1

February 1, 2024

1 Explore Basic Data Structure in R.

In R, basic data structures are essential for organizing and manipulating data. The primary data structures in R include vectors, matrices, arrays, lists, data frames, and factors.

1. Vectors:

Vectors are one-dimensional arrays that can hold elements of the same data type. They can be created using the c() function.

[1]:
```
# Creating a numeric vector
numeric_vector <- c(1, 2, 3, 4, 5)
print(numeric_vector)

# Creating a character vector
character_vector <- c("apple", "orange", "banana")
print(character_vector)
```

```
[1] 1 2 3 4 5
[1] "apple"  "orange" "banana"
```

2. Matrices:

Matrices are two-dimensional arrays with rows and columns. They can be created using the matrix() function.

[3]:
```
# Creating a matrix
mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
print(mat)
```

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

3. Arrays:

Arrays are multi-dimensional extensions of matrices. They can be created using the array() function.

[4]:
```
# Creating a 3-dimensional array
arr <- array(1:24, dim = c(2, 3, 4))
print(arr)
```

```
, , 1

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

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

, , 3

     [,1] [,2] [,3]
[1,]   13   15   17
[2,]   14   16   18

, , 4

     [,1] [,2] [,3]
[1,]   19   21   23
[2,]   20   22   24
```

4. Lists:

Lists can contain elements of different data types, and each element can be a vector, matrix, or even another list. They can be created using the list() function.

[5]:
```
# Creating a list
my_list <- list(numeric_vector, character_vector, mat)
print(my_list)
```

```
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "apple"  "orange" "banana"

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

5. Data Frames:

Data frames are used to store tabular data, where columns can be of different data types. They can be created using the data.frame() function.

[6]:
```r
# Creating a data frame
df <- data.frame(Name = c("Alice", "Bob", "Charlie"),
                 Age = c(25, 30, 22),
                 Score = c(95, 88, 75))
print(df)
```

```
     Name Age Score
1   Alice  25    95
2     Bob  30    88
3 Charlie  22    75
```

6. Factors:

Factors are used to represent categorical data and can be ordered or unordered. They can be created using the factor() function.

[7]:
```r
# Creating a factor
gender <- factor(c("Male", "Female", "Male", "Female"))
print(gender)
```

```
[1] Male   Female Male   Female
Levels: Female Male
```