



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

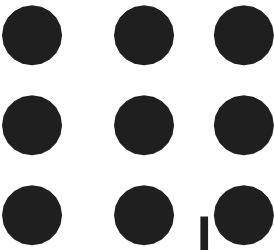
Department of Artificial Intelligence and Data Science

**Course Name – Big Data Analytics
III Year / V Semester**

Unit 2 – Data Science using Python

Topic - R Programming





R Programming

- R is a programming language and software environment for statistical analysis, graphics representation and reporting.
- R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R is freely available under the GNU General Public License.

Creating Variables in R

- Variables are containers for storing data values.
- R does not have a command for declaring a variable.
- A variable 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, just type the variable name:



R Programming



Example

```
name <- "John"  
age <- 40  
name # output "John"  
age # output 40
```

Basic Data Types

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

- numeric - (10.5, 55, 887)
- integer - (1L, 55L, 100L, where the letter "L" declares this as an integer)
- complex - (9 + 3i, where "i" is the imaginary part)
- character (a.k.a. string) - ("k", "R is exciting", "FALSE", "11.5")
- logical (a.k.a. boolean) - (TRUE or FALSE)



R Programming



Example

We can use the class() function to check the data type of a variable:

Example

```
# numeric
```

```
x <- 10.5
```

```
class(x)
```

```
# integer
```

```
x <- 1000L
```

```
class(x)
```

```
# complex
```

```
x <- 9i + 3
```

```
class(x)
```

```
# character/string
```

```
x <- "R is exciting"
```

```
class(x)
```

```
# logical/boolean
```

```
x <- TRUE
```

```
class(x)
```

Output

```
[1] "numeric"
```

```
[1] "integer"
```

```
[1] "complex"
```

```
[1] "character"
```

```
[1] "character"
```



R Programming



R Data Structure

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Data Frame

Vectors

- A vector is simply a list of items that are of the same type.
- To combine the list of items to a vector, use the `c()` function and separate the items by a comma.
- In the example below, we create a vector variable called `fruits`, that combine strings:

Example

Vector of strings

```
fruits <- c("banana", "apple", "orange")
```

```
# Print fruits
```

```
fruits
```



R Programming



R Data Structure

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Data Frame

Vectors

- A vector is simply a list of items that are of the same type.
- To combine the list of items to a vector, use the `c()` function and separate the items by a comma.
- In the example below, we create a vector variable called `fruits`, that combine strings:

Example

Vector of strings

```
fruits <- c("banana", "apple", "orange")  
# Print fruits  
fruits
```

Vector of numerical values

```
numbers <- c(1, 2, 3)  
# Print numbers  
numbers
```



R Programming

Lists

A list in R can 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:

Example

```
# List of strings  
thislist <- list("apple", "banana", "cherry")  
# Print the list  
Thislist
```

Access Lists

You can access the list items by referring to its index number, inside brackets. The first item has index 1, the second item has index 2, and so on:

Example

```
thislist <- list("apple", "banana", "cherry")  
thislist[1]
```



R Programming



Change Item Value

To change the value of a specific item, refer to the index number:

Example

```
thislist <- list("apple", "banana", "cherry")  
thislist[1] <- "blackcurrant"  
# Print the updated list  
Thislist
```

List Length

To find out how many items a list has, use the length() function:

Example

```
thislist <- list("apple", "banana", "cherry")  
length(thislist)
```




THANK YOU