

R basics, help, mistakes

Andrew Ba Tran

Contents

Some R code basics	1
R Workspace	1
Packages	2
Getting Help	2
Working with Packages	2
Common Mistakes	2

This is from the first chapter of learn.r-journalism.com.

Some R code basics

- `<-` is known as an “assignment operator” – it means “Make the object named to the left equal to the output of the code to the right”
- `&` means AND, in Boolean logic
- `|` means OR, in Boolean logic
- `!` means NOT, in Boolean logic
- When referring to values entered as text, or to dates, put them in quote marks like this: `"United States"`, or `"2016-07-26"`. Numbers are not quoted
- When entering two or more values as a list, combine them using the function `c`, for combine, with the values separated by commas, for example: `c("2017-07-26", "2017-08-04")`
- As in a spreadsheet, you can specify a range of values with a colon, for example: `c(1:10)` creates a list of integers (whole numbers) from one to ten.
- Some common operators:
 - `+` `-` add, subtract
 - `*` `/` multiply, divide
 - `>` `<` greater than less than
 - `>=` `<=` greater than or equal to, less than or equal to
 - `!=` not equal to
- **Equal signs can be confusing**
 - `==` tests whether the objects on either end are equal. This is often used in filtering data
 - `=` makes an object equal to a value, which is similar to `<-` but used within a function.
- Handling null values:
 - Nulls are designated as `NA`
 - `is.na(x)` looks for nulls within variable `x`.
 - `!is.na(x)` looks for non-null values within variable `x`

Here, `is.na()` is a **function**. Functions are followed by parentheses, and act on code/data in the parentheses.

{{% notice disclaimer %}} Object and variable names in R should not contain spaces {{% /notice %}}

R Workspace

- Your current R working environment
- Includes any user-defined objects (e.g. vectors, data frames, functions)

Function	Action
<code>getwd()</code>	List current working directory
<code>setwd("mydirectory")</code>	Change the current working directory to my directory
<code>ls()</code>	List the objects in the current work space
<code>rm(object)</code>	Delete object
<code>save(object, file="myfile")</code>	Save specific objects to a file
<code>load("myfile.RData")</code>	Load a work space into the current session (default = .RData)

Packages

- Collections of R functions, data, and compiled code in well-defined format
- Massively extend the functionality of R
- Thousands of user-written packages on CRAN
 - <https://cran.r-project.org/web/packages>

{{% notice tip %}} Mac users may need to alter their security preferences to allow apps authored by non-Apple developers to install. If you notice an error, try to change your system preferences. {{% /notice %}}

Getting Help

Function	Action
<code>help.start()</code>	General help
<code>help("foo")</code> or <code>?foo</code>	Help on function foo (the quotation marks are optional)
<code>help.search("foo")</code> or <code>??foo</code>	Search the help system for instances of the string foo
<code>example("foo")</code>	Examples of function foo (the quotation marks are optional)

Working with Packages

- `install.packages("packagename")`
- `update.packages()`
- `library(packagename)`
- `help(package="packagename")`
- `library()` #what packages are in the library
- `search()` #what packages are loaded

Common Mistakes

- **Using the wrong case**
 - `help()`, `Help()`, and `HELP()` are three different functions (and only the first one will work)
- **Forgetting to use quotation marks when they are needed**
 - `install.packages("gclus")` will work, while `install.packages(gclus)` will generate an error.
- **Forgetting to include the parentheses in a function call**
 - `help()` rather than `help`. Even if there are no options, you still need the `()`.

- **Using the \ in a path name on Windows**
 - R sees the backslash character as an escape character.
 - `setwd("c:\\mydata")` will generate an error. Use `setwd("c:/mydata")` or `setwd("c:\\mydata")` instead
- **Using a function from a package that is not loaded**
 - For example: The function `str_trim()` is contained in the **stringr** package.
 - If you try to use it before loading the package, you will get an error