The objective of this section is to briefly introduce you to the version control system Git and the repository website Github. The exposition is very need-to-know - for a great tutorial on the many ins and outs of Git, check out `https://www.atlassian.com/git/tutorials/`. There is also an excellent book that's free online at `http://git-scm.com/book/en/v2`. Chapters 1-3, and 6 cover the essentials well.

What is version control?

Version control refers to the management of changes to documents or programs. The main value of version control for us is the deliberate staging of our code so that we can look back at previous versions that are saved when we know they work. We can easily use a service like Dropbox to back things up, and save different versions, but we end up with far too many files going back for it to be very helpful if we don't remember the exact date we want to go back to. Crucially, one needs a version control system to look back at previous states of the *entire project*. Dropbox, only allows the user to look back at old versions of individual files.

You may end up opting not to use an explicit versioning system if you deem the setup costs not worth the future benefits. If so, I urge you to, at the very least, use an automatic online backup.

You will start by downloading and installing Git for your operating system from the following link:

`http://git-scm.com/downloads`

There are several GUI-based clients that you may wish to use:

`http://git-scm.com/downloads/guis`

I won't cover any of these, as I know none of them well.[1]. All Git commands can be accessed using the command line tool, so this is all you need to know. If you learn the command line tool first, it's highly likely that you'll easily understand a GUI based tool.

Starting Git

Once Git is installed, it should run directly from the terminal in Mac or Linux, or using the Bash shell emulator that gets installed with Git on Windows (named "Git Bash" on my installation – this is just an emulator of a Linux style terminal). Git, as we're using it, is a command line utility, so doesn't require its own window as such.

Starting a repository locally

There are two ways to get a repository started locally on your computer. The first is to use the command:

```
$ git init
```

[1] I have used the built in RStudio commands a couple of times

0.1 What's my current system?

I currently use Dropbox as my main online backup system. I should probably use something more secure like Carbonite, but I recently paid for a year with Dropbox. Using an automatic backup system that keeps versions is a good compromise if you don't want too much messing around with getting started. For some more complicated coding projects I use Git with Github for storage and sharing. For projects I want to keep private, I use Git with BitBucket.