

Getting started with StatsLunch

Robert Bagchi

January 28, 2016

Contents

1	Intro	1
2	Getting R	1
3	Rstudio	2
4	latex	2
5	Git	3

1 Intro

I noticed that we have actually covered a bunch of things in the first two classes and partly to keep track of them for myself and partly to make sure everyone has somewhere to go to catch on things they feel they've missed I decided to put together this "getting set up document".

2 Getting R

R is the stats package we'll be using throughout this seminar. Below are instructions on getting set up:

YOU can get hold of it by navigating to <https://www.r-project.org/> and downloading the latest version, which is v3.2.3 at the time of writing. Install however you need to on your system.

If you are using Linux, note that the version of R that is provided on the standard repositories (e.g. that you get by typing apt-get) is not the most up-to-date. You should add the R repository to your trusted sources list (stored in a file something like `/etc/apt/sources.list`) before using apt-get. Instructions for doing this are available at <https://cran.r-project.org/bin/linux/ubuntu/README>

If you are using Windows it helps to have R added to your path. To do this:

- Find the directory in which you've install R (on my computer this is in `C:\ProgramFiles\R\R-3.2.3`), navigate to the *bin* directory. This directory should contain the R executable, *R.exe*. Copy the path to this directory.
- Right click on **Start -> Computer** and scroll down to **Properties**
In the subsequent dialogue navigate to **Advanced system settings -> Environment Variables**.
In the box labelled **System variables** scroll down to **Path** and click **Edit**. Go to the end of the text in that box, type a ";" and then paste the path to your R executable. On my computer, I add the following to my path: `C:\ProgramFiles\R\R-3.2.3\bin\`

3 Rstudio

Rstudio is a great program for managing your R code. Cory has written a document describing some of it's features.

https://github.com/cmerow/Stats_Lunch/blob/master/1_19_16_Rstudio_Intro/Rstudio_Intro.Rmd

One reason we are using it in this seminar is that it simplifies the workflow we are planning to use that integrates R, LaTeX and **Git**. It can be downloaded from <https://www.rstudio.com/products/rstudio/#Desktop>

4 latex

LaTeX is a very powerful language for type setting. In the context of this seminar it's utility comes from allowing one to seamlessly integrate text, code and output in a single document. There is a bit of a learning curve to get good with LaTeX, but you don't have to be *good* to get up and running and find LaTeX *useful*

Download latex well before coming to class. It's a very large download. Download options are here:

<https://latex-project.org/ftp.html>

On windows - MikTeX works pretty much out of the box - <http://miktex.org/download>

Restart Rstudio once you've installed latex, and it should recognize that LaTeX is installed. Try building a basic sweave document (5 min) following the instructions here:

http://rstudio-pubs-static.s3.amazonaws.com/639_b3a59601ba94400aabb29025de83c10.html

You don't need to create the whole document according to these instructions; you've succeeded once you've entered a little text and compiled the pdf. Note that the file needs to be saved to a path on your computer that has **no spaces**. LaTeX doesn't like spaces unfortunately. If you have issues compiling this basic document, please troubleshoot for your system before class.

With windows, as with R, it helps to have the latex executable you're using in your path. My installation did this automatically, but if you want to check, follow the steps listed for R to examine your path. If it's not there, find out where latex was installed and add that location to your path. For example, this might be `C:\ProgramFiles(x86)\MiKTeX2.9\miktex\bin\`

5 Git

Git is a great way to collaborate when writing code, which is why we are using it in this seminar. There are many advantages to using Git including allowing lots of people to access your code, tools to compare different versions, avoid conflicts and many other things. Pariksheet will cover the details in week 3. For now, here's how to get started

- **Step 1: Sign up to GitHub.**

Go to in a web browser <https://github.com/join>

Fill out the forms.

- **Step 2: Get git on your computer.** You can get instructions on setting up git from Cory's presentation from week 1 https://github.com/cmerow/Stats_Lunch/blob/master/1_19_16_Rstudio_Intro/Rstudio_Intro.Rmd (under Git).

To reiterate what he says there,

download git from <https://git-scm.com/downloads>

Click on the latest source release (2.7.0) for your system and installing it. Make a note of which directory you used for the installation - you'll need it in the next step. I think it asks you during the installation if you want to add git to your path, do so.

- **Step 3: Linking to R Studio.** Once you have Git installed, you need to link it to RStudio. Click Tools -> Global Options, then click the Git/SVN tab.

In the box labelled Git executable add the path to the directory where you installed git.

On my computer (Windows) it's `C:/ProgramFiles/Git/cmd/git.exe`

I imagine it'll be something similar on everyone else's. You might not need this step if your path is set right during the Git installation, but that didn't seem to quite work for me.

You then want to create an RSA key - click on the button and follow the instructions (*disclosure* - I didn't do this through RStudio, so I don't know how hard it'll be - my guess is, not very.)

Hit OK.

- **Step 4: Check it's working** In RStudio, click **File -> New Project -> Version Control** then click **Git**

If it doesn't give you errors at that stage, you're probably all set up.

We can get to cloning repositories and so on in week 3. Pariksheet suggests that everyone gets familiar with some of the tools available by working through this tutorial <https://try.github.io/levels/1/challenges/1>