BDA

PRACTICAL ASSIGNMENT-1

Page No.
Date

AIM:-

Demonstrate the concept of variables and data types in R and create vectors, Matrices, Lists, Arrays, factors and data frames using it.

THEORY:-

Basic data types in R can be divided into the following types:-

(1) Numeric:- (10.5, 55, 787)

(2) Integer:- (1L, 55L, 100L, where the letter 'L' declares this as an integer)

(3) Complex:- (9+3i, where 'i' is the imaginary part)

(4) Character (a.k.a string):- ("K", "R is exciting", "FALSE", "11.5")

(5) Logical (a.k.a boolean):- (TRUE or FALSE)

class() function is used to check the data type of a variable.

→ Variables:-

Variables are containers for storing data values.
R doesnot have a command for declaring variable. It is created the moment you first assign a value to it.

To assign a value to a variable, use the '<-' sign. To output (or print) the variable value, type the variable name.

eg:
```
name <- "John"
age <- 40

name    # Output "John"
age     # Output 40
```

→ R Data Structures:-

(1) Vectors:-

A vector is simply a list of items that are of same type.
To combine the list of items to a vector, use the c() function and separate the items by comma.

eg:-
```
# Vector of strings
fruits <- c("banana", "apple", "orange")

# Print fruits

fruits
Vec <- ("A", "B", "C")
vec1 <- c("D", "E", "F")
vec2 <- c(vec1, vec)

vec2  # Output "D" "E" "F" "A" "B" "C"
```

(2) Lists :-

A list in R contain many different data types inside it.
A list is a collection of data which is ordered and changeable.
To create a list, use the list() function.

eg:-
```
list1 <- list("Hello", "World")
list1 # Output "Hello" "World"

list2 <- list(1, 2)
list3 <- c(list1, list2)
list3 # Output "Hello" "World" 1 2
```

(3) Matrices :-

A matrix is a two dimensional data set with columns and rows.
A column is a vertical representation of data, while a row is a horizontal representation of data.
A matrix can be created with the matrix() function.
Specify the nrow and ncol parameters to get the amount of rows and columns

Use the cbind() functions to add additional columns in a matrix.
Use the rbind() function to add additional rows in a Matrix.

eg:- mat ← matrix(c(1:4), nrow = 2, ncol = 2)
mat # Output $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$

mat1 ← matrix(c(5:8), nrow = 2, ncol = 2)
mat2 ← cbind(mat, mat1)
mat2 # Output $\begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$

mat3 ← rbind(mat, mat1)
mat3 # Output $\begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 5 & 7 \\ 6 & 8 \end{bmatrix}$

(4) Arrays:-

Compared to matrices, arrays can have more than ~~one~~ two dimensions.
We can use array() function to create an array, and the dim parameter to specify the dimensions

eg:- arr ← array(c(1:10), dim = c(2, 2, 2))
(columns, rows, dimension (no. of matrices))
arr
Output

$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$ $\begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$

(5) Data Frames:-

Data Frames are data displayed in a format as a table.
Data frames can have different types of data inside it. While the first column can be character, the second and third can be numeric or logical. However, each column should have the same type of data.
Use the data.frame() function to create a data frame.

eg:-
```
data_frame <- data.frame (
                id = c (1, 2, 3),
              name = c ("X", "Y", "Z"),
              dept = c ("AB", "CD", "EF")
              )

data_frame

# Output
```

id	name	dept
1	X	AB
2	Y	CD
3	Z	EF

(6) Factors:-

Factors are used to categorize data.
Example of factors are:

- Demography : Male / Female

- Music : Rock, Pop, Jazz
- Training : Strength, Stamina

To create a factor, use the factor() function and add a vector as argument

eg:- num ← c(1,1,2,3,2,4,5,6,8,8,9)
fact ← factor(num)
fact

Output

[1] 1 1 2 3 2 4 5 6 8 8 9
Levels: 1 2 3 4 5 6 8 9

is.factor() function is used to check if the vector is categorized or not.

eg:- check ← is.factor(num)
check # Output FALSE

check ← is.factor(fact)
check # Output TRUE