

Intro to Functions

R Programming Structures

Gaston Sanchez

CC BY-NC-SA 4.0

R Coding Compendium

[Donate](#)

About

There are many times where you don't just want to execute one statement after another: you need to control the flow of execution.

In this slides we talk about conditionals, commonly referred to as **if-else** statements

Main Idea

Execute some code when
a condition is fulfilled

Conditionals

If-else or **if-then-else**

This class of statements make it possible to choose between two (possibly compound) expressions depending on the value of a **logical condition**.

Motivation Example

Generate a random Normal number

```
x <- rnorm(1)
```

?

Is it positive or negative?

If **x** > 0



positive

If **x** < 0



negative

Example

```
x <- rnorm(1)

if (x > 0) {
  print("positive")
} else {
  print("negative")
}
```

Example

Equivalently

```
x <- rnorm(1)

if (x < 0) {
  print("negative")
} else {
  print("positive")
}
```


Anatomy of if-else

Anatomy of if-else

```
x <- rnorm(1)
```

```
if (x > 0) {  
  print("positive")  
} else {  
  print("negative")  
}
```


Anatomy of if-else

```
x <- rnorm(1)
```

if-else statement

```
if (x > 0) {  
  print("positive")  
} else {  
  print("negative")  
}
```

Anatomy of if-else

```
x <- rnorm(1)
  Logical condition  single TRUE
if (x > 0) { single FALSE
  print("positive")
} else {
  print("negative")
}
```

Anatomy of if-else

```
x <- rnorm(1)
```

```
if (x > 0) {  
  print("positive")  
} else {  
  print("negative")  
}
```

*What to do if
condition is TRUE*

Anatomy of if-else

```
x <- rnorm(1)
```

```
if (x > 0) {  
  print("positive")  
} else {  
  print("negative")  
}
```

*What to do if
condition is FALSE*

If-then-else

- ▶ `if()` takes a **logical** condition
- ▶ the condition must be a logical value **of length one**
- ▶ it executes the next statement if the condition is true
- ▶ if the condition is false, then it executes the expressions in the `else` clause

The logical condition must be of **length one!**

```
y <- rnorm(2)
```

```
if (y > 0) {  
  print("positive")  
} else {  
  print("negative")  
}
```

```
## Warning in if (y > 0) {: the condition has length > 1 and  
## will be used
```

```
## [1] "negative"
```


What if you don't care about
the condition being FALSE?

Example

When you don't care about the else clause:

```
x <- rnorm(1)

if (x < 0) {
  print("negative")
} else {
  print("positive") # don't care
}                  # don't care
```

Example

If you don't care about the else clause, then don't use it:

```
x <- rnorm(1)

if (x < 0) {
  print("negative")
}
```

Example

R is actually *nullifying* the else clause:

```
x <- rnorm(1)

if (x < 0) {
  print("negative")
} else NULL
```

More Examples

```
# ok  
if (x > 0) print("positive")  
  
# valid but not recommended  
if (x > 0)  
    print("positive")  
  
# ok  
if (x > 0) {print("positive")}  
  
# recommended  
if (x > 0) {  
    print("positive")  
}
```

Reminder of Comparison Operators

Operator	Description
<code>x == y</code>	equal
<code>x != y</code>	not equal
<code>x < y</code>	less than
<code>x > y</code>	greater than
<code>x <= y</code>	less than or equal
<code>x >= y</code>	greater than or equal

- ▶ recall that comparison operators produce logical values
- ▶ they are typically used in `if-else` statements

Reminder of Logical Operators

Operator	Description
<code>!x</code>	NOT
<code>x & y</code>	AND (elementwise)
<code>x && y</code>	AND (1st element)
<code>x y</code>	OR (elementwise)
<code>x y</code>	OR (1st element)
<code>xor(x, y)</code>	exclusive OR

- ▶ logical operators are also typically used in `if-else` statements

Multiple Nested If's

Multiple Nested If's

Generate a random Normal number. Is it positive? Is it negative?
Or is it zero?

```
x <- rnorm(1)

if (x < 0) {
  print("negative")
} else if (x > 0) {
  print("positive")
} else if (x == 0) {
  print("zero")
}
```

Multiple Nested If's

We can simplify the third condition as:

```
x <- rnorm(1)

if (x < 0) {
  print("negative")
} else if (x > 0) {
  print("positive")
} else {
  print("zero")
}
```

Donation

If you find any value and usefulness in this set of slides, please consider making a one-time donation in any amount (via paypal). Your support really matters.

Donate

https://www.paypal.com/donate?business=ZF6U7K5MW25W2¤cy_code=USD