRUE"
```

```r
sprintf("is four present in even?=%d", 4%in%Even) #%d is to get output in
terms of 1(true) or 0(false)
```

```
## [1] "is four present in even?=1"
```

```r
#If-Else function

#Identifying the eligibility for voting or driving or both from age data

age=19

if(age>=18){
  print("Drive and vote")
} else if(age>=16){
  print("Drive")
} else
  print("Not eligible")
```

```
## [1] "Drive and vote"
```

```r
#Switch concept

grade = "B"

switch (grade ,
  "A" = print("Great"),
  "B" = print("Good"),
  "c" = print("Average"),
  "D" = print("Bad"),
  print("no such grade"))
```

```
## [1] "Good"
```

```r
#String concept

str1= "this is a string module"
nchar(str1) #counting the number of characters in a string
```

```
## [1] 23
```

```r
str2= paste("owl", "bear", sep = ",") #sep refers to separator (how words in
string are to be separated)
str2
```

```
## [1] "owl,bear"
```

```r
substr(str2, 5, 8) #printing a sub-part of a string (4 and 8 are starting
character and ending character)
```

```
## [1] "bear"
```

```r
sub("owl", "hawk", str2) #replacing a word in a string
```

```
## [1] "hawk,bear"
```

```r
#Factors vector concept

Direction = c("up", "down", "right", "left", "down", "left")

factor(Direction) #Finding unique values from a function
```

```
## [1] up    down  right left  down  left
## Levels: down left right up
```

```r
#Data Frames concept

#Data frames concept enables you to document data in tabular form and present
data more logically and clearly

customerdata = data.frame(name = c("Tom", "sally", "Sue"), age= c(43,28,42),
stringsAsFactors = T)

customerdata
```

```
##    name age
## 1   Tom  43
## 2 sally  28
## 3   Sue  42
```

```r
NewRecord = data.frame(name="mark", age=22)
finalCustomerData = rbind(NewRecord, customerdata) #Binding two frames by
rows
finalCustomerData
```

```
##    name age
## 1  mark  22
## 2   Tom  43
## 3 sally  28
## 4   Sue  42
```

```r
customerdata[1,1:2] #fetching details from row 1, two columns
```

```
##    name age
## 1  Tom  43

CustomerId = c(101,102,103,104)
CustData= cbind(finalCustomerData,CustomerId) #binding two frames by column
CustData

##     name age CustomerId
## 1  mark  22        101
## 2   Tom  43        102
## 3 sally  28        103
## 4   Sue  42        104

#Matrix Concept

Mat= matrix(data = c(1,2,3,4,5), nrow = 5, ncol =5)
Mat

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    1    1
## [2,]    2    2    2    2    2
## [3,]    3    3    3    3    3
## [4,]    4    4    4    4    4
## [5,]    5    5    5    5    5

plot(Mat)
```
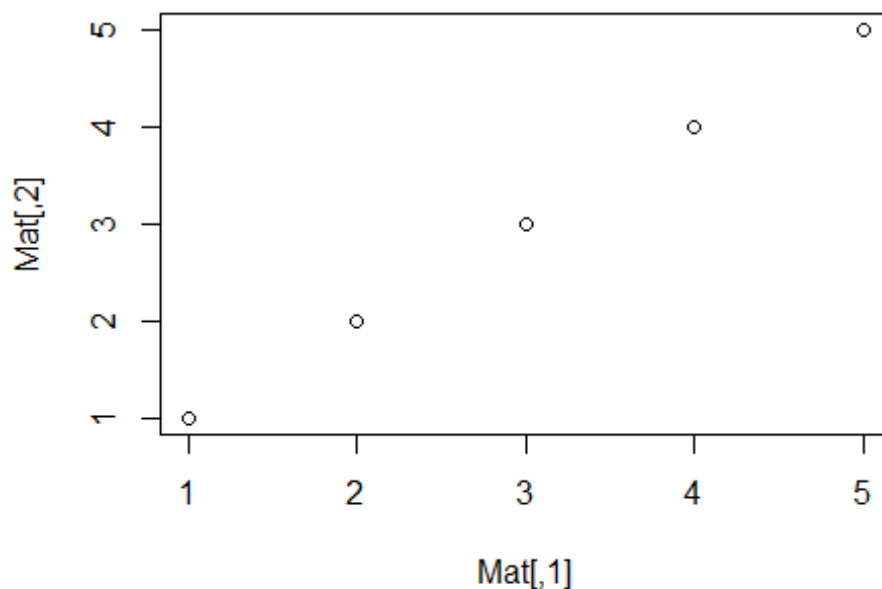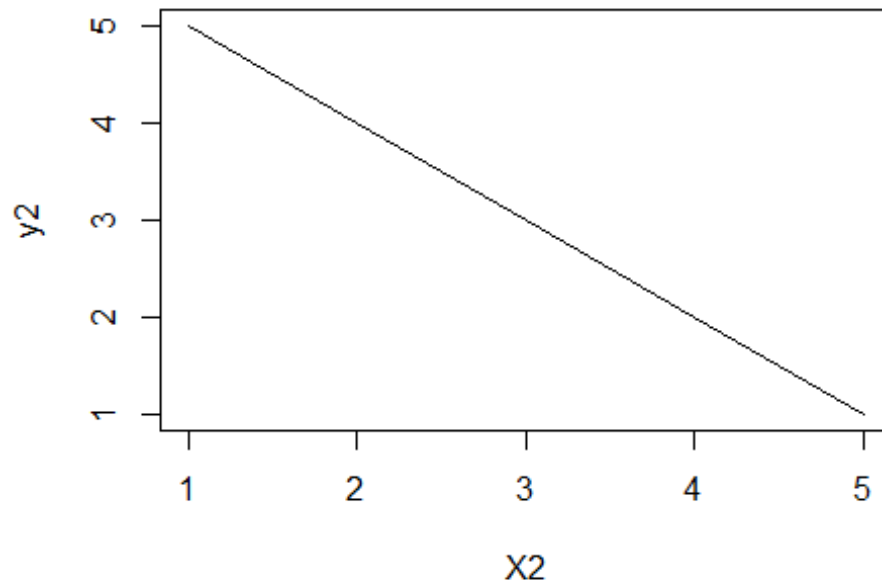
```
#Plotting a line
X2= c(1,2,3,4,5)
y2= c(5,4,3,2,1)

plot(X2, y2, type = "l")
```



```
#Array Concept

theArray = array(1:12, dim=c(2,3,2))# Total Elements = R x C x OD
theArray

## , , 1
##
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]    7    9   11
## [2,]    8   10   12

theArray [1, ,]# Accessing all elements from Row 1, all columns, all outer
dimensions & build C x OD (R x C)
```

```
##      [,1] [,2]
## [1,]    1    7
## [2,]    3    9
## [3,]    5   11
```

```r
theArray[1, ,1]# Accessing all elements from Row 1, all columns, first outer
dimension
```

```
## [1] 1 3 5
```

```r
theArray[, ,1]# Accessing all rows, all columns, first outer dimension
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```r
# Array with Four Outer Dimensions (OD)
theArray_4D = array(1:32, dim=c(2,4,4))
theArray_4D
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
## [2,]    2    4    6    8
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]    9   11   13   15
## [2,]   10   12   14   16
##
## , , 3
##
##      [,1] [,2] [,3] [,4]
## [1,]   17   19   21   23
## [2,]   18   20   22   24
##
## , , 4
##
##      [,1] [,2] [,3] [,4]
## [1,]   25   27   29   31
## [2,]   26   28   30   32
```

```r
theArray_4D [1, ,]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    9   17   25
## [2,]    3   11   19   27
## [3,]    5   13   21   29
## [4,]    7   15   23   31
```

```
theArray_4D[1, ,1]
```

```
## [1] 1 3 5 7
```

```
theArray[, ,1]
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

Learnings from the assignment

- R is a simple programming language containing 'easy to use' functions

- R programming allows instant execution of mathematical operations, conditions, and parameters

- R programming has simple functions that can be used in treatment of huge data sets in a short span of time

- R programming is also an efficient tool for statistical operations, graphical representation and more

- R programming enables simple and 'easy to understand' ways of documentation, calculations as well as analysis