R for life sciences. Chapter 2: Operations in R

2020-10-13

# Operations

## Sintax and operators

We have already seen in chapter 1 several operators to address or assign data into and object.

### Adressing operators

$ , @ : Address to a component into an objects, by names.

[ , [[ : Indexing components into an object.

? : Help.

<- : Assignment, right to left. The use of = for assignment is not advisable. Remember that **To write " <- " easily, use ALT + " - "** .

~ : As in formulae (write it with Alt-4, or Alt-Ñ in Spanish keyboard).

### Arithmetic operators

by order of precedence. If the objects are not numeric, they will be coerced into numeric, if possible.

^ : Exponential.

* , / : Multiply and divide.
* , - : Addition and subtraction.

%/% , %% : Divisor and remainder for a division.

### Comparison operators

Output will be a Logical or a list of logicals (True - False).

< , > , <= , >= : Leaser, greater, leaser or equal, greater or equal.

== , != : Equal and different.

%in% : Indicates matches.

### Logical operators

! : Logical NOT

& , && : Logical AND

, || : Logical OR

Let’s see some examples with operators:

We can see a table of operators, and more examples, using **?Syntax** (also precedence order).

### Some Arithmetical commands

There are too many commands in R to list them all, but some of them are frequently used for calculations.

This commands return a number:

An these operations can modify either a number or all the numbers in a vector or a matrix:

And of course, it is possible to make combinations of different commands. For example to calculate the **standard error (SE)** of x, which is:

being the standard deviation of x and *n* the sample size (number of items in the sample).

## Condicionals and recursive commands

The most used ones are **if()** and **for()**. Other control flow commands are while() and repeat(). if() can be used either with of without else. They function in much the same way as control statements in any Algol-like language. Also important the expressions ***break*** and ***next*** to control the flow.

Braces are not necessary in the same line, but is advisable to use them always because is a frequent source of errors.

Examples:

# Exercises

1. Open the data frame in **iris {datasets}**. Use the help to know about this data. In which units are measured the length and width of sepals and petals? How many variables and observations are there in **iris**?
2. Create a vector with the species names. Remember that genus should be with capital letters and species in small letters (e.g. “Iris setosa”).
3. Create a vector with the name of all quantitative variables
4. Make a data frame with the combination of the two previous vectors like this:

## Species Variable  
## 1 Iris setosa Sepal.Length  
## 2 Iris setosa Sepal.Width  
## 3 Iris setosa Petal.Length  
## 4 Iris setosa Petal.Width  
## 5 Iris versicolor Sepal.Length  
## 6 Iris versicolor Sepal.Width  
## 7 Iris versicolor Petal.Length  
## 8 Iris versicolor Petal.Width  
## 9 Iris virginica Sepal.Length  
## 10 Iris virginica Sepal.Width  
## 11 Iris virginica Petal.Length  
## 12 Iris virginica Petal.Width

1. Using dataframe from exercise 4, make a data frame with the following variables:

* Species.
* Variable.
* Mean, the mean for each variable and species.
* Standard\_error, the standard error for each variable and species.
* Median, the median for each variable and species.
* Minimum, the minimum for each variable and species.
* Maximum, the maximum for each variable and species.

1. Install the package “writexl” and use the command write\_xlsx to create a “yourname.xlsx” file with your data frame.

If you want, you can also use the command WriteXLS from the WriteXLS package, but you will need Perl installed in your computer.

Use the commands seen in this and previous chapters to do the code the neatest possible. Remember to comment each step to know what are you doing. When sourcing your script, the xlsx file should appear without errors nor warnings.

Always comment the “install.packages() line”.

# About this tutorial

Cite as: Alfonso Garmendia (2020) R for life sciences. Chapter 2: Operations in R. <http://personales.upv.es/algarsal/R-tutorials/02_Tutorial-2_R-operations.html>.

Available also in other formats (pdf, docx, …): <https://drive.google.com/drive/folders/19w914WCg8BVTVBE_zpgShmg2vpjguV1e?usp=sharing>.

Other simmilar tutorials: <https://garmendia.blogs.upv.es/r-lecture-notes/>

Originals are in bitbucket repository: <https://bitbucket.org/alfonsogar/tea_daa_tutorials>.

Document written in [Rmarkdown](http://rmarkdown.rstudio.com/), using [Rstudio](https://www.rstudio.com/).

* System: R version 3.6.3 (2020-02-29) x86\_64-pc-linux-gnu (64-bit) Ubuntu 18.04.5 LTS.
* Base packages: stats 3.6.3, graphics 3.6.3, grDevices 3.6.3, utils 3.6.3, datasets 3.6.3, methods 3.6.3, base 3.6.3.
* Other loaded packages: writexl 1.2, rmarkdown 1.17, googledrive 0.1.3.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-nc-sa/4.0/).