Task 1

Creating your own data using r programming for given following data sets:

By mauzumshamil

20bsc132

R has the following basic data structures:

1. Vector
2. List
3. Matrix
4. Data frame
5. Array
6. Factor

1} vectors;

Vectors are single-dimensional, homogeneous data structures. To create a vector, use the c() function.

Example:

> numeric\_vec <- c(1,2,3,4,5)

> integer\_vec <- c(1L,2L,3L,4L,5L)

> logical\_vec <- c(TRUE, TRUE, FALSE, FALSE, FALSE)

> complex\_vec <- c(12+2i, 3i, 4+1i, 5+12i, 6i)

> character\_vec <- c("techvidvan", "this", "is", "a", "character vector")

> numeric\_vec

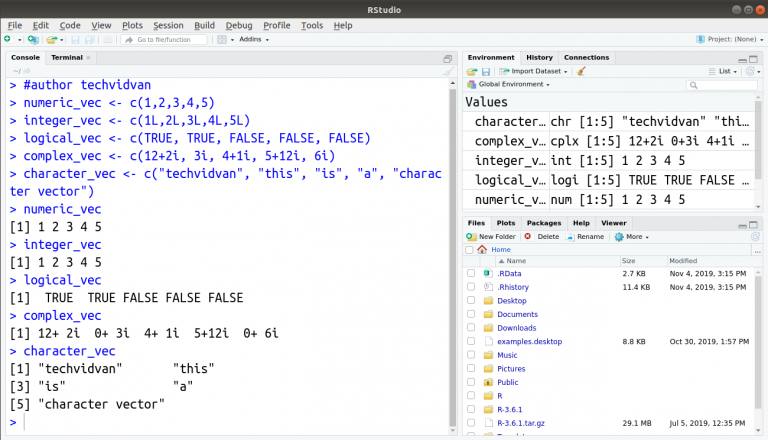
> integer\_vec

> logical\_vec

> complex\_vec

> character\_vec

Output:



2} Lists:

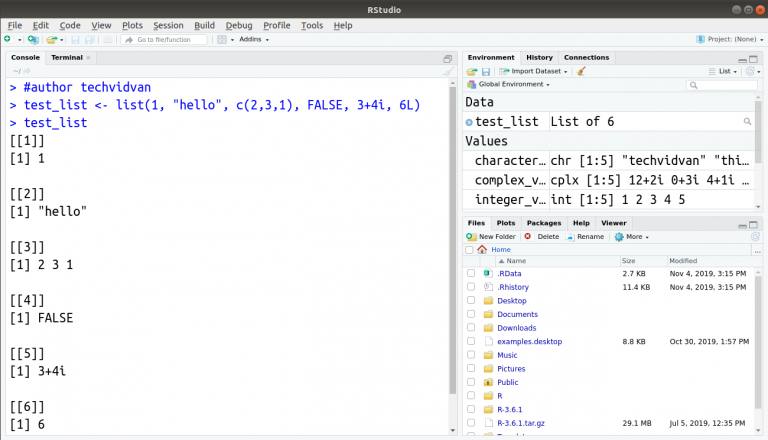
Lists are heterogeneous data structures. They are very similar to vectors except they can store data of different types. To create a list, we use the list() function.

Example:

> test\_list <- list(1, "hello", c(2,3,1), FALSE, 3+4i, 6L)

> test\_list

Outputs:



3} matrix

Matrices are two-dimensional, homogeneous data structures. This means that all values in a matrix have to be of the same type. Coercion takes place if there is more than one data type. They have rows and columns.

Example:

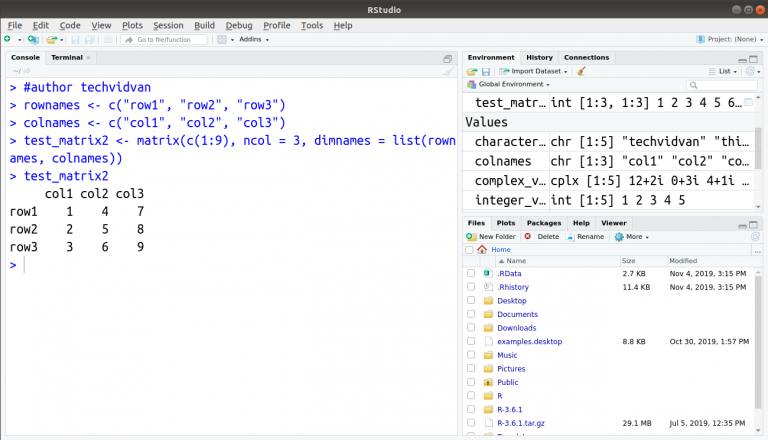
> rownames <- c("row1", "row2", "row3")

> colnames <- c("col1", "col2", "col3")

> test\_matrix2 <- matrix(c(1:9), ncol = 3, dimnames = list(rownames, colnames))

> test\_matrix2

Output:



4} data frame

Data frames are two-dimensional, heterogeneous data structures. They are lists of vectors of equal lengths

Example:

> student\_id <- c(1:5)

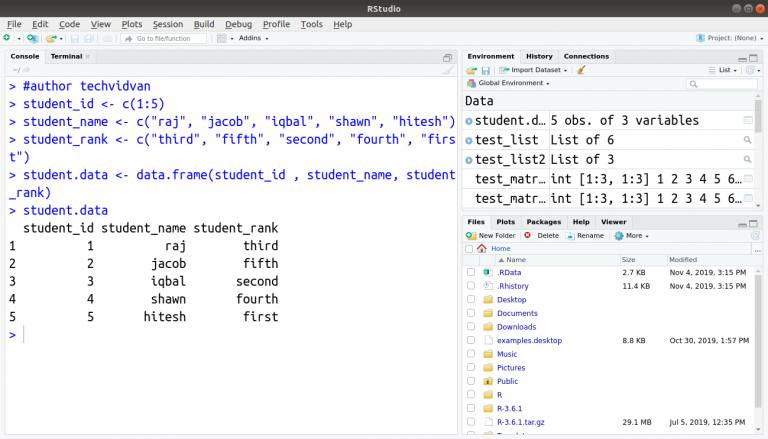
> student\_name <- c("raj", "jacob", "iqbal", "shawn", "hitesh")

> student\_rank <- c("third", "fifth", "second", "fourth", "first")

> student.data <- data.frame(student\_id , student\_name, student\_rank)

> student.data

Output:



5} arrays

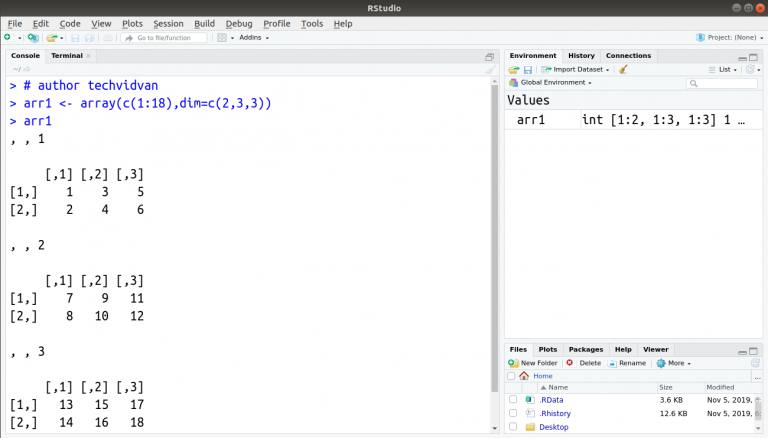
Arrays are three dimensional, homogeneous data structures. They are collections of matrices stacked one on top of the other in layers.

Example:

> arr1 <- array(c(1:18),dim=c(2,3,3))

> arr1

Output:



6}factors

Factors are vectors that can only store predefined values. They are useful for storing categorical data

Example:

> fac <- factor(c("a", "b", "a", "b", "b"))

> fac

output: