Assignment No. 6.

Date / /20

Title:- Data types, Graphics & control structure in R

Problem statement:- To strudy & practice various Commands using different data types, Graphics & control structures on R tool & study & practice of various control structure

Pre-Lab:- A basic understanding of any of the programing language will help in executing simple R commands & control structure

Theory:- A. Data types in R:- In constant to other programming languages like C & java in R, The variables are not declared as some data types. The variable are assigned with R-objects & the data type of R-object becomes the data type of the R-object becomes the data type of the variable. There are many types of R-objects The frequently used ones are -

- vector
- Lists,
- matrices.
- Array
- factors
- Data frames

The simplest of these objects is the vector object & there are six data types of these atomic vectors, also termed as six classes of vectors The R-objects are built upon the actomic vectors.

Data types	Examples.
logical	TRUE, FALSE
Numeric	12.35, 999
Integer	2L, 34L, 0L
complex	3+2i
character	'a', "good", "TRUE", '23.4'
Raw	"Hello" is stored as 48 65 6c 6c 6f.

vector:- When you want to create vector with more than one element, you should use c() function which means to combine the elements into a vector.

Assignment No. 6.

Title:- Data types, Graphics & control structure in R

Problem statement:- To study & practice various commands using different data types, Graphics & control structures on R tool & study & practice of various control structure

Pre-Lab:- A basic understanding of any of the programing language will help in executing simple R commands & control structure

Theory:- A. Data types in R:- In constant to other programming languages like C & java in R, The variables are not declared as some data types. The variable are assigned with R-objects & the data type of R-object becomes the data type of the R-object becomes the data type of the variable. There are many types of R-objects. The frequently used ones are -

- vector
- Lists,
- matrices.
- Array
- factors
- Data frames

The simplest of these objects is the vector object & there are six data types of these atomic vectors, also termed as six classes of vectors. The R-objects are built upon the actomic vectors.

Data types	Examples.
Logical	TRUE, FALSE
Numeric	12.35, 999
Integer	2L, 34L, 0L
complex	3+2i
character	'a', 'good', "TRUE", '23.4'
Raw	"Hello" is stored as 48 65 6c 6c 6f.

Vector:- When you want to create vector with more than one element, you should use c() function which means to combine the elements into a vector.

```
#Create a vector.
  apple <- c('red', 'green', 'Yellow')
   print (apple)
 # Get the class of the vector.
    print (class(apple))
```

2. List :- A List is an R-object which can contain many different types of elements inside it like vectors, functions & even another list inside it.

```
  # create a list.
   list <- list (c(2,5,3), 2.1, 3, sin)
   #print (list 1)
```

3. Matrices :- A matrix is a two dimensional rectangular data set. It can be created using a vector input to the matrix function.

```
  # create a matrix
    M = matrix(c('a', 'a', 'b', 'c', 'b', 'a'), nrow = 2 , ncol=3,
    byrow = TRUE)
    print(M)
```

4. Arrays :- While matrices are confined to two dimension arrays can be of any number of dimensions. The array function takes a dim attribute which creates the required number of dimension. In the below example we create an array with two elements which are 3x3 matrices each. # Create an array.

```
       a <- array (c('green', 'Yellow'), dim = c(3, 3, 2))
        print (a)
```

5. Factors :- Factors are ~~call~~ created using a vector. It stores the vector along with the distinct value of the elements in the vector as labels. The labels are always character irrespective of whether it is numeric or character or Boolean etc in the input vector. They are useful in statistical modeling.

factors are created using the factor() function The nlevels functions gives the count of levels.

```
# create a vector.
  apple_colors <- c('green', 'green', 'yellow', 'red', 'red',
                    'red', 'green')
  # create a factor object.
    factor_apple <- factor (apple_colors)
  # print the factor.
    print (factor-apple)
    print (nlevels (factor-apple))
```

6. Data frames :- Data frames are tabular data object unlike a matrix in data frame each column can contain different modes of data The first column can be numeric while the ~~secl~~ second column can be character & third column can be logical. It is a list of vectors of equal length. Data frames are created using the data.frame() function. # Create the data frame

```
    BMI <- data.frame(
            gender = c("male", "male", "female"),
            height = c(152, 171.5, 165),
            weight = c(81, 93, 78),
            Age    = c(42, 38, 26)
            ) print(BMI)
```

On vectors, lists, & matrices arithmetic functions can be performed using the basic arithmetic syntax.

7. Strings :- Any value written within a pair of single quote or double quotes in R is treated as a string. Internally R stores every string within double quotes even when you create them with single quote.

Rules applied in string construction:-

- The quotes at the begining and of a string should be both double quotes or both single quote. They can not be mixe
- Double quotes can be inserted into a string starting & ending with single quote.
- Single quote can be inserted into a string starting & ending with double quotes
- single quote can not be inserted into a string starting & ending with single quote.

```
b <- "Start & end with double quotes"
  print (b)
```

String Manipulation function used in R:-

Concatenating String - paste() function. formating numbe & strings - format() function. Counting number of characters in a string - nchar() function.

B. Graphics:- R programming language has numerous librari to create charts & graphs

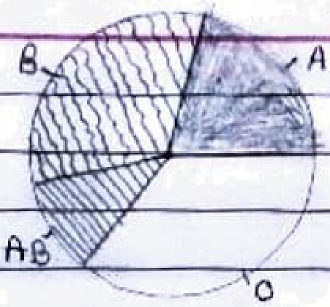
1. Pie Chart:- A pie-chart is a representation of values as slices of a circle with different colors. The slices are labeled & the numbers corresponding to each slice is also represented in the chart.

In R the pie chart is created using the pie() function which takes positive numbers as a vector input. The additional parameters are used to control labels, color, title etc.

The basic syntax for creating a pie-chart using R is -

```
pie(x, labels, radius, main, col, clockwise)
```

A very simple pie-chart is created using just the input vector & labels. The below script will create & save the pie chart in the current R working directory.

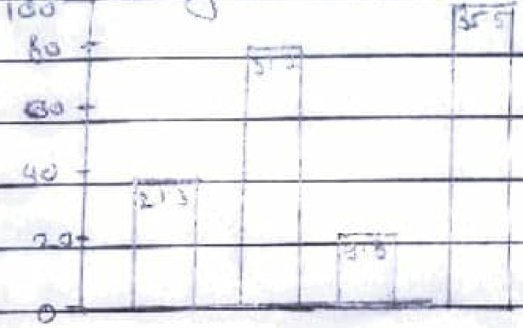


2. Bar chart :- A bar chart represents data in rectangular bars with length of the bar. R can draw both vertical & horizontal bars in the bar chart. In bar each of the bars can be given different colors. The basic syntax to create a bar-chart in R is -

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

Following is the description of the parameters used -

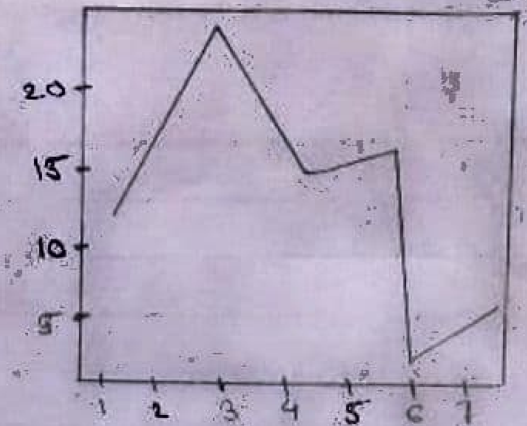
- H is a vector or matrix containing numeric values used in bar chart.
- xlab is the lable for x axis.
- ylab is the label for y axis.
- col is used to give colors to the bars in the graph.



5. Line graph :- A line chart is a graph that connects a series of point by drawing line segments between them. These points are ordered in one of their coordinate (usually the x-coordinate) value. Line charts. The plot() function in R is used to create the line graph.

The basic syntax to create a line chart in R is -

plot (v, type, col, xlab, ylab)



6. Scatterplots :- Scatterplots show many points plotted in the cartesian plane Each point represents the values of two variables one variable is chosen in the horizontal axis & another in the vertical axis. The simple scatterplot is created using the plot () function.

The basic syntax for creating scatterplot in R is -
plot (x, y, main, xlab, ylab, xlim, ylim, axes)

Following is the description of the parameter used -

- x is the data set whose value are the horizontal coordinats
- y is the data set whose value are the vertical coordinates.
- main is the tile of the graph.
- X lab is the label in the horizontal axis.
- ylim is the limits of the values of y used for plotting
- axes indicates whether both axes should be drawn on the plot.

C. Control structure in R :- As the name suggest, a control structure 'controls' the flow of code / commands written inside a function is a set of multiple commands written to automate a repetitive coding task.

1. if, else : This structure is used to test a condition. Below is the syntax :

```
if (condition) {
        # do something          } else {
                                    # do somethine else
                                }
```

2. vectorization with ifelse:

```
ifelse(x<=10, "x less than 10", "x greater than 10")
```

3. Other valid ways of writing if/else:

```
if (sample (x, 1) < 10) {
        y <- 5}
         else { y <- 0
                   }

        y <- if (sample(x, 1) <10)
      {
         } else{

              0}
```

4. For: This structure is used when a loop is to be executed fixed number of times. It is commonly use for iterating over the elements of an object (list, vector). Below is syntax: for (<search condition>) {# do something}

eg.

```
for (i in 1:10) { print (i) }
    x <- c ("apples", "oranges", "bananas", "strawberries")
  for (i in x) { print (x[i]) }
  for (i in 1:4) { print (x[i]) }
  for (i in seq (x)) { print (x[i]) }
  for (i in 1:4) print (x [i])
```

5. While:- It begins by testing a condition, & execute only if the condition is found to be true

```
    i <- 1
  while (i< 10) { print (i)
              i <- i +1) }
```

5. Break: A break statement is used inside a loop (repeat, for, while) to stop the iterations & flow the control outside of the loop. Below is syntax:

```
  break.
```

Past - Lab :- Student will be able to execute various R commands use control structures on R-tool & R studio for their application.

conclusion :- Thus excised Basic syntax, Data types, variable, operators, vectors, Lists, matrices, Data frames, Factors, various types of graph & the control structures taking suitable example.