BST02: Using R for Statistics in Medical Research

Part C: Functions and Loops

Nicole Erler

Department of Biostatistics, Erasmus Medical Center

n.erler@erasmusmc.nl

24 - 28 February 2020

Functions

Sometimes we want to perform the same action / manipulation on several objects.

- Option 1: copy & paste
  - a lot of work
  - susceptible to mistakes
- Option 2: functions

What are functions?

- a group of (organized) R commands
- a (small) programm with flexible (= not pre-specified) input

Almost all commands in R are functions!

Functions

Some examples:

- `mean()`
- `sum()`
- `plot()`
- ...

```
class(mean)
## [1] "function"
class(sum)
## [1] "function"
class(plot)
## [1] "function"
```

Even `class()` is a function:

```
class(class)
```

```
## [1] "function"
```

Functions

To write your own function

```
myfun <- function(arguments) {
  syntax
}
```

For example

```
square <- function(x) {
  x^2
}
```

```
square(3)
```

```
## [1] 9
```

Functions

Functions do not always need an argument

```
random <- function() {
  rnorm(1)
}
```

```
random()
## [1] 0.04227781
random()
## [1] -0.4901089
random()
## [1] -0.3752744
```

Functions

Functions can use multiple arguments

```
substract <- function(x, y) {
  x - y
}
```

```
substract(x = 5.2, y = 3.3)
```

```
## [1] 1.9
```

Multiple arguments are interpretet in the pre-defined order, unless they are named:

```
substract(5.2, 1.2)
```

```
## [1] 4
```

```
substract(y = 5.2, x = 1.2)
```

```
## [1] -4
```

Functions

We can also define default values for arguments

```
multiply <- function(x, y = 2) {
  x * y
}
```

```
multiply(x = 3, y = 3)
```

```
## [1] 9
```

```
multiply(x = 3)
```

```
## [1] 6
```

Functions

Go to Demo_Functions01.R

New topic

slide header