Install tutorial

Aurélien Ginolhac

14 April 2016

Contents

Install R

R is available for free for Windows, GNU/Linux and MacOS.
As for now, the latest version is `3.3.0`.

One common complain about `R` is the frequency of updates. This is true, but can be sorted out with a good **package manager**. Of note, major updates (`3.3`, early May 2016) are the only one forcing you to reinstall extra-packages and are released on a yearly basis.

Windows

Visit the download page and choose the corresponding installer for your platform.

To circumvent the lack of a package manager in Windows, it is possible to install the R library `installr`. This package provides the `updateR()` function which takes care of updating your R installation and migrates your installed libraries whenever a major `R` update is released. Recently, a package has been released which can also help you in this task .

MacOS

You can either browse to the download page or better use the Homebrew package manager.

Once `homebrew` is installed, you can easily install/uninstall many programs with their dependencies and utilities. This implies to use the terminal. If you're not comfortable with the use of the terminal, ask me! It is worth it. To install `R` using brew write in the following:

```
brew install r
```

Updates for all your software can be easily performed and will be automatically handled by the following line:

```
brew update; brew upgrade
```

GNU/Linux

Any Linux distributions is bundled with a great package manager such as `dpkg` or `rpm`. Visit the appropriate webpage for your distribution here.

After you get the correct entry in your package manager, it should quite easy

```
sudo apt-get install r-base r-base-dev
```

Install Rstudio

Rstudio is an Integrated Development Editor, it wraps and interfaces `R` but, `R` needs to be installed first. The free-version contains everything you need.

Visit the download page and choose the corresponding installer for your platform. As for now, the latest version is `0.99.1172`.

Open rstudio

They are 4 main panels in rstudio, but as the top-left is for scripting and by default missing, the layout should looks like
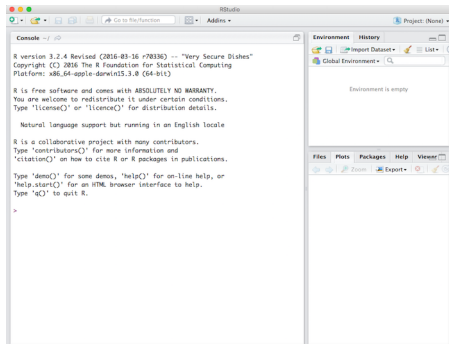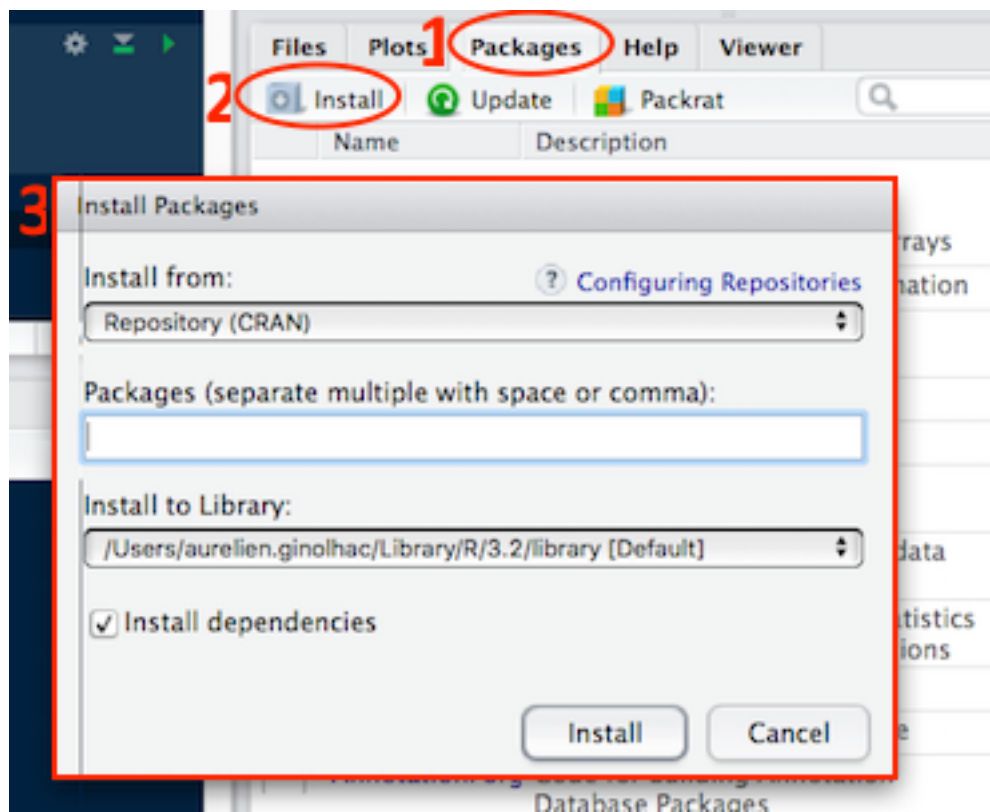

install R packages from the Comprensive R Archive Network (CRAN)

On the bottom-right panel, 5 tabs are present:

- Files
- Plots
- Packages
- Help
- Viewer

Click on the `Packages` tab (1.) and select the select the `Install` button (2.). Type `dplyr, tidyr, ggplot2, ggrepel, devtools, broom, purrr, shiny, readr, readxl, d3heatmap, stringr, rmarkdown, knitr, tibble` in (3.)

It takes some time, usually 10 seconds up to 5 minutes for a single package depending on its size and compilation stage.

The packages can also be installed in the console (*i.e.* the bottom-left panel). The snapshot below shows you how to install `devtools` using the console:

Of note, `devtools` is handy when you want to install a development version, for example if a bug was fixed but the package not yet submitted to CRAN.

```r
devtools::install_github("hadley/dplyr")
```

A note about package updates

The complaint of very frequent updates might be legitimate as it could be cumbersome to maintain `R` up-to-date when you have many packages. Actually, it is as easy as:

```r
update.packages()
```

But, you should rather use the green `Update` button next to the `Install` (located in the `Packages` tab). The advantage is that you can check which package you would like to update and look at the actual changes by clicking on the `NEWS` link.

install bioconductor packages

The bioconductor resource will be detailed later, but to save time, copy / paste the following code into the consoleto install the necessary packages before the workshop:

```r
source("https://bioconductor.org/biocLite.R")
# to install the version 3.3 of the bioconductor installer.
biocLite(c("limma", "Biobase", "GEOquery"))
```

Package updates

There is no interface for bioconductor packages thus you need to run the following lines to update the packages (you will be asked if you want to update some / all):

```r
source("https://bioconductor.org/biocLite.R")
biocLite()
```

About packages and major version

A very recent package `reinstall` is doing a neat job looking at the installed packages and those you already used by parsing your your R files (`.R`, `.Rmd` etc.).

For example, after I upgraded to `3.3` I forgot some packages and can decide if I want to install them or not.

```r
#devtools::install_github("calligross/reinstallr")
library("reinstallr")
reinstallr(path = "~/Work/")
The following packages were found in your source files and can be installed from CRAN:
ggdendro openxlsx UpSetR pastecs wesanderson DNAcopy impute gplots directlabels DT GGally purrr ggpmisc
Do you want to install them now?
y: Yes! Go ahed!
n: No, fortget it!
```
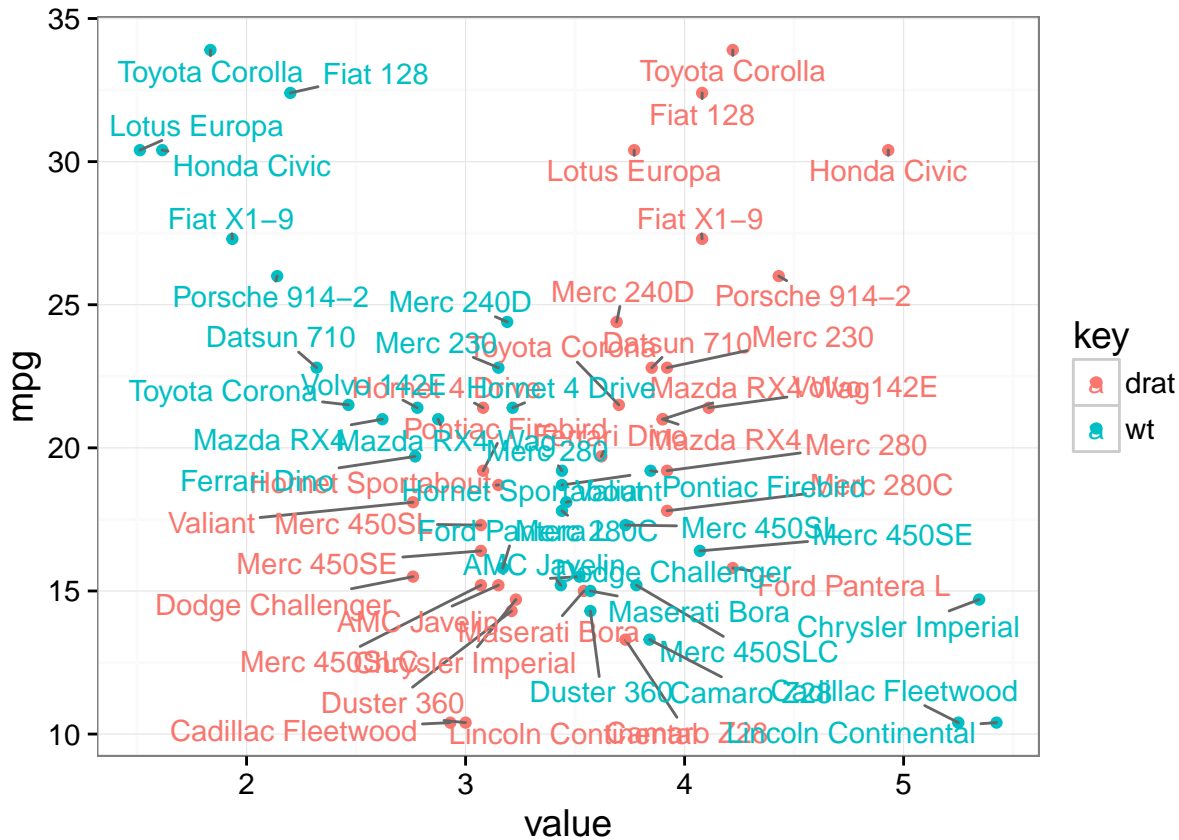
Testing your installation

Copy / paste the code below and you should obtain the following plot.

```r
library("ggplot2")
library("tidyr")
library("ggrepel")
library("dplyr", warn.conflicts = FALSE)
theme_set(theme_bw(14))
```

```
mtcars %>%
  tibble::rownames_to_column(var = "car_name") %>%
  gather(key, value, c(drat, wt)) %>%
  ggplot(aes(x = value, y = mpg, colour = key))+
  geom_point()+
  geom_text_repel(aes(label = car_name))
```


If not, please contact me with the error produced and the output of

```
sessionInfo()
```

```
## R version 3.3.0 (2016-05-03)
## Platform: x86_64-apple-darwin15.4.0 (64-bit)
## Running under: OS X 10.11.4 (El Capitan)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_0.4.3    ggrepel_0.5    tidyr_0.4.1    ggplot2_2.1.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.5       knitr_1.13        magrittr_1.5      munsell_0.4.3
##  [5] colorspace_1.2-6  R6_2.1.2          stringr_1.0.0     plyr_1.8.3
##  [9] tools_3.3.0       parallel_3.3.0    grid_3.3.0        gtable_0.2.0
```

```
## [13] DBI_0.4-1        htmltools_0.3.5  lazyeval_0.1.10  yaml_2.1.13
## [17] digest_0.6.9     assertthat_0.1   tibble_1.0-5     formatR_1.4
## [21] evaluate_0.9     rmarkdown_0.9.6  labeling_0.3     stringi_1.0-1
## [25] scales_0.4.0
```