Git Thee To A Repository

Using GitHub and RStudio to manage code

Here’s the plan:

Why do I even care about Git/GitHub?

RStudio gets along with this?

What does this look like in practice?

Why care about Git?

- Git is version control software
  - Version control means every change made to code is logged and you can (read: should) note what each change is
  - This lets you see how code changes over time, revert to early working versions if something breaks, and helps multiple contributors to all work on the same codebase
  - Seriously, the collaboration factor is huge
- GitHub is a hosting service that provides a ton of user friendly features for Git
  - Repository management
  - Web-based GUI

Some terminology

- Repository (aka “repo”)
  - Basically the equivalent to a project folder on your computer
  - Holds all the relevant code, documentation, *etc*. for a given project, *e.g.* a package lives in its own repository or all the code and writing for a paper might have a repository
- Commit
  - A saved change to a chunk of code, but also the act of saving that chunk
  - Always make sure to populate the commit message that says what the commit was!
  - Try to keep commits relatively atomic!
- Push
  - Sending all of the local changes you’ve committed back to the repo to be integrated
- Fork
  - Taking an existing repo and making a snapshot copy for yourself. This might be a speciation event or an interfertile subspecies in which case you’ll deal with...
- Pull request
  - Merging code from a forked repository into one of its siblings. You’re **requesting** that the owner of the repo **pull** in your changes

RStudio integration

First things first, you need to install Git for RStudio to use

Once RStudio knows that Git is installed and ready to rumble, you just need to make a project with version control

If you already have a project and you want to start using version control, it’s far and away easiest to make a new project on GitHub and copy the files over to it

This was seriously life-changing for me

What I’m glossing over

- Credentials!
  - Connecting RStudio to GitHub requires telling RStudio what your login credentials are for GitHub so that GitHub knows that you have the permissions to push commits or issue pull requests
  - It’s possible to save your credentials so you don’t need to enter them every time, thank goodness
  - You’ll also need an SSH encryption key and to make sure RStudio and GitHub are on the same page about it
- Merge conflicts
- Managing branches (families of forks with various purposes)

Check out Happy Git With R for a pretty dang friendly guide to these and more