

15.2.2

Functions in R

What functions will Jeremy need to master in R? His lead analyst, Colleen, suggested he start with the structure of functions themselves. Functions in R have a similar structure as Python functions, so Colleen thinks Jeremy will get the hang of it pretty quickly.

Functions behave similarly to methods (or even functions) in Python. In R, a function is used to perform a specific task and is denoted by parentheses. Functions can either be built-in, come from imported libraries, or be defined by the user.

In this module, we have already used two built-in functions—the `install.packages()` function and the `c()` function. One required the name of the package to install and returned no value. The other could contain an unlimited number of arguments and would return a vector containing a list of equal size and order.

Regardless of where a function comes from, all R functions use the same basic syntax:

```
function_name <- function(arg1, arg2=T, ...){  
  
  <BODY OF FUNCTION>  
  
  return <RETURN VALUE>  
}
```

There are four components of an R function:

- The **function name** is the name of the function, which can be used in the R console to call the function itself.
- Just like Python methods, R functions can have any number of **required** or **optional arguments**, depending on the design of the function.
- The **function body** includes data structures, if-statements, and other logical statements that define what the function does.
- The **return statement** is the last evaluated statement before returning the resulting value out of the function.



As we continue learning how to analyze and program using R, we'll encounter a wide variety of functions with different inputs, arguments, and outputs. Many of these functions, arguments, and outputs will be very similar to their Python counterparts.

If at any point you are unsure what an R function does or what it needs to execute, you can always type `?<name of function>` in the R console and it'll open the documentation in the Help pane. As we progress, we'll cover some RStudio shortcuts that help us easily navigate and implement obscure functions that we might need in specific situations.

Now that we have learned about assigning data structures and using functions, it's time to bring these concepts together and write our very first RScript.

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.