Notes: Create an R package

2016/09/08

This notes are based on a subset of the book "R packages" by Hadley Wickham. It is assumed that the reader is familiar with coding in R (e.g. creating functions from scratch, writing scripts, plots, etc), RStudio and command line instructions. I will not cover coding best practices or how to name objects which are very important things to know. Many thanks to Ernesto Barrios who encouraged me and shared his book with me and to Pierre Jacob for his useful tips.

1

The first thing I realised was that I had a super old version of R. I downloaded the current version (3.3.1) for Mac OS from: `https://cran.r-project.org/`. Then, RStudio was still using the old version after reloading it so I had to remove the old version (2.13) from my computer with the following command line :

```
cd /Library/Frameworks/R.framework/Versions/
rm -r 2.13
```

After this, RStudio automatically detected the new version. If you have lots of packages installed in your old version you might want to save the folder with your downloaded packages and include it in the new version! But do update your old packages with:
`update.packages(checkBuilt=TRUE)`.
Succesively, I installed the following build-your-own-package packages and checked the RStudio version was the right one:

```
# Install packages that will help you create your own package
install.packages(c("devtools", "roxygen2", "testthat", "knitr"))
# Package that checks the version of R Studio you have installed
install.package("rstudioapi")
rstudioapi::isAvailable("0.99.149")
```

2

Choose a not-too-cryptic but not-too-long name for your package. Upper and lower case letters and dot (·) are allowed for the name but the latter should be avoided due to S3 extensions.

3

RStudio menu: File → New Project → New Directory → R package or

```
devtools::create("path/to/package/pkgname")
```

The following three components are created: a directory called *R*, DESCRIPTION and NAMESPACE files and a RStudio file called *pkgname.Rproj*.

Add or suggest package dependencies with:

```
#This command directly imports the package between brackets
devtools::use_package("coda")
#This command suggests the user to import the package between brackets
devtools::use_package("melt","Suggests")
```

⋮

To be completed

⋮

5 Don'ts

- Never use `library()`, `require()` or `source()`.