Tips and Tricks for Data Management and Analysis

Sean Sylvia

February 14, 2024

Goal: Replicability

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

Why?

- Science
- Efficiency (they call it REsearch for a reason)
- Journal revisions
- Sanity

Key resources

- "Code and Data for the Social Sciences: A Practitioner's Guide" by Matt Gentzkow and Jesse Shapiro
- IPA's Best Practices for Data and Code Management
- Many others on good coding practices...

Outline

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

1. Folder and File Structure
2. Coding Tips
3. Version Control

Folder and File Structure

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

- Use a logical folder structure for project
- Several ways to do this, rule is to keep like files together
- Organize into inputs and outputs

My usual structure for "Analysis" folder:

- Code
  - Programs
  - Project
- Data
  - Raw
  - Working
- Results
- Archive
- Scratch

Coding Tips

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

1. Keep data with unique key as long as possible
2. Use a "master do file" (automate everything)
3. Use relative references
4. Abstraction

Keep data with unique key as long as possible

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

- Often work with data at different levels (clinics, docs, etc)
- Best to keep these in separate databases with unique "keys" (IDs) as long as possible
  - "Normalized" data is easy to understand

Steps:

1. Store data in normalized files (think of preparing for release)
2. Construct second set of normalized files that include transformations, analysis variables, etc
3. Merge together to create analysis dataset

Use master do-file (Automate everything)

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

- Automate as everything you can: from raw data to tables, figures
  - Reduces errors
  - Automatically documents each step in analysis

- Keep "Master" do file:
  - Runs all do files associated with project in order (defines pipeline)
  - Can contain general project information and what each file does
  - List all user written commands and explains

Use relative references

VS. absolute reference

- Absolute reference: "/Users/sean/Dropbox/Project/Data/data.dta"
- Instead:
  global datadir "/Users/sean/Dropbox/Project/Data"
  use $datadir/data.dta

- Makes folders portable
- Easy to collaborate

Abstraction

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

- Abstraction: Turning specific instances into a general purpose tool
- Use abstraction to
  - Shorten code (less cutting and pasting chunks)
  - Improve Clarity
  - Reduce editing errors
- E.g. define "program" in Stata

Version Control

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

Git

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

What is Git?

- (Distributed) Version control system
- Like a remote "track changes"
- Optimized for code
- Allows for systematic collaboration on data analysis
- Initially takes some commitment (but totally worth it)

Git vs. Dropbox?

- Versions of project vs. versions of each file
- Easily see changes with 'diff' comparing old and new code
- Each version tied to meaningful change
- Collaborations: Branching to experiment, ask for review

GitHub

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

- Online hosting platform for Git (Bitbucket is another)
- Not necessary, but makes things easy
- Desktop application for workflow

Basic steps:

- Create repository (from a directory on your computer)
- Make a "commit"
- "Pull"/sync any changes
- Work on code
- Commit

Can also make "branches": Great for collaborating

Setup

- Sign up for a GitHub acct:
  - Pro version FREE for students
- Download and install GitHub Desktop
  - Note there are other GUI that are more fully featured. GitHub Desktop is easy to use
- Authenticate to GitHub
- configure default editor
- Done!

Create New Repository

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

- File – New Repository
- Fill Fields
- Warning: Dropbox and GitHub don't play well together, put repository in unsynced folder
- Initialize with a ReadMe file
- setup .gitignore file
- Download .xls and .do file from my repository
- Place in repository

First Commit

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

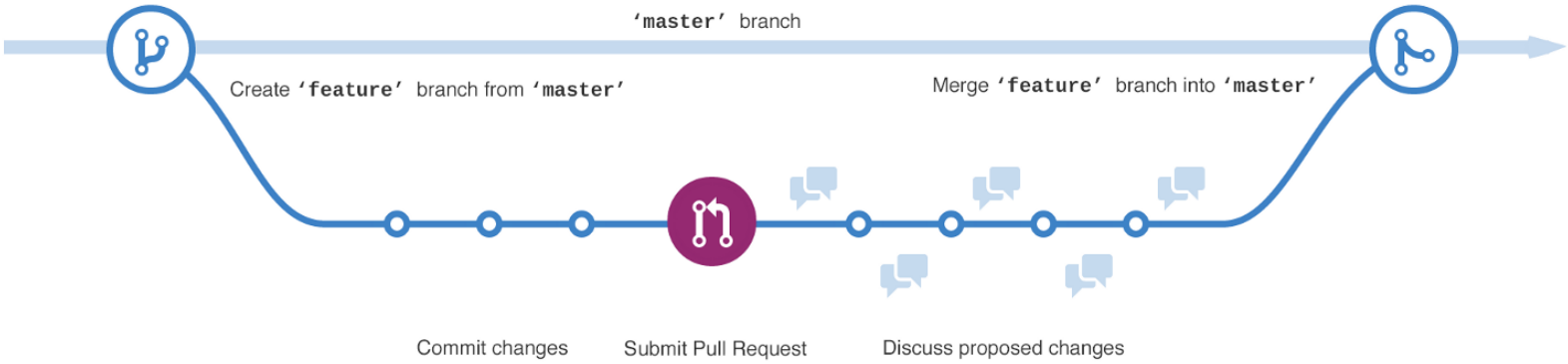
- See changes tracked in GitHub Desktop
- Publish to GitHub
- View it on github.com

Make, commit, and push changes

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

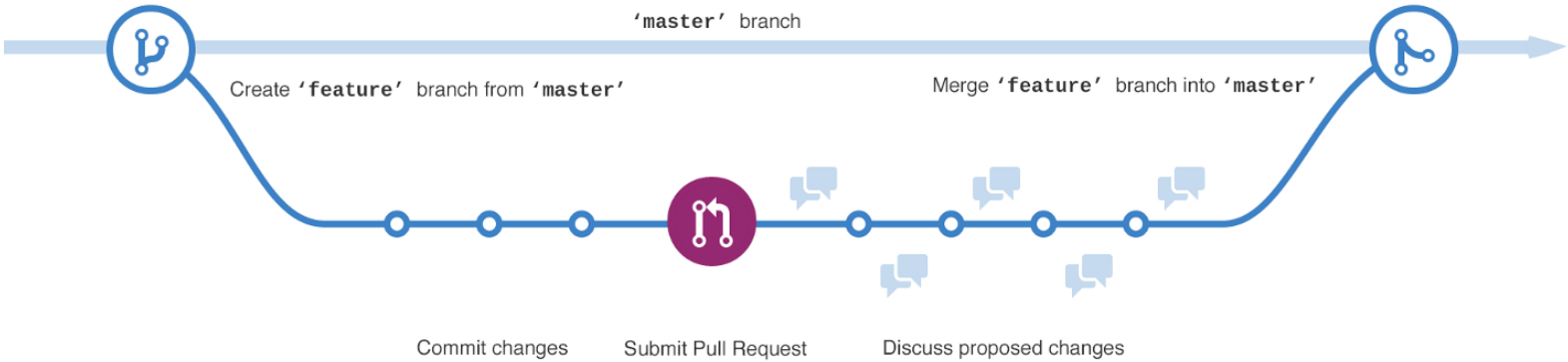
- Open do file in preferred editor
- Make a change to .do file
- Add text to ReadMe
- See changes tracked in GitHub Desktop
- Add commit summary in Desktop changes list
- Commit change to Master
- Push changes to GitHub

Workflow



- Branch
- Add commits
- Pull Request
- Code Review and discussion
- Merge into master branch

Practice



- Create a Branch
- Make changes
- Commit changes
- Open Pull Request
- Code Review
  - Review changes
  - Close Pull Request
- Merge into master branch
- Can delete branch if desired

Git Lingo

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

- Fork Make a copy of someone elses GitHub repo in your own GitHub account.
- Clone Make a copy of the your GitHub repo on your local computer. In CLI: git clone copies a remote repo to create a local repo with a remote called origin automatically set up.
- Pull You incorporate changes into your repo.
- Add Adding snapshots of your changes to the Staging area.
- Commit Takes the files as they are in your staging area and stores a snap shot of your files (changes) permanently in your Git directory.
- Push You push your files (changes) to the remote repo.
- Merge Incorporate changes into the branch you are on.
- Pull Request Term used in collaboration. You issue a pull request to the owner of the upstream repo asking them to pull your changes into their repo (accept your work).

Collaboration Exercise

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

- Choose a partner
- One person: Add collaborator on GitHub (Settings – collaborators)
- Collaborator clones owner's repo
  - Accept access to owner repo on GitHub.com
  - Sync in GitHub Desktop
- Collaborator makes changes to do file
  - add covariates to regression
  - save file
  - commit changes
  - submit pull request
- Original Owner comments on GitHub, discussion over changes, new commit
- Close Pull request
- Merge to Master

Conflict Resolution

UNC GILLINGS SCHOOL OF GLOBAL PUBLIC HEALTH

- Person A
  - Add another covariate on regression line
  - Commit and Push changes to GitHub
- Person B
  - Modify same line of do file without updating from GitHub
  - Commit changes locally
  - Submit Pull request
- Both: On GitHub, got to Pull requests – Resolve Conflicts
  - Decide to a) keep only your branch's changes, b) keep only the other branch's changes, or c) make a brand new change
  - Delete the conflict markers `<<<<<<<`, `=======`, `>>>>>>>`
  - `Once resolved, click "Mark as Resolved"`
  - `commit merge`
  - `merge pull request`

Other Resources

UNC
GILLINGS SCHOOL OF
GLOBAL PUBLIC HEALTH

- There are alternative workflows besides the feature-branch workflow
- If you want to look cool, you can use the command line for Git
- TONS of resources on the web for more advanced use (but mostly unnecessary)
- **Conclusion:** Some fixed cost to learn, but well worth the time in the long run