

Module-1

Geethu Gopinath, Statistics Department

What are data types in R?

There are many basic data types in R, which are of frequent occurrence in coding R calculations and programs.

- Numeric
- Integer
- Complex
- Logical
- Character

Numeric Data Type

Decimal values are referred to as numeric data types in R. This is the default working out data type. If you assign a decimal value for any variable x like given below, x will become a numeric type.

```
> g = 62.4      # assign a decimal value to g
> g            # print the variable's value - g
```

Integer Data Type

If you want to create an integer variable in R, you have to invoke the `as.integer()` function to define any integer type data.

```
> s = as.integer(3)
> s          # print the value of s
```

Complex Data Type

A complex value for coding in R can be defined using the pure imaginary values 'i'.

```
> k = 1 + 2i    # creating a complex number
> k            # printing the value of k
```

Module-1

Geethu Gopinath, Statistics Department

Logical Data Type

A logical value is mostly created when a comparison between variables are done.

```
> a = 4; b = 6           # sample values
> g = a > b             # is a larger than b?
> g                     # print the logical value
```

Output:

```
[1] False
```

Character Data Type

A character object can be used for representing string values in R. You have to convert objects into character values using the `as.character()` function within your code like this:

```
> g = as.character(62.48)
> g           # prints the character string
```

Output:

```
[1] "3.14"
```

```
> class(s)      # print the class name of s
```

Output:

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

What are operators in R?

The operators are those symbols which tell the compiler for performing precise mathematical or logical manipulations. R programming is loaded with built in operator and various types are:

Types of Operators

- The Arithmetic Operators
- The Relational Operators
- The Logical Operators
- The Assignment Operators



Module-1

Geethu Gopinath, Statistics Department

Arithmetic Operators

Addition

```
g <- c (4, 6.5, 6)
s <- c (8, 3, 5)
print (g + s)
```

Subtraction

```
g <- c (2, 5.5, 6)
s <- c (8, 3, 4)
print (g - s)
```

Multiplication

```
g <- c ( 26.5,8)
s <- c(6, 4, 3)
print (g * s)
```

Division

```
g <- c( 2,4.6,8)
s <- c(8, 4, 3)
print (g / s)
```

Modulus

```
g <- c ( 2, 5.5, 8)
s <- c (8, 4, 5)
print (g% / %s)
```

Relational Operators

Greater than

```
g <- c (2,5.5,6,9)
s <- c (8,2.5,14,9)
print (g > s)
```

Less than

```
g <- c (2, 5.6, 6,9)
s <- c(8,2.5,14,9)
print (g < s)
```



Module-1

Geethu Gopinath, Statistics Department

Equals Operators

```
g <- c (2,5.5,6,9)
s <- c (8,2.5,14,9)
print (g == s)
```

Less than or equal

```
g <- c (2, 5.5, 6, 9)
s <- c (8, 2.5, 14, 9)
print (g <= s)
```

Greater than or equal

```
g <- c(2,5.5,6,9)
s <- c(8, 2.5, 14, 9)
print(g>=s)
```

Not Equal

```
g <- c(2, 5.4, 8, 9)
s <- c(8, 2.5, 14, 8)
print(g!=s)
```

Logical Operators

Element Wise-Logical AND Operator

```
g <- c(3, 1, TRUE, 2+3i)
s <- c(4,1,FALSE, 2+3i)
print (g & s)
```

It unites each element of the 1st vector with the equivalent element of the 2nd vector and returns TRUE or FALSE.

Element Wise-Logical OR Operator

```
g <- c(3,0, TRUE, 2+2i)
s <- c(4,0, FALSE, 2+3i)
print (g | s)
```

It unites each element of the 1st vector with the equivalent element of the 2nd vector.

Logical NOT Operator

```
k <- c (3,0, TRUE, 2+2i)
```



Module-1

Geethu Gopinath, Statistics Department

```
print (!k)
```

Logical AND Operator

```
g <- c(3,0,TRUE,2+2i)
s <- c(1,3,TRUE,2+3i)
print (g && s)
```

Logical OR Operator

```
g <- c (0,0,TRUE,2+2i)
s <- c (0,3,TRUE,2+3i)
print (g||s)
```

Assignment Operators

There are three types of operators used for assigning values to vectors.

```
g1 <- c (2,1,TRUE, 2+3i)
g2 <<- c (2,1,TRUE, 2+3i)
g3 = c (2,1, TRUE, 2+3i)
print (g1)
print (g2)
print (g3)
```