Expressing Analyses Well with Literate Computing

Dan MacLean

2021-04-20

Contents

Chapter 1

Setting up

The primary purpose of this course is to help you to record your analysis and results in a way that can be read and reproduced easily by you and by others. The course will show you two closely related notebooking tools in R and Python that are extremely useful for creating and organising an analysis and for re-doing it when any new data come available or changes are required.

We'll be working in R and Python for this short course.

1.1 Prerequisites

1.1.1 Knowledge prerequisites

The materials in this book assume that you already know something (but not necessarily a great deal) of the languages R and Python, so there won't be any introduction to the languages themselves. The rest of this chapter will help you set up the software you need to practice with those tools.

1.1.2 Software prerequisites

You need to install the following stuff for this book:

1. R
2. RStudio
3. Some R packages: `rmarkdown`, `knitr`
4. Python 3 via Anaconda
5. A reasonably recent web-browser

1.2 Installing R

Follow this link and install the right version for your operating system https://www.stats.bris.ac.uk/R/

1.3 Installing RStudio

Follow this link and install the right version for your operating system https://www.rstudio.com/products/rstudio/download/

1.4 Installing R packages in RStudio

1.4.1 Standard packages

In the RStudio console, type

```
install.packages(c("rmarkdown", "knitr"))
```

1.5 Installing Python 3 with Anaconda

Follow this link and install Python *3.x* for your operating system. https://www.anaconda.com/distribution/

1.5.1 Note for macOS users

Accept all of the defaults during installation

Here is a video tutorial https://www.youtube.com/watch?v=TcSAln46u9U

1.5.2 Note for Windows users

Install Python 3 using all of the defaults for installation except make sure to check **Add Anaconda to my PATH environment variable**.

Here is a video tutorial https://www.youtube.com/watch?v=xxQomzZ8UvA

1.5.3 Note for Linux Users

You'll need to be able to use the command-line to install with Anaconda. If you aren't comfortable with this, ask for assistance from the local support team.

1. Open https://www.anaconda.com/download/#linux[1] with your web browser.
2. Download the Python 3 installer for Linux.

[1]

3. Open a terminal window. 4.Type `bash Anaconda3-` and then press Tab. The name of the file you just downloaded should appear. If it does not, navigate to the folder where you downloaded the file, for example with: `cd Downloads`. Then, try again.
4. Press `enter`. You will follow the text-only prompts. To move through the text, press the `spacebar`.
5. Type `yes` and press `enter` to approve the license.
6. Press `enter` to approve the default location for the files.
7. Type `yes` and press `enter` to prepend Anaconda to your `PATH` (this makes the Anaconda distribution the default Python).
8. Close the terminal window.

1.6 Starting a Jupyter Notebook

1.6.1 macOS

1. Start the `Terminal` application in `Applications -> Utilities`
2. Type `jupyter notebook`, it should start in your web browser

1.6.2 Windows

1. From the Start menu, search for and open `Anaconda 3` or `Jupyter Notebook`. You should be able to start a notebook directly by clicking the `Jupyter Notebook` icon.

1.6.3 Linux

1. Open the terminal application. It is *usually* in the task bar or dock
2. Type `jupyter notebook`, it should start in your web browser

1.7 Installing Python Packages with `conda`

You can use `conda` to install new Python packages using the Terminal by typing `conda install <package_name>`.

You can install the required packages with the following commands:

```
conda install jupyter
```

Accept all defaults when the system asks a question.

Chapter 2

Motivation

2.1 If we didn't have to do it over and over, it wouldn't be called *re*-search.

Developing a data analysis is hard, it can involve many mis-steps and changes of mind from redoing of little bits here and there that weren't quite right the first time, to introducing new ideas or removing whole sections that didn't work out. This iterative process is completely in-line with all other aspects of research and means that we have a personal need to be able to record exactly what we've done with high accuracy, and high reproducibility. It is also our scientific responsibility and an aspect of scientific integrity that we are clear and open about the methods we use as they are key in the interpretations and understanding of the results that we get. In the jargon of the field we think of this as 'keeping a proper lab book', but when it comes to using a computer, what we need to record, when is not often clear nor is it sometimes easy to do so. As a result the methods sections of many reports, theses and papers report scientific computing in a vague and uninterpretable way, making statements like 'tests were done in Excel' or 'GenStat was used to perform *t*-tests', or 'a custom R script was used'. These nebulous reports are useless for anyone trying to understand exactly what was done and reports using them are unreproducible. That they pass reviewers so often is a clear indication of the failure of reviewing of methods. In practice these sorts of write ups are no better and no more informative than announcing that statistical analyses were done with a magic spell.

A major failing of computer graphical interfaces is that they do not make it easy for us to repeat actions, which is ironic as computers are excellent at repeating instructions very quickly. Scripts and programs are required to get the best reproducibility out of our computers, but scripts in R and Python (and any other computer language) are not easily read by people, even those with a great deal of experience in programming. Very quickly reproducible scripts become unusable lumps of code because users can't tell what is in them and what they are supposed to do, a phenomenon that has it's own