Foundations of R Software

Lecture 17

Basics of Calculations

::::

Relational and Logical Operators

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Relational Operators and Comparisons

The following table shows the operations and functions for relational comparisons (True or False).

TRUE and FALSE are reserved words denoting logical constants.

Operator	Executions
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
==	Exactly equal to
!=	Not equal to
!	Negation (not)

Zero is considered as

FALSE

and

non-zero numbers are

taken as TRUE

Logical Operators and Comparisons

Operator	Executions	
!	Logical not	
&	Element-wise logical and	Performs operation element-wise
&&	Logical and	examines only the first element of the operands
|	Element-wise logical or	Performs operation element-wise
||	Logical or	examines only the first element of the operands

Logical Operators and Comparisons

Operator	Executions
`xor(x,y)`	Either `x` or `y` (`x` or `y` but not `x` and `y`)
`isTRUE(x)` `isFALSE(x)`	test if `x` is TRUE test if `x` is FALSE
`TRUE` `FALSE`	true false

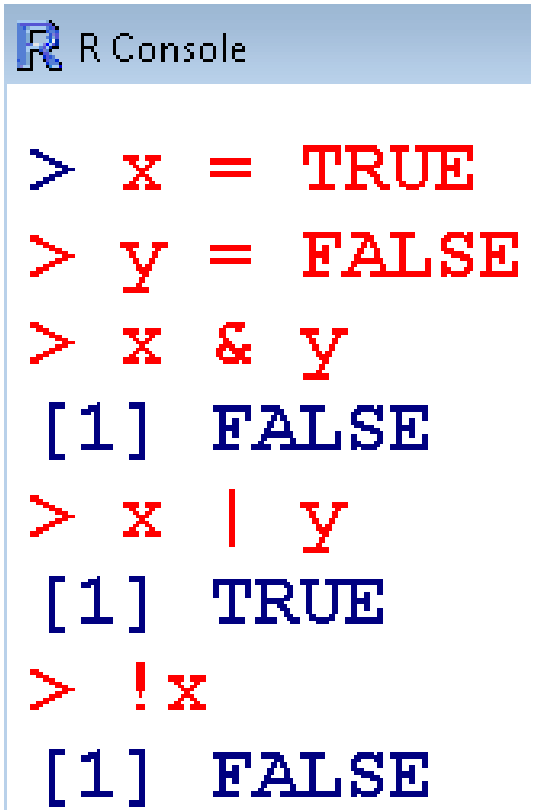
Example of Standard logical operations

Truth table

Statement 1 :: (x)	Statement 2 :: (y)	Outcome :: x and y	Outcome :: x or y
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Example of Standard logical operations

```
> x = TRUE
> y = FALSE

> x & y      # x AND y
[1] FALSE


> x | y      # x OR y
[1] TRUE


> !x         # negation of x
[1] FALSE
```



Example

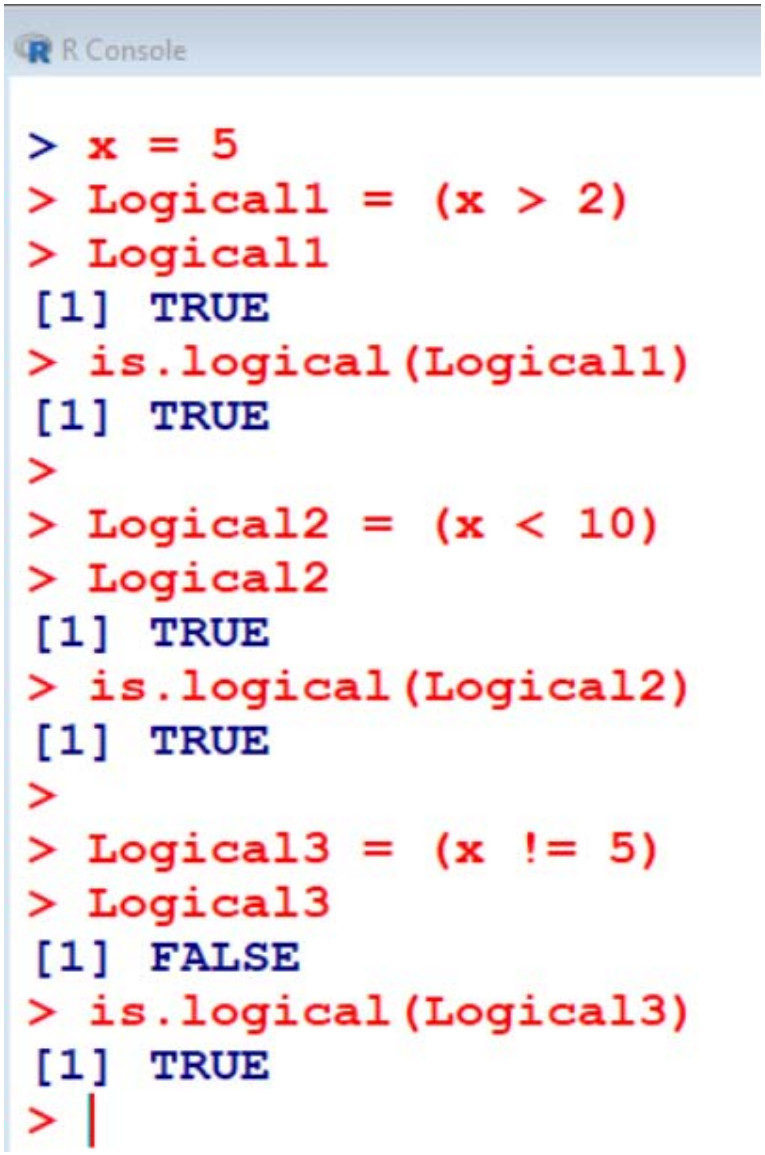
```
> x = 5
```

```
> Logical1 = (x > 2)
> Logical1
[1] TRUE
> is.logical(Logical1)
[1] TRUE
```

```
> Logical2 = (x < 10)
> Logical2
[1] TRUE
> is.logical(Logical2)
[1] TRUE
```

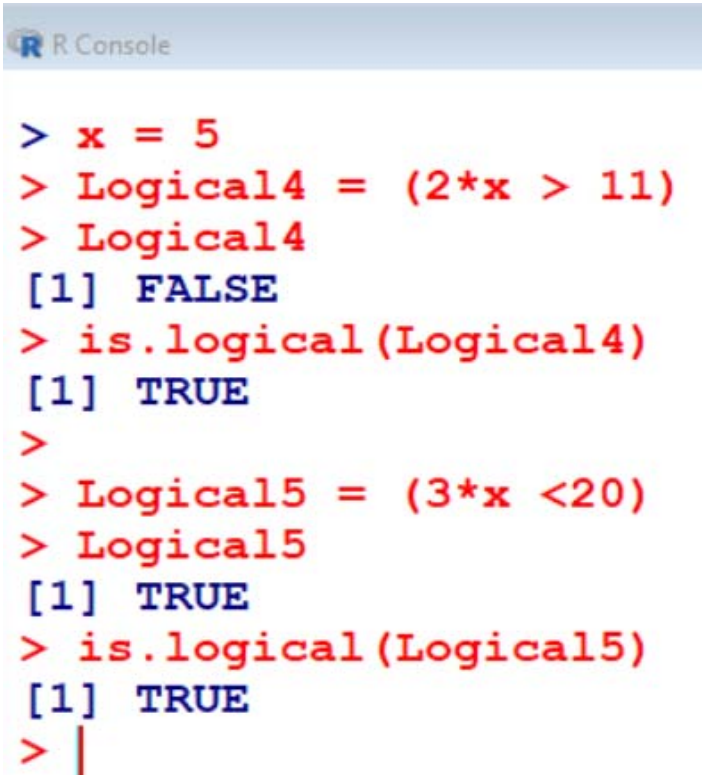
```
> Logical3 = (x != 5)
> Logical3
[1] FALSE
> is.logical(Logical3)
[1] TRUE
```

Example


R Console
> x = 5
> Logical1 = (x > 2)
> Logical1
[1] TRUE
> is.logical(Logical1)
[1] TRUE
>
> Logical2 = (x < 10)
> Logical2
[1] TRUE
> is.logical(Logical2)
[1] TRUE
>
> Logical3 = (x != 5)
> Logical3
[1] FALSE
> is.logical(Logical3)
[1] TRUE
>

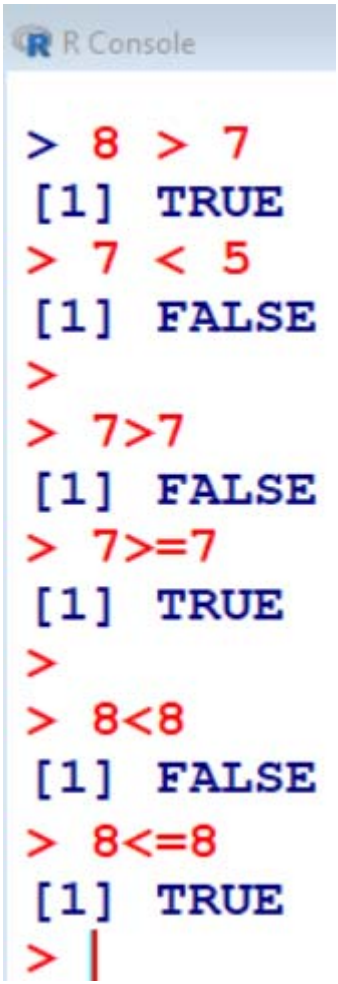
Example

```
> x = 5
> Logical4 = (2*x > 11)
> Logical4
[1] FALSE
> is.logical(Logical4)
[1] TRUE

> Logical5 = (3*x <20)
> Logical5
[1] TRUE
> is.logical(Logical5)
[1] TRUE
```



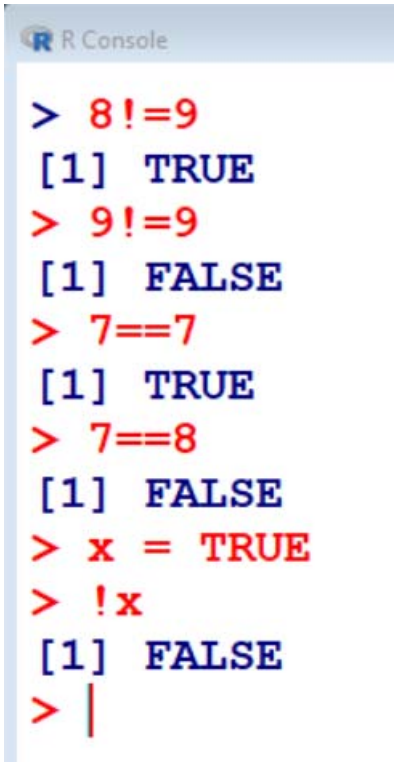
Examples:

```
> 8 > 7
[1] TRUE

> 7 < 5
[1] FALSE

> 7 > 7
[1] FALSE

> 7 >= 7
[1] TRUE

> 8 < 8
[1] FALSE

> 8 <= 8
[1] TRUE
```



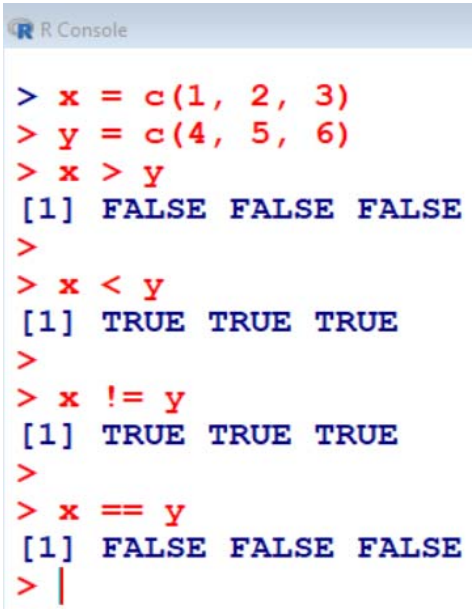
Examples:

```
> 8 != 9
[1] TRUE

> 9 != 9
[1] FALSE

> 7 == 7
[1] TRUE

> 7 == 8
[1] FALSE

> x = TRUE
> !x
[1] FALSE
```



Examples:

```
> x = c(1, 2, 3)
> y = c(4, 5, 6)

> x > y     # Compares 1 > 4, 2 > 5, 3 > 6
[1] FALSE FALSE FALSE

> x < y     # Compares 1 < 4, 2 < 5, 3 < 6
[1] TRUE TRUE TRUE

> x != y    # Compares 1 ≠ 4, 2 ≠ 5, 3 ≠ 6

[1] TRUE TRUE TRUE

> x == y    # Compares 1 = 4, 2 = 5, 3 = 6
[1] FALSE FALSE FALSE
```



Examples:

Is 8 less than 6?

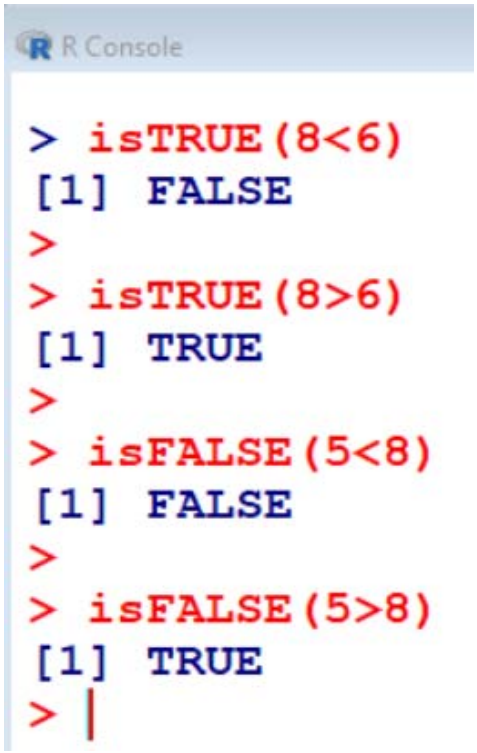
```
> isTRUE(8 < 6)
[1] FALSE
```

Is 8 greater than 6?

```
> isTRUE(8 > 6)
[1] TRUE
```

Is 5 less than 8?

```
> isFALSE(5 < 8)
[1] FALSE
```

Is 5 greater than 8?

```
> isFALSE(5 > 8)
[1] TRUE
```