## ****Arithmetic Operators****

Arithmetic operations in R simulate various math operations, like addition, subtraction, multiplication, division, and modulo using the specified operator between operands, which may be either scalar values, complex numbers, or vectors. The R operators are performed element-wise at the corresponding positions of the vectors.

### ****Addition operator (+)****

The values at the corresponding positions of both operands are added. Consider the following R operator snippet to add two vectors:

# R program to illustrate

# the use of Arithmetic operators

vec1 <- c(0, 2)

vec2 <- c(2, 3)

# Performing operations on Operands

cat ("Addition of vectors :", vec1 + vec2, "\n")

cat ("Subtraction of vectors :", vec1 - vec2, "\n")

cat ("Multiplication of vectors :", vec1 \* vec2, "\n")

cat ("Division of vectors :", vec1 / vec2, "\n")

cat ("Modulo of vectors :", vec1 %% vec2, "\n")

cat ("Power operator :", vec1 ^ vec2)

**Output**

Addition of vectors : 2 5

Subtraction of vectors : -2 -1

Multiplication of vectors : 0 6

Division of vectors : 0 0.6666667

Modulo of vectors : 0 2

Power operator : 0 8

**Logical Operators**

Logical operations in R simulate element-wise decision operations, based on the specified operator between the operands, which are then evaluated to either a True or False boolean value. Any non-zero integer value is considered as a TRUE value, be it a complex or real number.

**Element-wise Logical AND operator (&)**

Returns True if both the operands are True.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1)  list2 <- c(0,4+3i)  print(list1 & list2) |

**Output : FALSE TRUE**

Any non zero integer value is considered as a TRUE value, be it complex or real number.

**Element-wise Logical OR operator (|)**

Returns True if either of the operands is True.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1)  list2 <- c(0,4+3i)  print(list1|list2) |

**Output : TRUE TRUE**

**NOT operator (!)**

A unary operator that negates the status of the elements of the operand.

* R

|  |
| --- |
| list1 <- c(0,**FALSE**)  print(!list1) |

**Output : TRUE TRUE**

**Logical AND operator (&&)**

Returns True if both the first elements of the operands are True.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1)  list2 <- c(0,4+3i)  print(list1 && list2) |

**Output : FALSE**

Compares just the first elements of both the lists.

**Logical OR operator (||)**

Returns True if either of the first elements of the operands is True.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1)  list2 <- c(0,4+3i)  print(list1||list2) |

**Output : TRUE**

The following R code illustrates the usage of all Logical Operators in R:

* R

|  |
| --- |
| # R program to illustrate  # the use of Logical operators  vec1 <- c(0,2)  vec2 <- c(**TRUE**,**FALSE**)    # Performing operations on Operands  cat ("Element wise AND :", vec1 & vec2, "\n")  cat ("Element wise OR :", vec1 | vec2, "\n")  cat ("Logical AND :", vec1 && vec2, "\n")  cat ("Logical OR :", vec1 || vec2, "\n")  cat ("Negation :", !vec1) |

**Output**

Element wise AND : FALSE FALSE

Element wise OR : TRUE TRUE

Logical AND : FALSE

Logical OR : TRUE

Negation : TRUE FALSE

**Relational Operators**

The relational operators in R carry out comparison operations between the corresponding elements of the operands. Returns a boolean TRUE value if the first operand satisfies the relation compared to the second. A TRUE value is always considered to be greater than the FALSE.

**Less than (<)**

Returns TRUE if the corresponding element of the first operand is less than that of the second operand. Else returns FALSE.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1,"apple")  list2 <- c(0,0.1,"bat")  print(list1<list2) |

**Output : FALSE FALSE TRUE**

**Less than equal to (<=)**

Returns TRUE if the corresponding element of the first operand is less than or equal to that of the second operand. Else returns FALSE.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1, "apple")  list2 <- c(**TRUE**, 0.1, "bat")    # Convert lists to character strings  list1\_char <- as.character(list1)  list2\_char <- as.character(list2)    # Compare character strings  print(list1\_char <= list2\_char) |

**Output : TRUE TRUE TRUE**

**Greater than (>)**

Returns TRUE if the corresponding element of the first operand is greater than that of the second operand. Else returns FALSE.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1, "apple")  list2 <- c(**TRUE**, 0.1, "bat")  print(list1\_char > list2\_char) |

**Output : FALSE FALSE FALSE**

**Greater than equal to (>=)**

Returns TRUE if the corresponding element of the first operand is greater or equal to that of the second operand. Else returns FALSE.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1, "apple")  list2 <- c(**TRUE**, 0.1, "bat")  print(list1\_char >= list2\_char) |

**Output : TRUE TRUE FALSE**

**Not equal to (!=)**

Returns TRUE if the corresponding element of the first operand is not equal to the second operand. Else returns FALSE.

* R

|  |
| --- |
| list1 <- c(**TRUE**, 0.1,'apple')  list2 <- c(0,0.1,"bat")  print(list1!=list2) |

**Output : TRUE FALSE TRUE**

The following R code illustrates the usage of all Relational Operators in R:

* R

|  |
| --- |
| # R program to illustrate  # the use of Relational operators  vec1 <- c(0, 2)  vec2 <- c(2, 3)    # Performing operations on Operands  cat ("Vector1 less than Vector2 :", vec1 < vec2, "\n")  cat ("Vector1 less than equal to Vector2 :", vec1 <= vec2, "\n")  cat ("Vector1 greater than Vector2 :", vec1 > vec2, "\n")  cat ("Vector1 greater than equal to Vector2 :", vec1 >= vec2, "\n")  cat ("Vector1 not equal to Vector2 :", vec1 != vec2, "\n") |

**Output**

Vector1 less than Vector2 : TRUE TRUE

Vector1 less than equal to Vector2 : TRUE TRUE

Vector1 greater than Vector2 : FALSE FALSE

Vector1 greater than equal to Vector2 : FALSE FALSE

Vector1 not equal to Vector2 : TRUE TRUE

**Assignment Operators**

Assignment operators in R are used to assigning values to various data objects in R. The objects may be integers, vectors, or functions. These values are then stored by the assigned variable names. There are two kinds of assignment operators: Left and Right

**Left Assignment (<- or <<- or =)**

Assigns a value to a vector.

* R

|  |
| --- |
| vec1 = c("ab", **TRUE**)  print (vec1) |

**Output : "ab" "TRUE"**

**Right Assignment (-> or ->>)**

Assigns value to a vector.

* R

|  |
| --- |
| c("ab", **TRUE**) ->> vec1  print (vec1) |

**Output : "ab" "TRUE"**

The following R code illustrates the usage of all Relational Operators in R:

* R

|  |
| --- |
| # R program to illustrate  # the use of Assignment operators  vec1 <- c(2:5)  c(2:5) ->> vec2  vec3 <<- c(2:5)  vec4 = c(2:5)  c(2:5) -> vec5    # Performing operations on Operands  cat ("vector 1 :", vec1, "\n")  cat("vector 2 :", vec2, "\n")  cat ("vector 3 :", vec3, "\n")  cat("vector 4 :", vec4, "\n")  cat("vector 5 :", vec5) |

**Output**

vector 1 : 2 3 4 5

vector 2 : 2 3 4 5

vector 3 : 2 3 4 5

vector 4 : 2 3 4 5

vector 5 : 2 3 4 5

**Miscellaneous Operators**

These are the mixed operators in R that simulate the printing of sequences and assignment of vectors, either left or right-handed.

**%in% Operator**

Checks if an element belongs to a list and returns a boolean value TRUE if the value is present  else FALSE.

* R

|  |
| --- |
| val <- 0.1  list1 <- c(**TRUE**, 0.1,"apple")  print (val %**in**% list1) |

**Output : TRUE**

Checks for the value 0.1 in the specified list. It exists, therefore, prints TRUE.

**%\*% Operator**

This operator is used to multiply a matrix with its transpose. Transpose of the matrix is obtained by interchanging the rows to columns and columns to rows. The number of columns of the first matrix must be equal to the number of rows of the second matrix. Multiplication of the matrix A with its transpose, B, produces a square matrix. 

* R

|  |
| --- |
| mat = matrix(c(1,2,3,4,5,6),nrow=2,ncol=3)          print (mat)          print( t(mat))          pro = mat %\*% t(mat)          print(pro) |

**Input :**

**Output :[,1] [,2] [,3] #original matrix of order 2x3**

**[1,] 1 3 5**

**[2,] 2 4 6**

**[,1] [,2] #transposed matrix of order 3x2**

**[1,] 1 2**

**[2,] 3 4**

**[3,] 5 6**

**[,1] [,2] #product matrix of order 2x2**

**[1,] 35 44**

**[2,] 44 56**

The following R code illustrates the usage of all Miscellaneous Operators in R:

* R

|  |
| --- |
| # R program to illustrate  # the use of Miscellaneous operators  mat <- matrix (1:4, nrow = 1, ncol = 4)  print("Matrix elements using : ")  print(mat)    product = mat %\*% t(mat)  print("Product of matrices")  print(product,)  cat ("does 1 exist in prod matrix :", "1" %**in**% product) |

**Output**

[1] "Matrix elements using : "

[,1] [,2] [,3] [,4]

[1,] 1 2 3 4

[1] "Product of matrices"

[,1]

[1,] 30

does 1 exist in prod matrix : FALSE