Git for Social Scientists: Introduction to Version Control with Git

Tomoya Sasaki

Massachusetts Institute of Technology

November 4th, 2022

Introduction

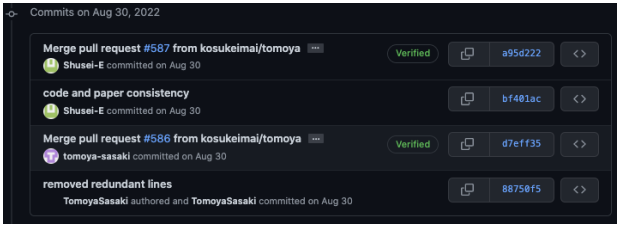
- Why Git? Why Github? Why version control?
- These are essential tools for programmers
- How about social social scientists?
- My opinion: Social scientists also benefit from version control with Git
  - Increase in collaborative projects
  - Demand for clean replication materials
  - Complex data manipulation/preprocessing/analysis
- "Code and Data for the Social Sciences: A Practitioner's Guide" by Gentzkow and Shapiro has a chapter dedicated for version control
- This workshop
  - Introduction to version control
  - Pros and cons of Git/Github
  - Brief introduction of these tools

What is version control?

- Version control: tracking and managing changes to file content
- Git: (the most popular) software for version control
- Github: service to host your git on the Internet (alternatives include GitLab, Bitbucket ...)
- Repository: unit of a version control project, contains a folder with a subfolder named `.git`
  - Local repository: repository (folder) in your own computer
  - Remove repository: repository (folder) in a web hosting services such as Github
- `.git` folder in a repository tracks and stores every single change you make in the corresponding repository
- I focus on Git and Github because they are extremely popular than their alternatives

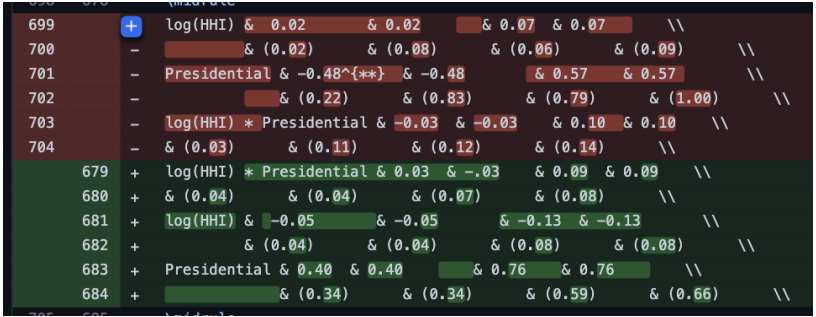
Why Git and Github: Tracking who/how/when

- You can identify
  - who made changes
  - how they made the changes
  - when they made the changes
- You can check the entire history since you created a repository and move back to previous versions easily
- Github can visualize them nicely
- Useful when you
  - want to revert your (particular) changes
  - work on a collaborative project
- You don't need to keep
  - different versions of the same file: `clean_data_1104.R`, `clean_data_1020.R`
  - the same file edited by different people: `clean_data_tomoya.R`, `clean_data_adam.R`



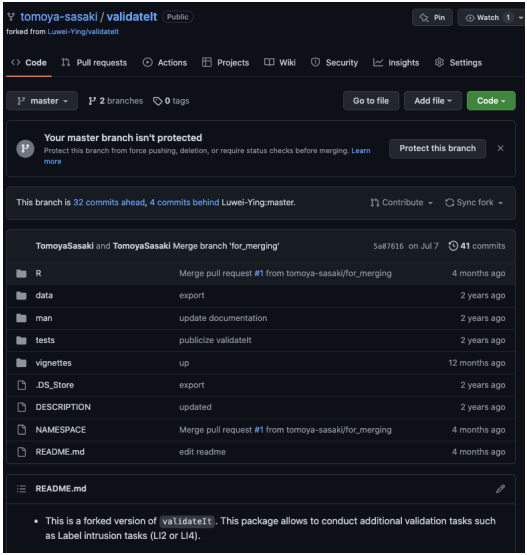
Why Git and Github: Tracking who/how/when

- You can check how the results change when we try different specification
- Easy to track which part of the results changed



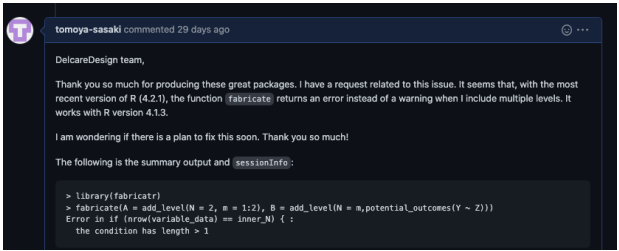
Other side benefits of Git/Github

- Hosting a customizable website (free, no ads, tons of templates)
- Contribute to software packages hosting on Github
- Tweak a package developed by someone else for your own purposes
- Send a request to package developer (often happens at "Issue")
- Nice integration with popular apps/websites such as Rstudio and Overleaf



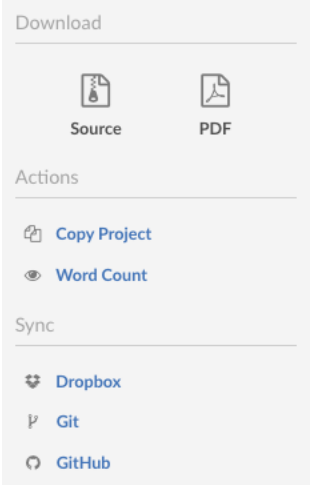
Other side benefits of Git/Github

- Hosting a customizable website (free, no ads, tons of templates)
- Contribute to software packages hosting on Github
- Tweak a package developed by someone else for your own purposes
- Send a request to package developer (often happens at "Issue")
- Nice integration with popular apps/websites such as Rstudio and Overleaf



Other side benefits of Git/Github

- Hosting a customizable website (free, no ads, tons of templates)
- Contribute to software packages hosting on Github
- Tweak a package developed by someone else for your own purposes
- Send a request to package developer (often happens at "Issue")
- Nice integration with popular apps/websites such as Rstudio and Overleaf



Limitations: not suitable to track large files

- Github imposes file size limits:
  - 25 MB per file limit (you can change this limit up to 100MB by changing setup)
  - 1GB per repository limit
- Remember that `.git` tracks and stores all the change you make in a repository

  $\rightsquigarrow$ if you store a huge file in the repository and let `.git` tracks its changes, the `.git` folder can grow quite huge

- Use `.gitignore` to specify files that Git should ignore

.gitignore

```
*.csv # ignore csv files
/data/ # ignore data folder
```

- Include huge files as well as sensitive files (password, API key etc) in `.gitignore`

Limitations: difficult to track non-text files

- Git cannot track line by line changes for non-text files such as PDF, Microsoft Word/Excel/Powerpoint, JPG, ...
- Note that Git still tracks changes
- The value of Git/Github is limited
- In the left example, Git/Github recognizes the changes as the changes in file sizes
  $\rightsquigarrow$ even though you update a figure in PDF or PNG format, Git/Github might not recognize it unless the file size changes...