

Helicopter Introduction to Github¹

Min-Yang Lee

NOAA NMFS Northeast Fisheries Science Center, Woods Hole, MA, USA

March 7, 2022

¹https://github.com/NEFSC/READ-SSB-Lee-WorkingEfficiently/presentations/github_overview.Rmd

Goal:

- ▶ Understand what github is and why it's a useful tool.
- ▶ Use the web editor to make a change to a repository.

Pre-work:

- ▶ Get a github account.
 - ▶ This will take you about 15 minutes of actual time, plus some emails to ITD.
 - ▶ Follow the [github rules](#), which are long and complicated.
- ▶ Look through one of these two pages and find a mistake or a gap in the information.
 - ▶ <https://github.com/NEFSC/READ-SSB-Lee-metadata> or
 - ▶ <https://github.com/NEFSC/READ-SSB-Lee-WorkingEfficiently>
- ▶ Spend 10-15 minutes looking through your own code, emails, 3 ring binders, or documentation. Find a widget that you'd like to share.

What is Github?

- ▶ Github is a tool to help you produce reproducible research.
- ▶ Github is Google Docs for code.
- ▶ Github is track changes for projects.
- ▶ Github has *some* lightweight project management tools. You can:
 - ▶ Track issues and assign them to people.
 - ▶ Aggregate related problems into a project.
 - ▶ Break down a long range project goal into smaller chunks.

Why use it?

- ▶ Collaborate with colleagues.
 - ▶ Work simultaneously and iterate quickly when developing code.
 - ▶ No emailing code back and forth.
 - ▶ Write collaboratively: [Paper](#) and [repository](#)
- ▶ Makes your life easier (maybe) when you do your revisions after 6 months in review.
- ▶ You can't "break" someone's code.
 - ▶ Every version that you tell Github to save is saved.
 - ▶ You can always go back to a previous version. If you've written a good enough note that you can find that version quickly.
- ▶ Continuity when staff turnover

Getting Started with Editing a Document

- ▶ You can use github's online editor for simple things.
- ▶ Just need a github account.
- ▶ Workflow 1:
 - ▶ Edit a document.
 - ▶ Write a *commit message*: a note about what you did
 - ▶ Save it by *committing* to the main *branch*.
 - ▶ This is my favorite for small things? What's small – you know it when you see it.
 - ▶ You may not have permissions to do this

Getting Started II

- ▶ Workflow 2:
 - ▶ Edit a document.
 - ▶ Write a *commit message*
 - ▶ “Save and *Fork*” the repository.
 - ▶ Submit a *pull request*: ask the owner to review and integrate changes.
 - ▶ When the changes are integrated, delete your *Fork*.
 - ▶ You can always do this
- ▶ Workflow 3:
 - ▶ Create a *new branch* in the repository.
 - ▶ Edit a document(s).
 - ▶ Write a *commit message*
 - ▶ Committing to that branch.
 - ▶ Submit a pull request for the owner to review and integrate your changes.
 - ▶ When the changes are integrated, delete the branch.

For more complicated tasks:

- ▶ Install git and either github desktop or Rstudio on your computer (IT-helpdesk)
- ▶ Do the github dance
 - ▶ Pull: get the most recent code from github.
 - ▶ Create a branch: make a duplicate of the code. Give it a snazzy name.
 - ▶ Write code and save (just like usual)
 - ▶ Commit: tell git that you *really* want to save it.
 - ▶ Push: Send your new, snazzily-named branch up to Github.
 - ▶ Pull request: Ask someone review your code and integrate it into the main branch.
- ▶ Lots of guides on the internet on how to use git and github.

Privacy and Control

- ▶ The owner of the repository can control who can see the repository
 - ▶ Anyone
 - ▶ Just certain named people
- ▶ The owner of the repository can control who can make changes to the repository:
 - ▶ Anyone
 - ▶ Just certain named people

A few guidelines

- ▶ *main* should always work. For everyone.
- ▶ No passwords, API keys, PII, or confidential data
 - ▶ Environment variables or
 - ▶ .gitignore
 - ▶ Load in data from Oracle or from locations on the network.
- ▶ Small data on the repository is fine.