

Introduction to R Software

Basics of Calculations

:::

Truth Table and Conditional Executions

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

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

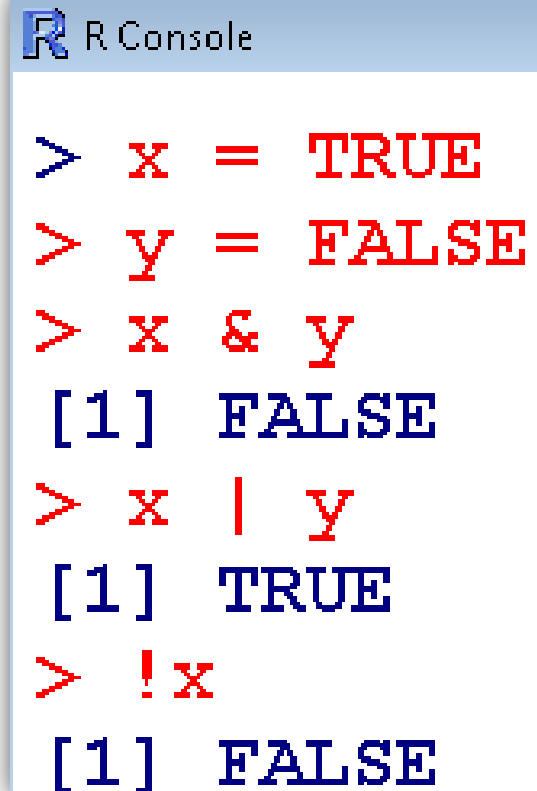
```
> x = TRUE
```

```
> y = FALSE
```

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

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

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



```
R Console  
  
> x = TRUE  
> y = FALSE  
> x & y  
[1] FALSE  
> x | y  
[1] TRUE  
> !x  
[1] FALSE
```

Example

```
> x <- 5
```

```
> Logical1 <- (x > 2)
```

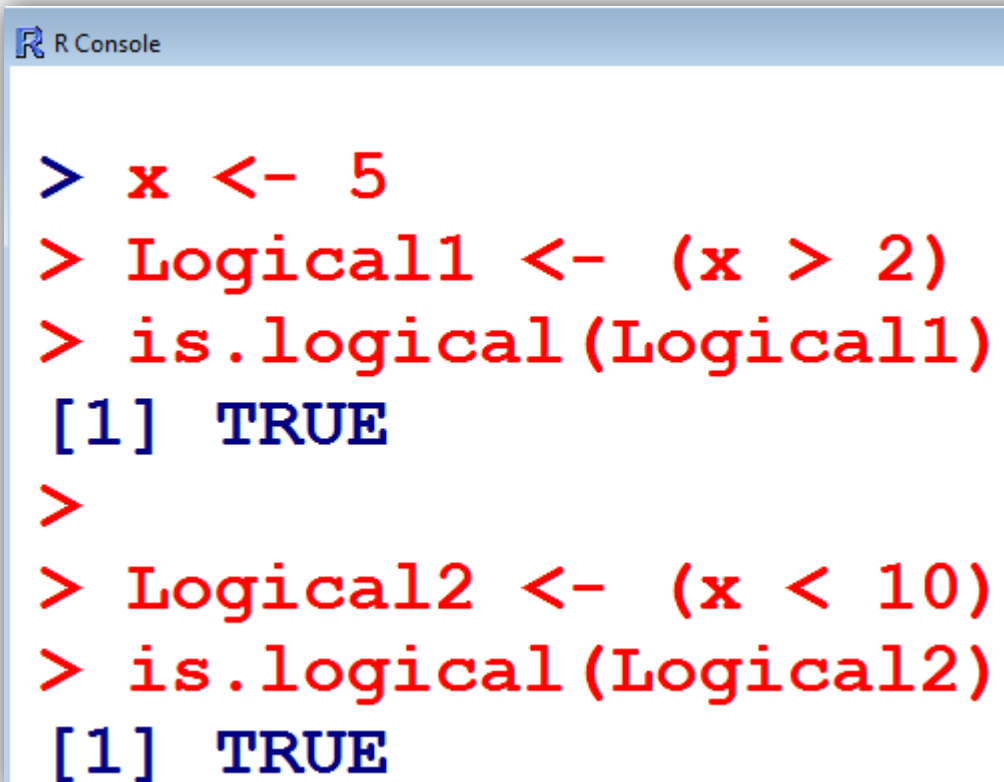
```
> is.logical(Logical1)
```

```
[1] TRUE
```

```
> Logical2 <- (x < 10)
```

```
> is.logical(Logical2)
```

```
[1] TRUE
```



R Console

```
> x <- 5  
> Logical1 <- (x > 2)  
> is.logical(Logical1)  
[1] TRUE  
>  
> Logical2 <- (x < 10)  
> is.logical(Logical2)  
[1] TRUE
```

Example

```
> x <- 5
```

```
> Logical3 <- (2*x > 11)
```

```
> is.logical(Logical3)
```

```
[1] TRUE
```

```
> Logical4 <- (3*x < 20)
```

```
> is.logical(Logical4)
```

```
[1] TRUE
```

R Console

```
> x <- 5
```

```
> Logical3 <- (2*x > 11)
```

```
> is.logical(Logical3)
```

```
[1] TRUE
```

```
>
```

```
> Logical4 <- (3*x < 20)
```

```
> is.logical(Logical4)
```

```
[1] TRUE
```

Control structures in R :

Control statements,

loops,

functions

Conditional execution

1. Conditional execution

Syntax

```
if (condition) {executes commands if condition is TRUE}  
if (condition) {executes commands if condition is TRUE}  
else { executes commands if condition is FALSE }
```

Please note:

- The condition in this control statement may not be vector valued and if so, only the first element of the vector is used.
- The condition may be a complex expression where the logical operators "and" (&&) and "or" (||) can be used.

1. Conditional execution

Example

```
> x <- 5
```

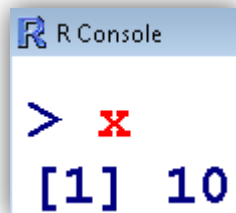
```
> if ( x==3 ) { x <- x-1 } else { x <- 2*x }
```

Interpretation:

- If $x = 3$, then execute $x = x - 1$.
- If $x \neq 3$, then execute $x = 2 * x$.

In this case, $x = 5$, so $x \neq 3$. Thus $x = 2 * 5$

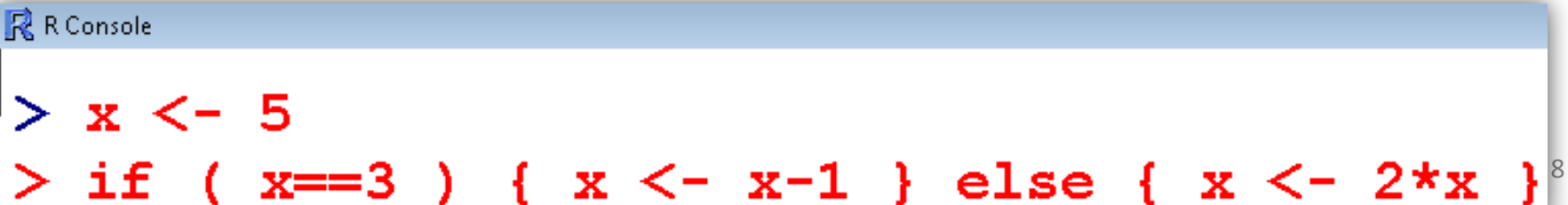
```
> x  
[1] 10
```



R Console

```
> x  
[1] 10
```

Now choose $x = 3$ and repeat this example



R Console

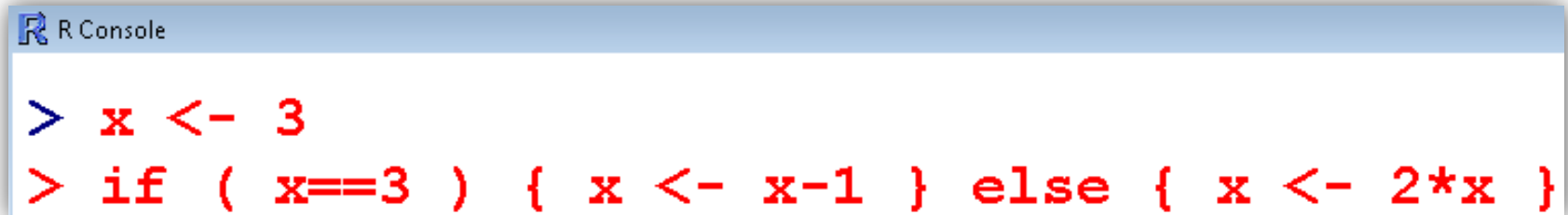
```
> x <- 3  
> if ( x==3 ) { x <- x-1 } else { x <- 2*x }
```


1. Conditional execution

Example

```
> x <- 3
```

```
> if ( x==3 ) { x <- x-1 } else { x <- 2*x }
```



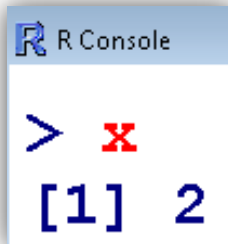
A screenshot of an R Console window. The title bar says 'R Console'. The console shows two lines of code: the first line is `> x <- 3` and the second line is `> if (x==3) { x <- x-1 } else { x <- 2*x }`. Both lines are highlighted in red.

Interpretation:

- If $x = 3$, then execute $x = x - 1$.
- If $x \neq 3$, then execute $x = 2*x$.

In this case, $x = 3$, so $x = 3 - 1$

```
> x  
[1] 2
```



A small screenshot of an R Console window. The title bar says 'R Console'. The console shows two lines of code: the first line is `> x` and the second line is `[1] 2`. Both lines are highlighted in red.