NYU Center for Data Science

Lab 1: Git and GitHub

Who here is using GitHub regularly (raise your hand)?

What is Git?

Git and Version Control

- **Version Control Systems (VCS)**
    - tools that store different versions of project files
    - revert to earlier versions

- **Git** is a **distributed** VCS

- Why might you want to use a version control system?

What is GitHub?

Github and (large scale) collaboration

- **GitHub** is a website that hosts git repositories online
    - We'll use it in this class for code submissions for the three class projects
- In industry, working collaboratively is the norm, and Github is a standard way of allowing people to work on code collaboratively
- You can share your code with the world, and they can make independent contributions to it.
- It reinforces reproducibility standards (more on this later, e.g. README.md files)

Basic Github functionality: Commits, forks, cloning, push, pull, etc.

Basics (Forking a repo)

- A **Fork** refers to a copy of a repository.
  - Forking a repo lets you experiment with a project without affecting the original repository.
- One can fork a repo by clicking the fork option at the top right corner of a github repository page
- For more information, follow this link: https://help.github.com/articles/fork-a-repo/

Basics (Forking a repo)

- After forking the repo, click on the '**Clone or download**'
- Paste the link in the following command on the terminal (Mac), command line (Linux/WSL) or Git Bash (Windows):
  - **git clone <paste link here>**

Basics (Making changes to a repo)

- **Pull**: Use **git pull origin main** to pull any latest changes from the forked repo to your local copy.
- **Status**: Use **git status** command to see the staged (shown in green) and un-staged (shown in red) files in your local repository.
- **Staging**: Use **git add <filename>** to stage a changed file for commit
- **Commit**: Use **git commit -m "<your message here>"** to commit the staged files.
  - Keep your message short, descriptive and specific.
- **Push**: Use **git push origin main** to push all the changes made locally to the origin.

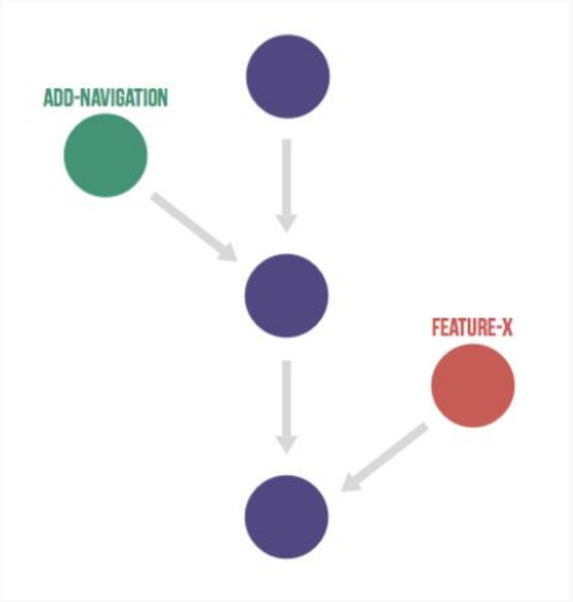
Some conceptual nuances: What’s the difference between

-forking and cloning?

-push and commit?

Basics (Branching & Merging)

- **Branching**:
  - **git switch –c <branch name>** to create a new branch
  - **git switch <branch name>** to switch to a different branch
- **Push**:
  - **git push origin <branch name>** to push any changes made on this branch.
- **Merging**:
  - **git merge <branch name>** to merge changes in <branch name> to your current branch.



Why do such a thing?
Why would one create a branch in the first place?

Known pain point: Technically, GitHub only recognizes files, not folders

Basics (logging)

- **Log**: Use **git log <options>** to view the history of changes

- Different options, e.g.:
    - **git log --help**
    - **git log --decorate --all**

More references available: https://swcarpentry.github.io/git-novice/

Setting up git

- Go to https://github.com/
  - Sign Up/Create new account if you don't already already have one.
- Mac and Linux users generally have git already installed on their system
- For Windows users follow instructions on this link:
  - Windows Subsystem for Linux: https://docs.microsoft.com/en-us/windows/wsl/install-win10
- OR install git on Windows:
  - https://help.github.com/articles/set-up-git/#setting-up-git