How to install packages in RStudio

After you have successfully installed both R and RStudio, you are ready to start programming in R.

R has a plenty of built-in functions, but most of the time additional functions are needed. The true power of this programming language lies in the support from approximately 10,000 additional packages that can be installed to widen its functionality.

To install the packages that we are going to use during our 3-day workshop, please follow these steps:

1. Open RStudio from Start Menu (Windows users) or Applications folder (OS X users).
2. Assume that we want to install the package called: dplyr (it is a well-known R package, indeed)
3. There are two ways to do it (actually, there are more than two ways, but let's stick with the following two):
    a. In the RStudio R console (the lower left quadrant of your RStudio workspace), type: install.packages('dplyr')
    b. From the Tools menu in RStudio, select Instal Packages… then start typing the package name and select it from the list
4. Most of the CRAN packages - those found in the standard, official R repository, called CRAN (Comprehensive R Archive Network) can be installed by following the standard procedure (described in Step 3). However, besides this basic package installation procedure, R packages from various sources can be installed in other ways as well. During the workshop we will demonstrate how to install R packages from the GitHub repository (which is not R specific) using the devtools package from R, and how to install packages from the Bioconductor repository.
5. Note: sometimes, most often in the R code comments, we use enclose the names of R packages in braces, like: {dplyr}, in order to visually mark the mention of an R package clearly.
6. Note: during the installation of some packages, you may notice how R starts installing additional packages - some packages that you have not selected for installation at all! The reason for this is the following: some R packages rely on R functions developed in some other packages. When an R package uses functions from another package, the later is called its dependency. R will automatically try to install all the dependencies of the package that you have selected for installation, so don't be surprised if installing one package actually triggers and installation of a dozen of other packages at your system. It's perfectly normal.

We will use many different R packages during the workshop. Here is a list of the packages that we will be needing; please take some time to practice package installation. To have the following packages pre-installed on the onset of our Workshop would really save us some time:

stringr, ***stringi***, ***dplyr***, ***tidyr***, ***hflights***, ***ggplot2***, ***corrplot***, ***ppcor***, ***Hmisc***, ***car***, ***QuantPsyc***, ***broom***, ***lattice.***

REMINDER: to install a package from the RStudio IDE, type in the console e.g.:

```
install.packages(“dplyr”)
```