Github: A Collaborative Code-Sharing Tool

Nelsha Athauda

2022-10-18

What is Github?

"The complete developer platform to build, scale, and deliver secure software."

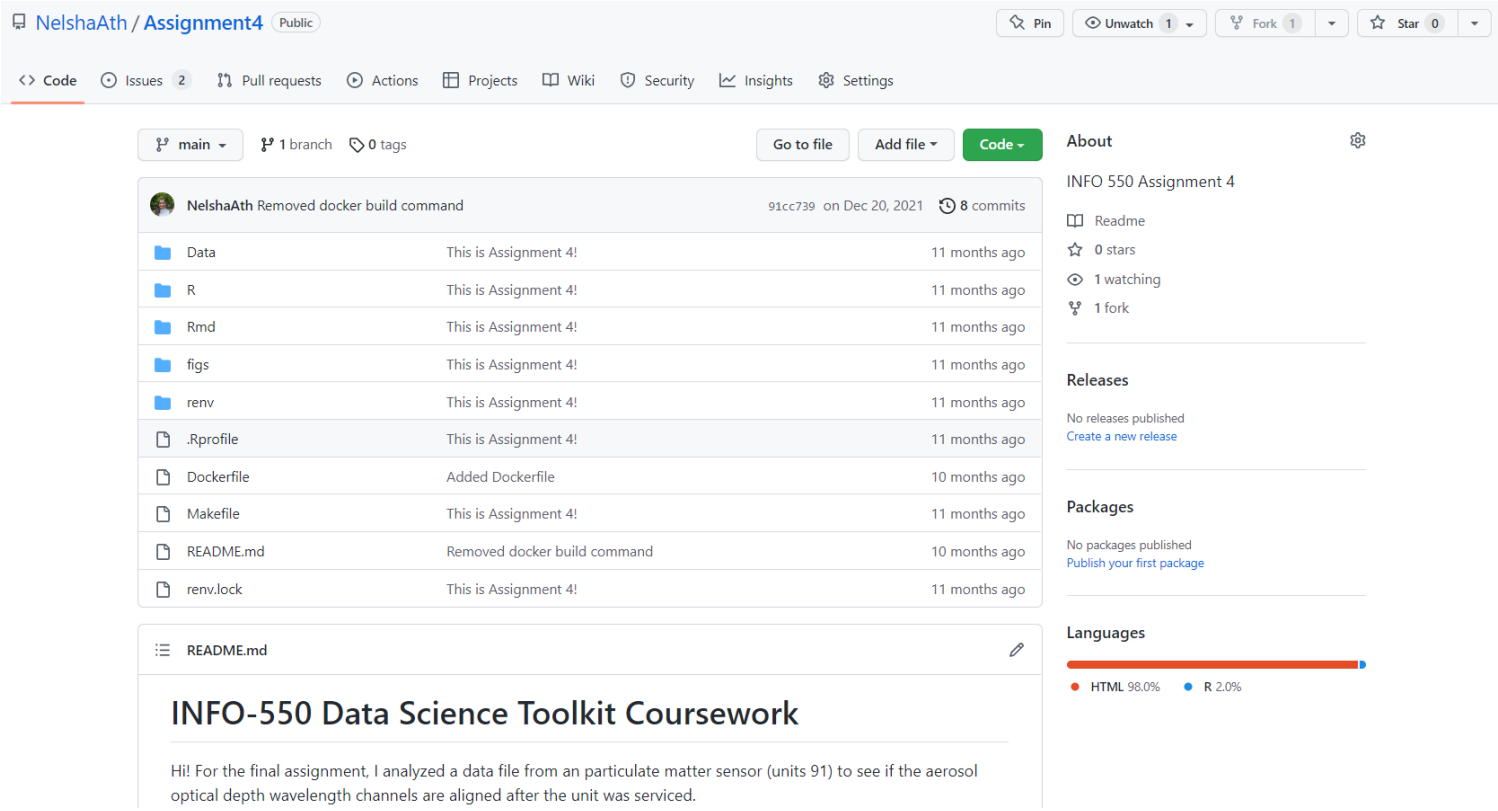
- Data Analysis
- Version History
  - See history of changes made to a document
  - Revert to older versions of a document
  - Try new things out without breaking things that work
- Collaboration

tl;dr: Stop saving fifty copies of the same file with different names

Terminology

- Repository
  - A directory of (ideally) plain-text files (.html, .txt, .R, .Rmd, etc.) included in a project
- Commit
  - A unique flag for a particular state of your project
  - Every push to the server is an updated version = a new commit
- History
  - All of the commits for a project
- Branch
  - An independent line of commits of your repository -Fork
  - A copy of someone else's GitHub repository that you can edit

Example repository



This is a repository of the code that I wrote for my Data Science Toolkit Final Exam. It compares data across two collocated air quality samplers. The repository is the directory of files. The README.md file describes the project and how to run it.

Example commit

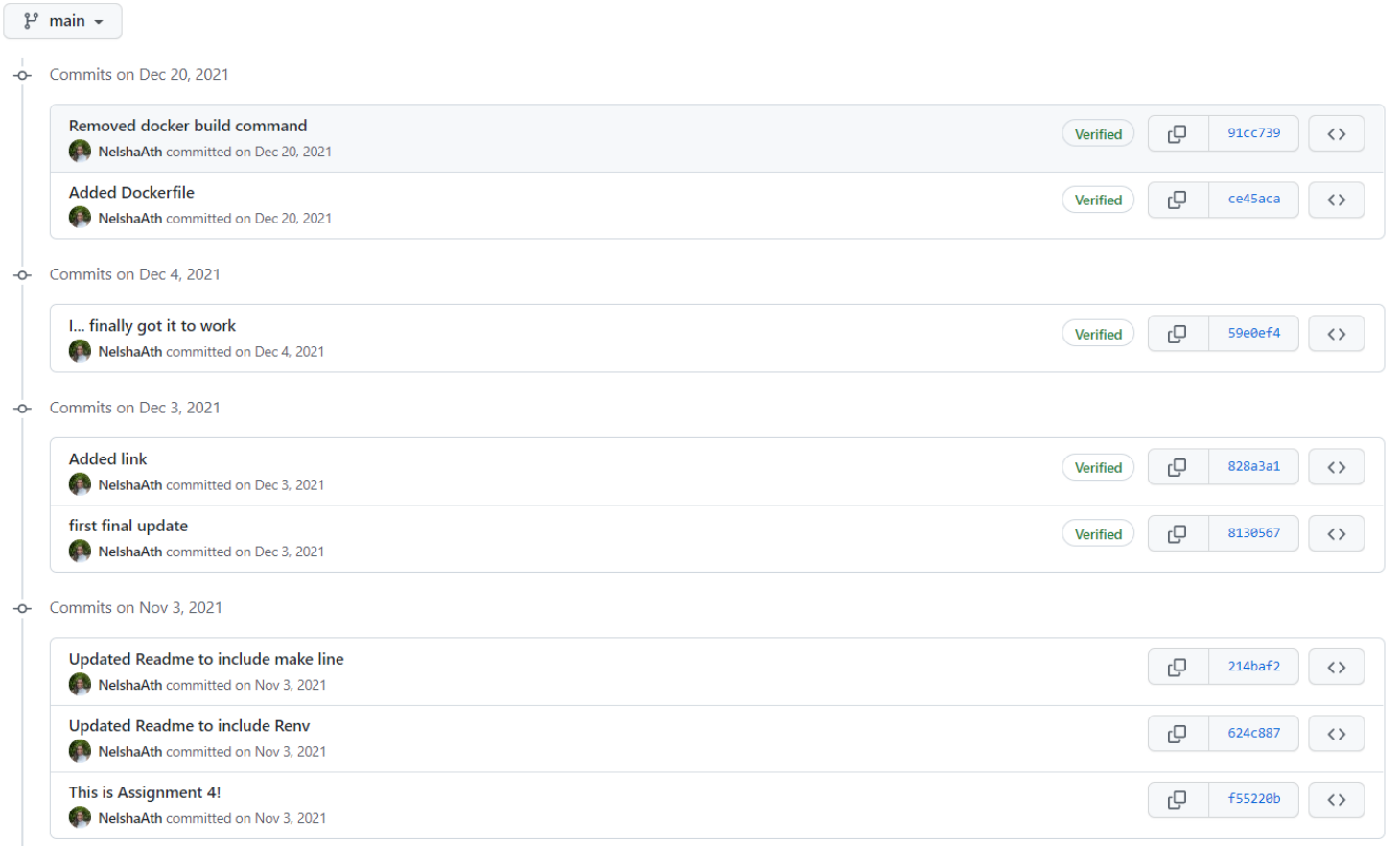
Showing **1 changed file** with **3 additions** and **7 deletions**.

10 README.md

```
@@ -14,22 +14,18 @@ A makefile is in the main project folder which is able to generate the report wi
14 14  make
15 15  ```
16 16
17    - ## Final Assignment :D
   17 + ## Final Assignment
18 18  Link to Docker Hub Image: https://hub.docker.com/repository/docker/nelshaath/info_550
19 19
20 20  To download the Docker image of this project, pull the image in your terminal:
21 21  ```
22 22  docker image pull nelshaath/info_550:latest
23 23  ```
24    - and then build the image
25    - ```
26    - docker build -t info_550 .
27    - ```
28 24  To see the output you will need to mount the directory to a local folder on your device. You must change the path to a folder on your device!!!
29 25  ```
30 26  docker run -v /path/to/project/R:/info_550/Rmd -it info_550
31 27  ```
32 28
33    - The output is the Report.html file <3
   29 + The output is the Report.html file
34 30
35    - Please email me at Nelsha.Athauda@emory.edu if you're having any problems! Happy end of term~
   31 + Please email me at Nelsha.Athauda@emory.edu if you're having any problems!
```

If you click on a commit on GitHub, you can see what lines were modified. The old version appears in red and the new version in green.

Commit History



This is the full project history, each commit is a different version.

Using a terminal

Unless you want to upload your files directly to the website (inconvenient), you will need to learn how to use your computer's terminal.

All devices have a command line (Command Prompt) but you can download a WSL (Windows Subsystem for Linux) like Ubuntu which can download R itself and be used as a one-stop console and terminal.

WSL or Command Prompt can be selected in your IDE (I use RStudio or Sublime Text Editor 4)

Starting with Github

Go to https://github.com/ and create a GitHub account. Also go into your settings and create a PAT (Personal Access Token) which will be required to access your Github account from a terminal.

Copy and paste your PAT somewhere easy to find or you'll have to make a new one!!

Using Simple bash commands to create a local directory

```
cd ~/Desktop # Change Directory to where your project is
mkdir myproject # Make directory titled ___
cd myproject/ # Navigate to the project

git init # Initialize repository
```

Important: For any file addresses, use forward slashes only! A back slash in bash means to disregard the next character (used if you have spaces in the address)

Copied from File explorer: C:\Users\nrathauda\OneDrive - University of Alaska\R

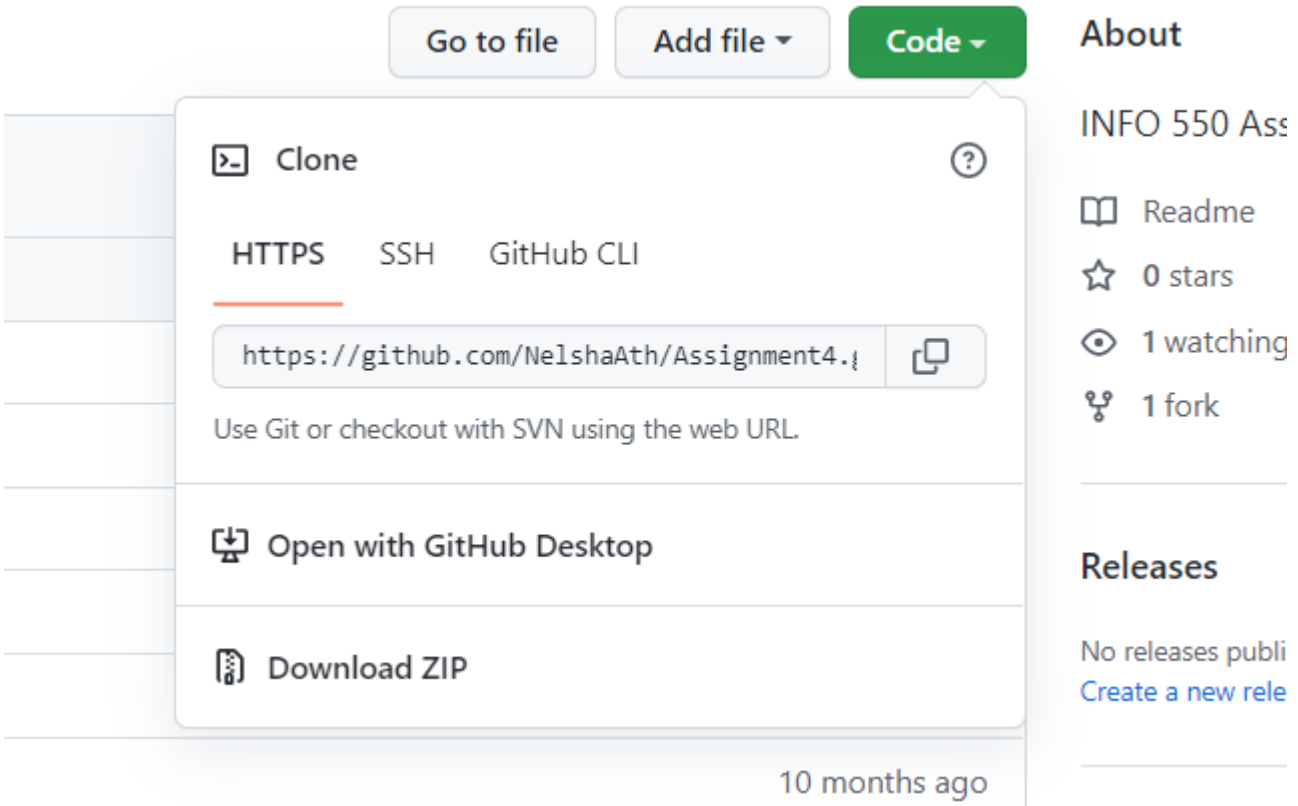
Adapted to bash: C:/Users/nrathauda/OneDrive\ -\ University\ of\ Alaska/R

Maintain your local repository

```
git add # tell git which files to add to this commit
git add . # The period will add all files in the folder
git commit -m "I fixed so many things!" # make a commit with a messag
git status # shows what files have changed since your last commit
```

Create a online repository (GitHub)

GitHub is the online platform. Create a repository on the website and copy the HTTPS share link to create a remote (bridge) between the Hub and your local repository



Pushing your local git repo onto your online GitHub repo

```
git remote add origin https://github.com/[your-username]/[repository-
git push -u origin main
# Pushing your origin (local repo) to the Hub (online repo) on a brar
```

After sending git push, your terminal will ask for your GitHub username and password/PAT

Check your GitHub profile

If everything pushed successfully, there should be a recent commit!

Using Branches

If you want to try something risky but don't want to mess up your current working code you can use branches Commit the working version on your normal branch (Main) Start working on your risky changes but commit the updates to a new branch (Ex. Branch1) If your new branch works out, you can merge the changes you made onto your original Main branch

```
# make a branch called Branch1
git branch Branch1
# checkout is used to switch between branches
git checkout Branch1
# see all branches
git branch
```

Using Branches (cont.)

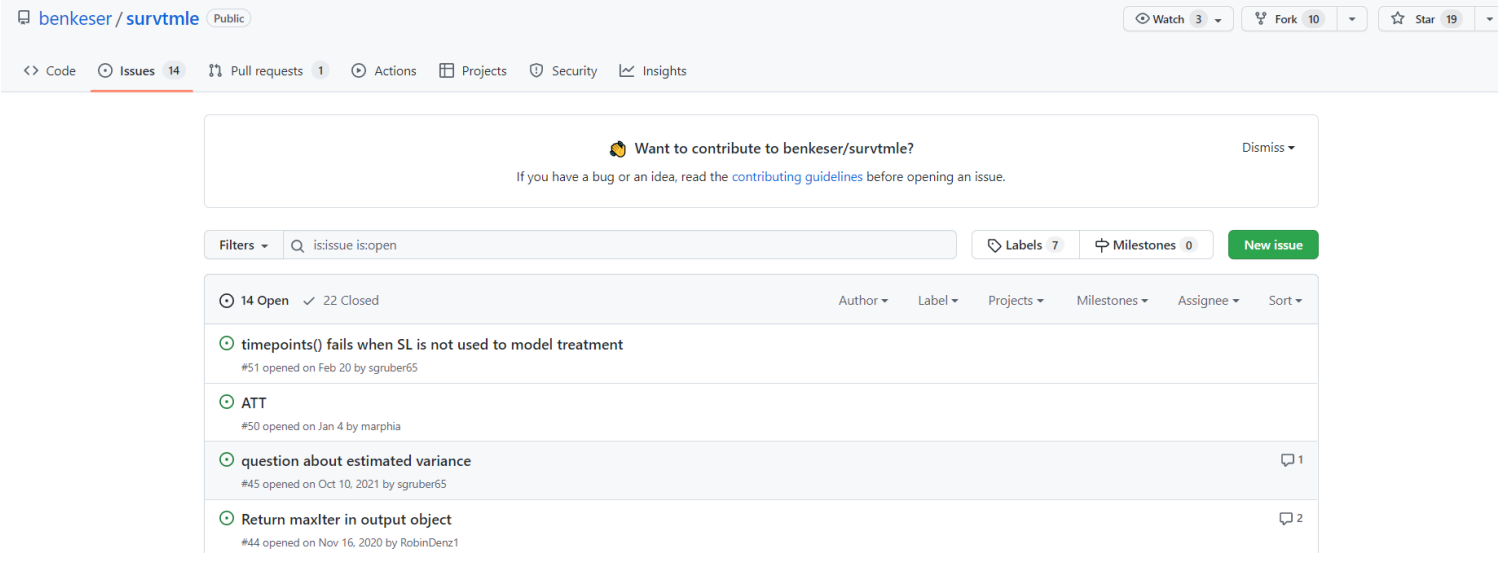
Both branches can be edited, merging will add/delete to the branch you want it to act on

```
# Switch back to the original branch
git checkout Main
# Merge the changes on Branch1 to Main
git merge Branch1
```

Merge conflicts will come up if you edited the same lines in different branches and will have to be resolved manually (git will show you which lines)

Other ways to collaborate

If you notice a bug in someone's code you can file an issue (Similar to a YouTube comment)



Or you can create a pull request

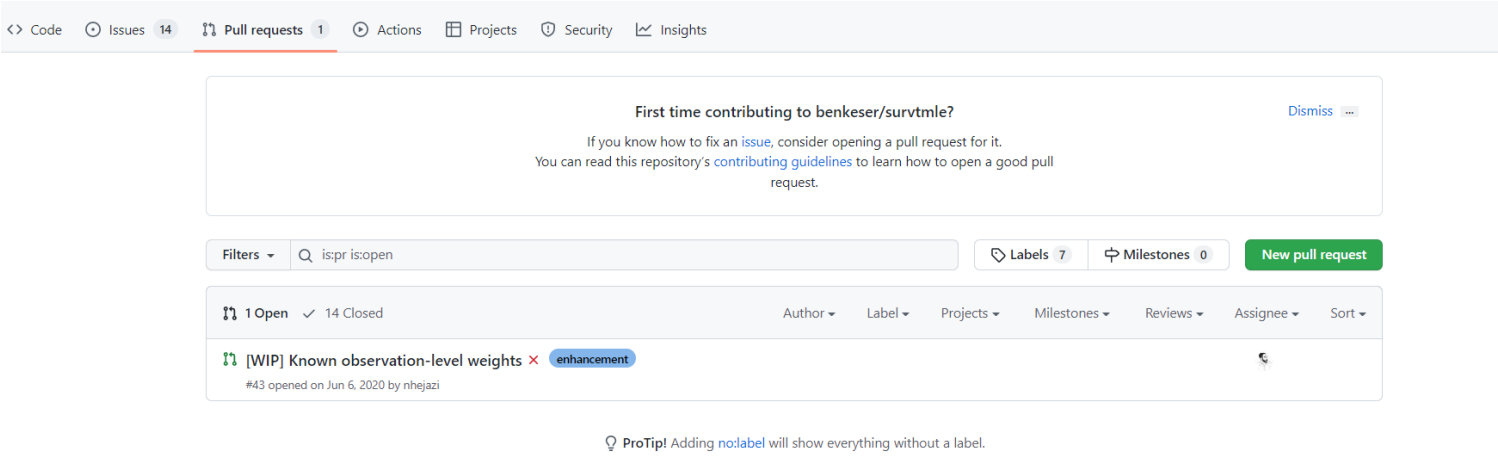
- Fork the repo on GitHub
- git clone to download to local machine
- Modify code (read contributing guidelines first!)
- Commit changes
- git push back to GitHub

Pull Request Workflow

```
# add friend's repo as a remote branch
git remote add wrongdoer https://github.com/wrongdoer/repo
# downloads friend's branch, but do not merge it yet
git fetch wrongdoer master
# view all local and remote branches
git branch -a
# checkout friends remote branch
git checkout remotes/wrongdoer/master
# make a local branch based on friend's repo
git checkout -b wrongdoer
# test out the branch; make sure it works as expected
[...]
# checkout local master, merge, push
git checkout master
git merge wrongdoer
git push origin master
```

Now the fixed code is in YOUR repository

To give the fixed code to the person who created the version with a bug, create a pull request on Github, in their repository



On the receiving end of a pull request?

Merge it on GitHUb itself To add it to your local copy, git fetch or git pull

git fetch

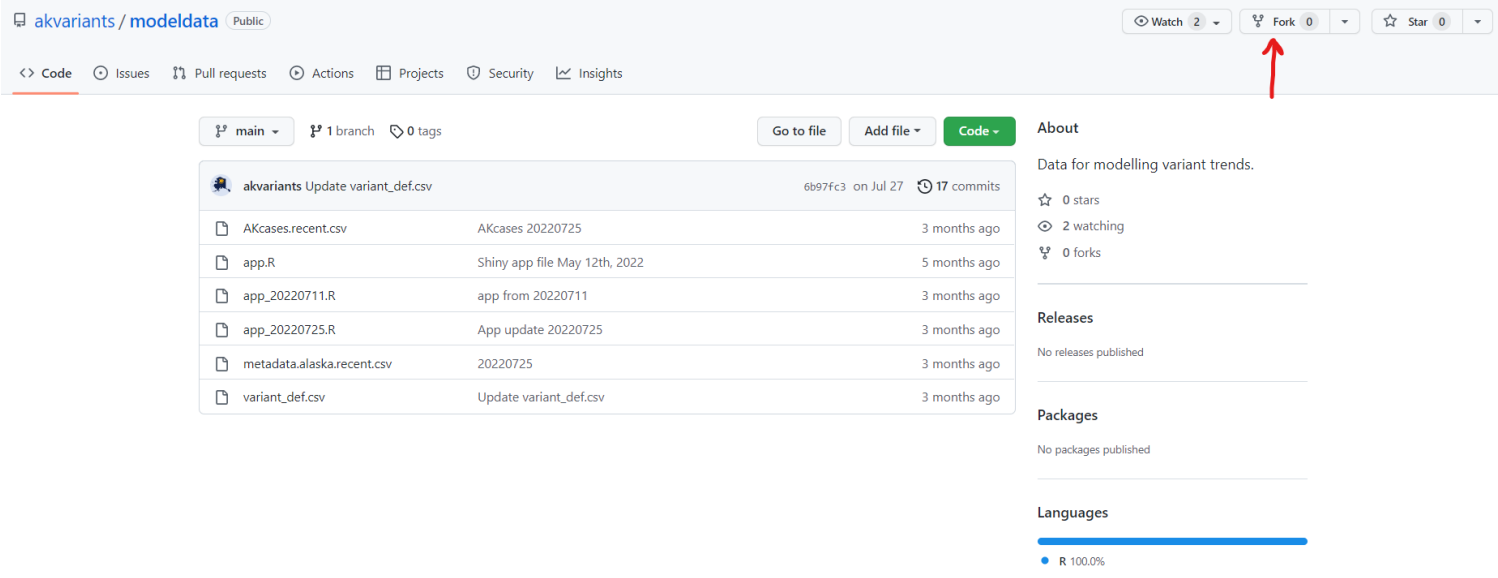
- Download code separately from your local repository.
- Explicitly merge into your local repository.

git pull

- fetch and merge at the same time
- Faster, but you can't verify that the code works before changing your local repository.

Want to see/use/adapt someone else's code?

Find the repository on GitHub and fork it to create a copy that you can alter.



To bring the copy to your local device, use git clone

```
git clone https://github.com/Your-name/The-Forks-name
```

Great Job!

Git has an incredibly steep learning curve but it's version control and reproducibility features make it worth it.