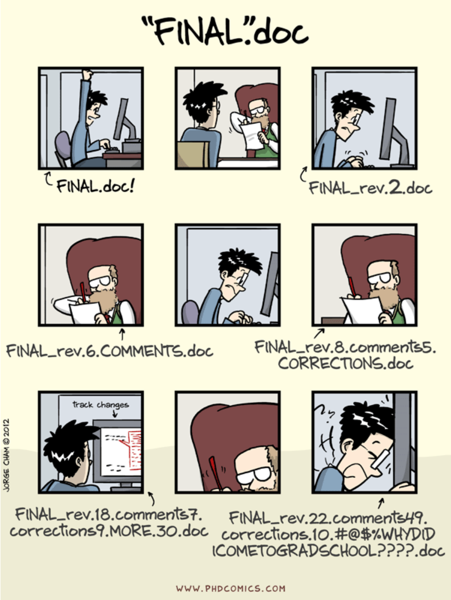
Introduction to Github

Olga Chyzh

February 19, 2019

Why Use Github?



Why Use Github?

- Data analysis produces many files!
- Structure: Github allows for version control via a tree structure
- Organization: You can sync your files to Github from multiple computers
- Backup: Copies of your files are stores online
- Reproducibility/Transparency: Anyone can access your code to learn how you did things
- Collaborations: Multiple authors can work on the same file

Some Terminology

- A Repository/Repo: a directory
- Master: the main version of the file
- A branch: a copy of the master where you can make changes without changing the master
- A fork: a copy of someone else's repository
- Commit: save changes
- Stage: prepare files for uploading to Github
- Pull/Push: download/upload any changes from/to Github to/from your computer

Set Up:

- This is a great reference source.
- Make sure you have a Github account, Rstudio, and Git for Windows or macOS (Install the Xcode command line tools)
- Close and reopen Rstudio, go to `Tools`>`Shell` (may have to go to `Tools`>`Global Options`>`Git` and configure the path to `bin/git.exe` first)
- Configure git with your user name and email (https://happygitwithr.com/hello-git.html)

Let's Try It

- Copy my repo's (IntroGithub) HTTPS clone URL to your clipboard via the green "Clone or Download" button
- In Rstudio start a new Project: `File > New Project > Version Control > Git`
- In the "repository URL" paste the URL of your new GitHub repository

Create You Own Branch to Work In

- In the shell, type: `git branch yourname`
- Now you can switch to your brach (upper right corner) and edit the file without affecting the master.
- Switch to your branch, make some edits, save them
- To have Github track your branch, open the shell and type `git push -u origin your_branch_name`
- Now you can Pull/Push your branch (always Pull before your Push).

Daily Workflow

- Once your project or a part of it works (e.g., a script produces an output without errors), stage your changes and make a commit
- Enter a meaningful commit message (e.g. “WIP” for “work in progress”)
- Do some more work, make sure the project still works, make another commit, but this time check the ‘amend the previous commit’ box
- Once you are ready to share your progress with others, commit amending again, but enter a meaningful commit message. Push.