Intro R miscellanea

Roberto Molowny-Horas

April 24-25, 2023

To tell R we want to go/work on one particular directory we use **setwd()**. To know where we are (i.e. the absolute filpath of the current directory) we can use:

```
getwd()
```

To know which objects we have stored in our environment we can use ls() or objects().

```
ls()
objects()
```

To look up the help file for a function, try **?** or **help()**.

```
?log
help(log)
```

If you don't know exactly what you're looking for, but you know that it must contain a given string, you can try ??.

```
??logarithm
```

Try the following to get a big help file.

```
help.start()
```

In R people use <- instead of = to assign values to variables. Either one works fine, but tradition has it that a good R programmer should use <-. ;)

```
a <- 1
b = 1
```

A good introduction to R can be found in the "An introduction to R" manual (in PDF) that comes enclosed with the R installation. Just write **help.start()** and copy/paste the URL that appears in the console ('http://127.0.0.1:10946/doc/html/index.html (http://127.0.0.1:10946/doc/html/index.html)', or similar) on a browser.

Packages

A package is a very useful feature of R. It packs together a set of functions which perform some (very) useful operations. They are usually “thematic”, meaning that all functions in a given package have been designed to solve one specific problem, or several related ones. This is, however, not forced strictly and we can find all kind of packages doing all kind of things. To install a specific package we do e.g.:

```
install.packages(c("sp","raster"))
```

This fetches the package over the internet from a centralized R server (‘https://cran.r-project.org/ (https://cran.r-project.org/)’) and stores on our local disk. To incorporate those functions of the package into our program we use these two commands, which differ in minor things (see their help documentation):

```
library(sp)       # Recommended!!
require(sp)
```

Then you can start using functions from those packages. An alternative way, which I use very often, is to call those functions without loading the package. However, that package should have been installed previously with **install.packages()**! For instance, I want to use function **npbs** from package **nptest** to check whether the median

```
# install.packages("nptest")
npbs <- nptest::np.boot(iris$Sepal.Width, statistic = mean)    # nonparametric bootst
rap
hist(iris$Sepal.Length, breaks = 50)
print(npbs)
```

If we need a help for one specific package we can do:

```
help(package=sp)
```

Vignettes are useful descriptions of code and related maths that are included in many packages. We can check all vignettes available at our R installation:

```
browseVignettes()                # It opens in external browser.
vignette(package="sp")           # List of available vignettes for the "sp" package.
vignette("over",package="sp")    # PDF of vignette "over".
```