

R Operators

Types of Operators 1 Arithmetic Operators 2 Relational Operators 3 Logical Operators 4 Assignment Operators 5 Miscellaneous Operators

Arithmetic Operators

'+' Adds two vectors

'-' Subtracts second vector from the first

'*' Multiplies both vectors elementwise

'/' Divide the first vector elements with the second vector elements elementwise

'%%' Give the remainder of the first vector with the second elementwise

'%/%' Quotient (integer part) of division of first vector with second elementwise

'^' The first vector elements raised to the exponent of second vector elements elementwise

```
x = c(15,22)
y = c(3,5)
x
```

```
## [1] 15 22
```

```
y
```

```
## [1] 3 5
```

```
cat("\n Addition :",x+y)
```

```
##
## Addition : 18 27
```

```
cat("\n Substraction :",x-y)
```

```
##
## Substraction : 12 17
```

```
cat("\n Multiplication :", x*y)
```

```
##
## Multiplication : 45 110
```

```
cat("\n Division :", x/y)
```

```
##  
## Division : 5 4.4
```

```
cat("\n Mod (remainder):",x%%y)
```

```
##  
## Mod (remainder): 0 2
```

```
cat("\n Quotient ( integer part) of division",x%/y)
```

```
##  
## Quotient ( integer part) of division 5 4
```

```
cat("\n Power operation:",x^y)
```

```
##  
## Power operation: 3375 5153632
```

Relational Operators

Following list shows the relational operators supported by R language.

Working: Each element of the first vector is compared with the corresponding element of the second vector. The result of comparison is a Boolean value.

1. '<' elementwise check if first vector element is less than second vector element and return TRUE or FALSE for every element
2. '>' elementwise check if first vector element is greater than second vector element and return TRUE or FALSE for every element
3. '==' elementwise check if first vector element is equal to second vector element and return TRUE or FALSE for every element
4. '!=' elementwise check if first vector element is not equal to second vector element and return TRUE or FALSE for every element
5. '>=' elementwise check if first vector element is greater than equal to second vector element and return TRUE or FALSE for every element
6. '<=' elementwise check if first vector element is less than equal to second vector element and return TRUE or FALSE for every element

```
x = c(15,22,35,100)  
y = c(3,5,56,100)  
cat("\n Less than :",x<y)
```

```
##  
## Less than :: FALSE FALSE TRUE FALSE
```

```
cat("\n Greater than::",x>y)
```

```
##  
## Greater than:: TRUE TRUE FALSE FALSE
```

```
cat("\n Equal ::", x==y)
```

```
##  
## Equal :: FALSE FALSE FALSE TRUE
```

```
cat("\n Not equal ::", x!=y)
```

```
##  
## Not equal :: TRUE TRUE TRUE FALSE
```

```
cat("\n Greater than equal::",x>=y)
```

```
##  
## Greater than equal:: TRUE TRUE FALSE TRUE
```

```
cat("\n Less than equal ::",x<=y)
```

```
##  
## Less than equal :: FALSE FALSE TRUE TRUE
```

Logical Operators

Following list shows the logical operators supported by R language. It is applicable only to vectors of type logical, numeric or complex. All numbers greater than 1 are considered as logical value TRUE.

Each element of the first vector is compared with the corresponding element of the second vector. The result of comparison is a Boolean value.

Element wise operators :: &, |, ! (Compares all elements one by one)

Non Zero values are TRUE [Negative nos are TRUE and positive nos are TRUE]

Zero is FALSE

```
v <- c(3,0,56.23,2+2i)
t <- c(0,3,TRUE,2+3i)
z = c(TRUE,FALSE,TRUE,FALSE)
v
```

```
## [1] 3.00+0i 0.00+0i 56.23+0i 2.00+2i
```

```
t
```

```
## [1] 0+0i 3+0i 1+0i 2+3i
```

```
z
```

```
## [1] TRUE FALSE TRUE FALSE
```

```
print(v&t)
```

```
## [1] FALSE FALSE TRUE TRUE
```

```
print(v&z)
```

```
## [1] TRUE FALSE TRUE FALSE
```

```
print(v|t)
```

```
## [1] TRUE TRUE TRUE TRUE
```

```
print(v|z)
```

```
## [1] TRUE FALSE TRUE TRUE
```

Special Logical Operators &&, || ONLY Compares only first element of vectors and return TRUE or FALSE

```
print(v&& t)
```

```
## Warning in v && t: 'length(x) = 4 > 1' in coercion to 'logical(1)'
```

```
## Warning in v && t: 'length(x) = 4 > 1' in coercion to 'logical(1)'
```

```
## [1] FALSE
```

```
print(v||t)
```

```
## Warning in v || t: 'length(x) = 4 > 1' in coercion to 'logical(1)'
```

```
## [1] TRUE
```

```
print(v&&z)
```

```
## Warning in v && z: 'length(x) = 4 > 1' in coercion to 'logical(1)'
```

```
## Warning in v && z: 'length(x) = 4 > 1' in coercion to 'logical(1)'
```

```
## [1] TRUE
```

```
print(v||z)
```

```
## Warning in v || z: 'length(x) = 4 > 1' in coercion to 'logical(1)'
```

```
## [1] TRUE
```

Working of logical operators on atomic vectors

Here both & and && also | and || return the same results

Non Zero values are TRUE [Negative nos are TRUE and positive nos are TRUE]

Zero is FALSE

```
x=10  
y=0  
print(x&y)
```

```
## [1] FALSE
```

```
print(x&&y)
```

```
## [1] FALSE
```

```
print(x|y)
```

```
## [1] TRUE
```

```
print(x||y)
```

```
## [1] TRUE
```

Assignment Operators <-, =, <<- :: left assignment ->, ->> :: right assignment

```
x = 10
y<-20
z<<-24

34-> n
35 ->> b
cat(x,y,z,n,b)
```

```
## 10 20 24 34 35
```

Miscellaneous Operators

These operators are used to for specific purpose and not general mathematical or logical computation.

Colon operator

operator = Colon operator. It creates the series of numbers in sequence and returns a vector. Each time step is +1 or -1 based on start and end values.

```
v1 = 2:8
v1
```

```
## [1] 2 3 4 5 6 7 8
```

```
v4 = 10:5
v4
```

```
## [1] 10 9 8 7 6 5
```

```
v3 = -3:-10
v3
```

```
## [1] -3 -4 -5 -6 -7 -8 -9 -10
```

```
v4 = -23:-15
v4
```

```
## [1] -23 -22 -21 -20 -19 -18 -17 -16 -15
```

```
v5 = 11.5:15.7  
v5
```

```
## [1] 11.5 12.5 13.5 14.5 15.5
```

%in% operator

%in% = This operator is used to identify if an element belongs to a vector. %in% returns a logical vector length same as the first vector, with a TRUE where that element can be found in the second argument and a FALSE where that element can not be found in second vector.

```
v1 <- 8  
v2 <- 15  
x=c(8,15)  
t <- c(9,12,45,8,67,5,8)  
print(v1 %in% t)
```

```
## [1] TRUE
```

```
print(v2 %in% t)
```

```
## [1] FALSE
```

```
print(t %in% v1)
```

```
## [1] FALSE FALSE FALSE TRUE FALSE FALSE TRUE
```

```
print(t %in% v2)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
x=c(8,15)  
t <- c(9,12,15,8,67,5,8)  
print(t %in% x)
```

```
## [1] FALSE FALSE TRUE TRUE FALSE FALSE TRUE
```

```
print(which(t %in% x))
```

```
## [1] 3 4 7
```

%*% operator

%*% = This operator is used to get dot product of two matrices

```
M = matrix( c(2,6,5,1,10,4), nrow=2,ncol=3)
M1 = matrix( c(20,60,50,1,10,4), nrow=3,ncol=2)
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    2    5   10
## [2,]    6    1    4
```

```
ans = M %*% M1 #size 2 by 2
print(ans)
```

```
##      [,1] [,2]
## [1,]  840   92
## [2,]  380   32
```