Overview

October 30, 2017

Why R?

- Open source and extensible. It is also free of charge. More importantly, you (and the community) can inspect the code for any function.
- Very popular *programing language* for statistics. "It promotes experimentation and exploration, which improves data analysis." Lots of libraries for all kinds of specialized tasks.
- Great for visualization. Excellent packages for graphics.
- Excellent documentation and online help.
- A very active and helpful community.

What is R anyway?

If you are coming from SAS or Stata, you are better off thinking about it as a programming language and not as a statistical environment:

- Interpreted: allows direct executions of code, which makes it slower than compiled languages.
- Dynamically typed: Easy and less boiler plate, makes metaprogramming a lot easier.
- Multi-paradigm. Although most users exploit object orientation.

The extension of the R files is usually .R

RStudio

R can be downloaded from the Comprehensive R Archive Network, CRAN. We will be using RStudio, a popular IDE. It is important to keep in mind that R (the language) and RStudio (the GUI) are separate things, and it is entirely possible to use different workflows with other tools or text editors:

- emacs through ESS (what I use).
- vim with the Vim-R-Plugin.
- Sublime Text.

A few useful resources

There is a constantly growing collection of materials available offline and online to learn R. The Journal of Statistical Software and the Use R! series from Springer regularly publish applications of R to different domains.

A good overview for beginners is Learning R.

SAS users may find useful R for SAS and SPSS users, although I have never used it myself.

For the analysis of compley survey data, you may want to take a look to "Complex Surveys. A Guide to Analysis Using R".

The official documentation in CRAN (The Comprehensive R Archive Network) is simply excellent but it goes well beyond the scope of this class.

Looking around

RStudio offers four basic windows.

- Console (R interpreter)
- Code, where we will write our code.
- History/Environment
- Plots/Packages/Help

Getting help

The documentation of R has a well-earned reputation of being excellent. It can be accessed through the interpreter. For instance, if we wanted to get information about what lm does, or what paramaters it takes or some examples of usage, we would type:

?lm

The R community is very helpful and active. If you ever get stuck in a problem, the best solution is to ask in StackOverflow, a very large community of programmers using the #r tag.

For obvious reasons, the language is sometimes referred to as "Rstats" (social media and search engines, amirite?).

Within Westat, there is a growing community of users and we have a number of resources for Q&A and sharing information or announcements.

About this document

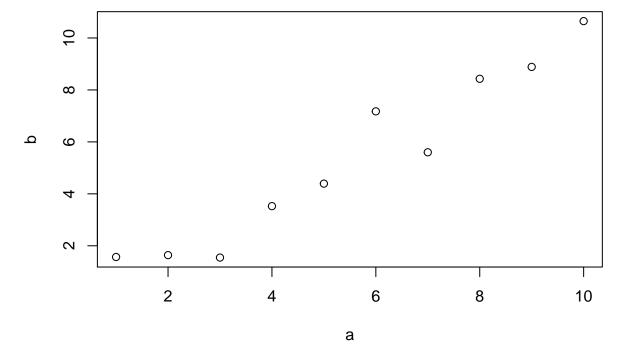
I have prepared these materials using using Rmarkdown, a format that makes it easy to create dynamic documents. The text is written in markdown, the easiest possible markup language ("easy-to-read, easy-to-write"): _italic_, **bold**, ... but it also allows to include chunks of executable R code. It simplifies reproducibility and it is very easy to share.

For instance,

is rendered as:

```
N <- 10
a <- 1:N
b <- a + rnorm(N)
plot(a, b); title("An ugly plot")</pre>
```

An ugly plot



GitHub and git more generally deserve a class on its own (we cover it in the Advanced R class), but here it will only be used to host the notes.