Working with R Data types

Aim

To write a program to working with R data types in R studio.

Algorithm

1. There are several datatypes in R.
2. We have use Logical, Numeric, Real, Integer, Complex**,** Character, Raw, List, Vector, Matrix, Array, Factor, Data frame.
3. Using these data types, we had implemented some examples .

Source Code

```
#Logical data type
bool1 <- TRUE
print(bool1)
print(class(bool1))
bool2 <- FALSE
print(bool2)
print(class(bool2))
#Numeric data type

# floating point values
weight <- 63.5
print(weight)
print(class(weight))

# Real numbers
height <- 182
print(height)
print(class(height))
#Integer datatype
integer_variable<- 186L
print(class(integer_variable))
# Complex data type
complex_value<- 3 + 2i

# print class of complex_value
```

```
print(class(complex_value))
complex_value


#Character data type
# Create a string variable

fruit <- "Apple"
print(class(fruit))
# Create a character variable
my_char<- 'A'
print(class(my_char))

#Raw data type
# convert character to raw

raw_variable<- charToRaw("Welcome to r program")
print(raw_variable)
print(class(raw_variable))
# convert raw to character
char_variable<- rawToChar(raw_variable)
print(char_variable)
print(class(char_variable))
# Create a vector.
apple <- c('red','green',"yellow")
print(apple)
# Get the class of the vector.
print(class(apple))
# Create a list.
list1 <- list(c(2,5,3),21.3,sin)
# Print the list.
print(list1)
# Create a matrix.
M = matrix( c('a','a','b','c','b','a'), nrow = 2, ncol = 3, byrow = TRUE)
print(M)
# Create an array.
a <- array(c('green','yellow'),dim = c(3,3,2))
print(a)
```

```
# 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))

# 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)
```

OUTPUT:

```
[1] TRUE
[1] "logical"
[1] FALSE
[1] "logical"
[1] 63.5
[1] "numeric"
[1] 182
[1] "numeric"
[1] "integer"
[1] "complex"
[1]"character"
[1]"character"
 [1] 57 65 6c 63 6f 6d 65 20 74 6f 20 72 20 70 72 6f 67 72 61 6d
[1] "raw"
[1] "Welcome to r program"
[1] "character"
[1] "red"    "green" "yellow"
[1] "character"
[[1]]
[1] 2 5 3
[[2]]
[1] 21.3
[[3]]
function (x) .Primitive("sin")
    [,1] [,2] [,3]
[1,] "a"  "a""b"
[2,] "c"  "b""a"
, , 1
    [,1]    [,2]    [,3]
[1,] "green" "yellow" "green"
[2,] "yellow" "green" "yellow"
[3,] "green" "yellow" "green"
```

```
, , 2
[,1]    [,2]    [,3]
[1,] "yellow" "green" "yellow"
[2,] "green" "yellow" "green"
[3,] "yellow" "green" "yellow"
[1] green  greenyellowred  red    red    green
Levels: green redyellow
[1] 3
  gender height weight Age 1
Male152.0       81 42
2   Male171.5  93  38
3 Female165.0    78 26
```

Result:

Thus, the above program was executed successfully and the output was verified.

Data import and export

Aim

To write a program to data import and export in R studio.

Algorithm

1. Install the “readxl” package using **install.packages()** command.
2. Then load the package using library(“readxl”)
3. Store the excel value to excel variable using read_xlsx() methed.
4. Print the variable excel.
5. Import the csv file to data variable using read.csv() method.
6. Print the variable data.
7. Install the “openxlsx” package using **install.packages()** command.
8. Then load the package using library(“openxlsx”)
9. Create s.no,fruits,shop1,shop2 value using c() method.
10. Frame the all data using data.frame() methed and store in rate object.
11. Then write the dataset in specified location using write.xlsx(rate,file=”d://”)

Source Code

#Data import

```
install.packages("readxl")

library("readxl")

excel<-read_xlsx("student.xlsx",sheet=1)
excel
```

#csv file

```
data<-read.csv("stud_1.csv")
data
```

OUTPUT:

```
idname          englishmathsscience

<dbl><chr>    <dbl><dbl>     <dbl>

1    1anu              90      98         89

2    2banu             87      99         92

3    3mani             89     100         90

4    4kavi             90      89         98

5    5hari             91      90         96

6    6janu             93      92         98
```

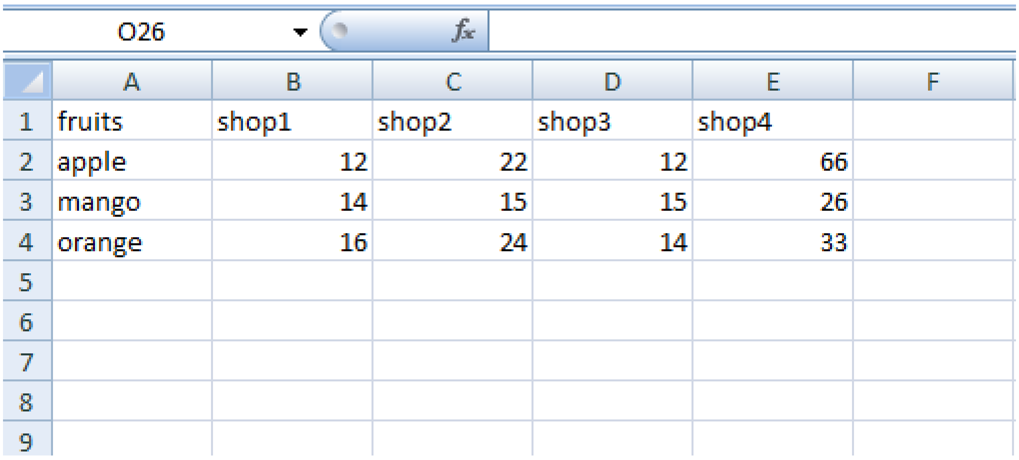
```
idname          englishmathsscience

<dbl><chr>    <dbl><dbl>     <dbl>

1    1venu             90      98         89

2    2guru             87      99         92

3    3jemi             89     100         90
```

#Data export

```
install.packages("openxlsx")

library(openxlsx)

s.no <- seq(1,3,by=1)

fruits <- c("apple","mango","orange")
shop1 <-c(12,14,16)
shop2 <- c(22,15,24)
shop3 <- c(12,15,14)
shop4 <-c(66,26,33)
rate <- data.frame(fruits,shop1,shop2,shop3,shop4)
rate
write.xlsx(rate,file="G:/R/fruits.xlsx")
```

OUTPUT:

```
 fruits shop1 shop2 shop3 shop4

 1apple      12       22       12       66

2mango       14       15       15       26

3orange      16       24       14       33
```



	A	B	C	D	E	F
1	fruits	shop1	shop2	shop3	shop4	
2	apple	12	22	12	66	
3	mango	14	15	15	26	
4	orange	16	24	14	33	
5						
6						
7						
8						
9						

Result:

Thus, the above program was executed successfully and the output was verified.

Ex. No.
Date:

Plots and Graphs

Page No:

Aim

To write a program for working with plots and graphs in R studio.

Algorithm

1. Initialize x variable and store the numbers using c() function.
2. Using barplot(x) function to display the barplot.
3. Initialize y variable and store the numbers using c() function.
4. Put the name for particular value using name() method.
5. Display the pie chart using pie(y) method.
6. Set the z value that take from Orange predefined dataset.
7. Create scatter plot using the plot() method.

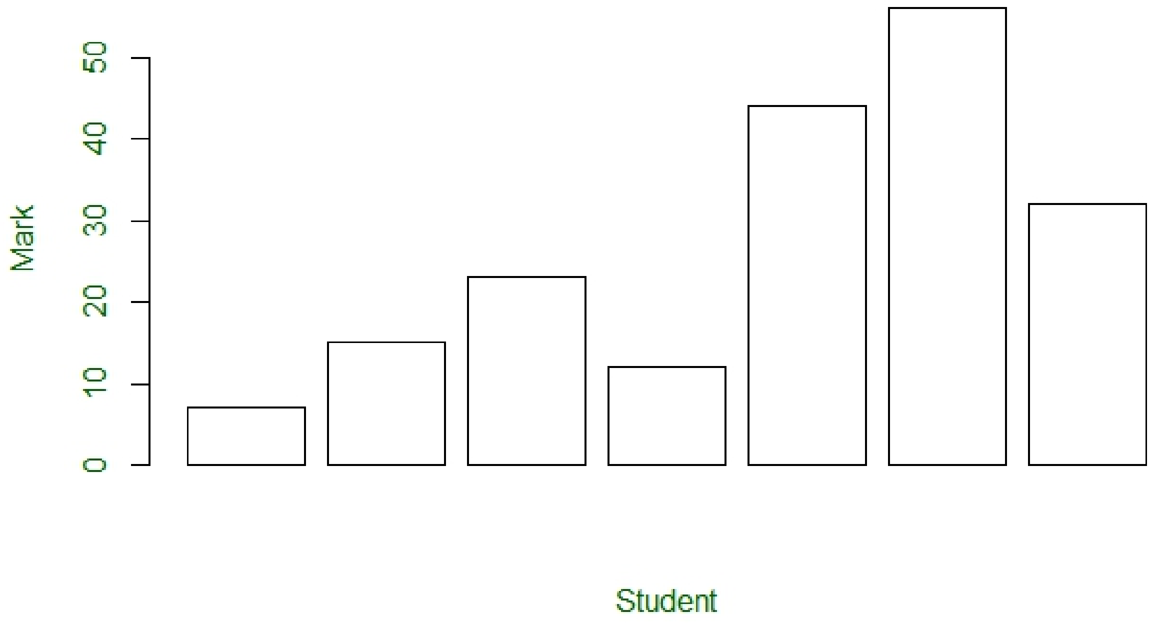
Source Code

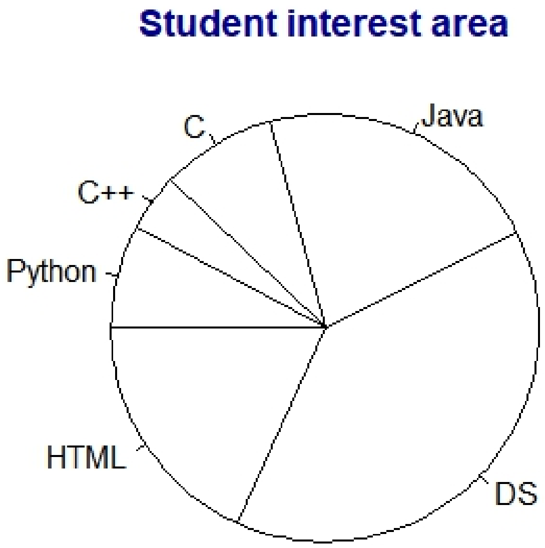
```
# Bar Plot or Bar Chart

x <- c(7, 15, 23, 12, 44, 56, 32)

barplot(x, xlab = "Student",

ylab = "Mark", col = "white",

col.axis = "darkgreen",

col.lab = "darkgreen")

# Pie Diagram or Pie Chart

y <- c(210, 450, 250, 100, 50, 90)

names(y) <- c("HTML", "DS", "Java", "C", "C++", "Python")

pie(y, labels = names(y), col = "white",

   main = "Student interest area", radius = -1,

col.main = "darkblue")
```

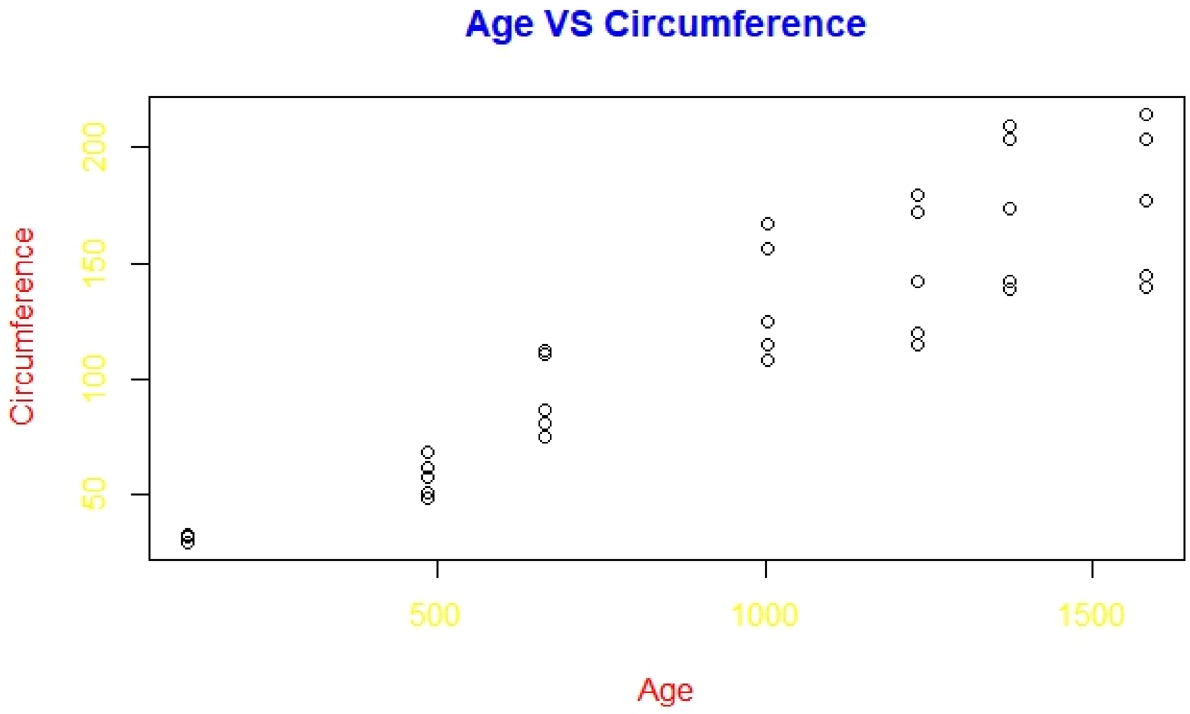
#Scatter Plot

```
z <- Orange[, c('age', 'circumference')]

plot(x = z$age, y = z$circumference, xlab = "Age",

ylab = "Circumference", main = "Age VS Circumference",

col.lab = "red", col.main = "blue",

col.axis = "yellow")
```

Output







Result:

Thus, the above program was executed successfully and the output was verified.