Causal Inference

Mixtape Session

MIXTAPE SESSIONS

Roadmap

- Hidden curriculum
    - Background
    - Empirical workflow
    - Hierarchical folder structure
    - Naming conventions
    - Version control
    - Soft skills

Version control

1. hierarchical folder structure
2. automation
3. naming conventions
4. **version control**

Version control

- “Version control” is the practice of tracking and managing changes to software code and documentation
- Old school version control was to use a suffix attached to the filename, maybe with date and author abbreviation (`Paper_Revision_V3_SC_May052021.tex`)
- Creates dozens and maybe more local versions which depending on the project may make management difficult and therefore allow for errors
- As software production became more collaborative and complex, new version control systems were created so that we didn’t have to use the date/author suffix system
- Git is the most popular version control sytem

What is git?

- Open source project developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system (also open source)
- Since 2005, Junio Hamano has been the core maintainer
- Many many many projects rely on git for version control, including commercial projects
- Git is “distributed architecture”, like “deep” software (e.g., operating system, programming language), installed on your local machine
- Used to coordinate production, editing and maintenance of software
- Nice because it allows for simultaneous editing by multiple sources, and reconciling differing versions, as well as distributing those versions back to people

Git and Github

- Git and Github are related but distinct
- **Git** is a mashup between Dropbox and Microsoft “track changes” (credit: Grant McDermott)
- **GitHub** is a company that provides services that are built on git
- **Imperfect analogy**: Github is to git what beer bottles are to beer (not exactly, but almost)

Github repository

- Think of it for now as like your Dropbox folder in that it’s like a folder stored in the GitHub servers
- A repository contains all of your project’s files and each file’s revision history.
- You can discuss and manage your project’s work within the repository.
- It tracks all changes made to files in your project, building a history over time.
- If you delete the .git/ folder, you have deleted your project’s history.

Getting started

- Set up git (steps not shown): `https://docs.github.com/en/get-started/quickstart/set-up-git`
- Create a free account at github (steps not shown): `https://github.com`
- Download github desktop (next)
- Set local path (next) – do *not* put it in dropbox or cloud based directory. Put it in something like /Documents or /Desktop
- Clone a repo (next)
- Go through git operations (next)

The Four Git Operations

1. **Staging**: tell git you want to add changes to the repo history
2. **Commit**: tell git you are sure that you want this particular thing you did to be part of the repo history
3. **Pull**: tell git to get any new changes made on the github repo and put it on your local machine (hence “pull”)
4. **Push**: push any local stuff in your .git/folder to the github repo which allows then anyone else to pull it