

Git for Social Scientists: Introduction to Version Control with Git

Tomoya Sasaki¹

Massachusetts Institute of Technology

November 4th, 2022

¹This workshop is inspired by workshops by Suyeol Yun and Shiro Kuriwaki.

Introduction

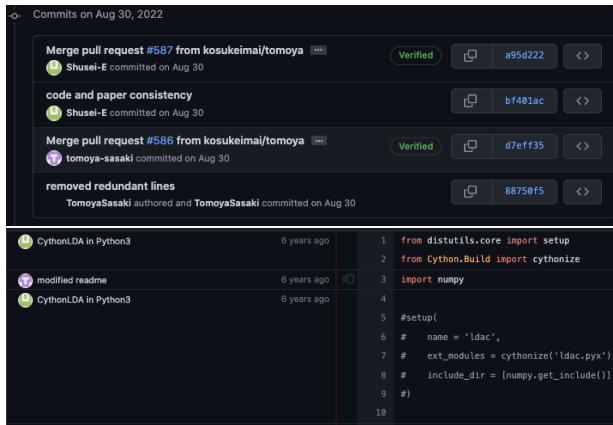
- Why Git? Why Github? Why version control?
- These are essential tools for programmers
- How about social scientists?
- My opinion: Social scientists also benefit from version control with Git
 - Increase in collaborative projects
 - Demand for clean replication materials
 - Complex data manipulation/preprocessing/analysis
- “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
 - Remote 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/Github because they are extremely popular than their alternatives

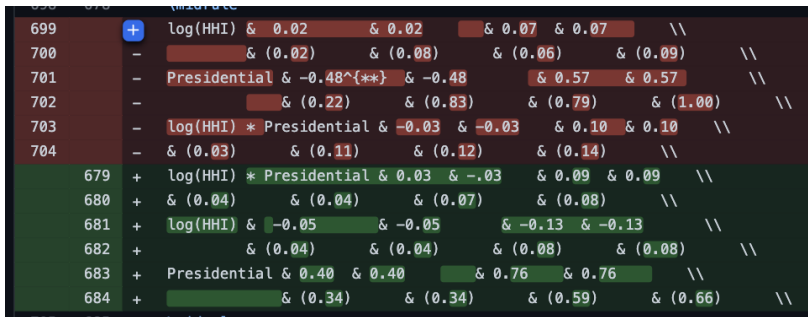
Benefit of Git/Github: Tracking who/how/when

- 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 (undo changes)
- 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



Benefit of Git/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

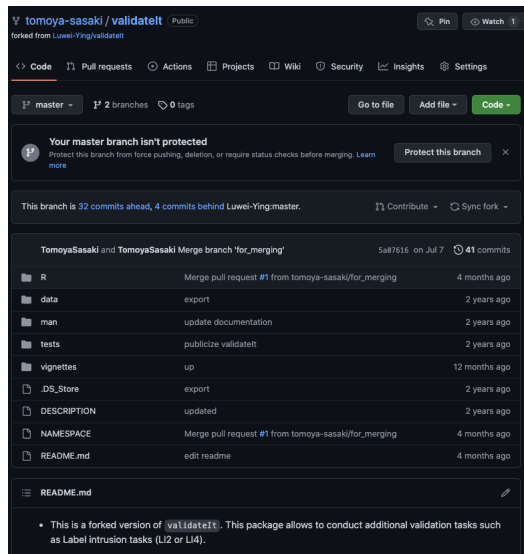


The image shows a screenshot of a regression results table, likely from a software package like Stata or R, with Git diff highlights. The table has columns for coefficients and standard errors. The rows are numbered 699 to 704 and 679 to 684. The first column contains a sign (+ or -) and the second column contains the variable name. The subsequent columns contain the coefficient and its standard error in parentheses. The last column contains a double backslash (\\). The rows are grouped into two sections: 699-704 (top) and 679-684 (bottom). The top section (699-704) has a blue '+' sign in the first column, and the bottom section (679-684) has a green '+' sign in the first column. The Git diff highlights are shown in red and green boxes around the coefficients and standard errors.

699	+	log(HHI)	& 0.02	& 0.02	& 0.07	& 0.07	\\
700	-		& (0.02)	& (0.08)	& (0.06)	& (0.09)	\\
701	-	Presidential	& -0.48^{**}	& -0.48	& 0.57	& 0.57	\\
702	-		& (0.22)	& (0.83)	& (0.79)	& (1.00)	\\
703	-	log(HHI) * Presidential	& -0.03	& -0.03	& 0.10	& 0.10	\\
704	-		& (0.03)	& (0.11)	& (0.12)	& (0.14)	\\
679	+	log(HHI) * Presidential	& 0.03	& -0.03	& 0.09	& 0.09	\\
680	+		& (0.04)	& (0.04)	& (0.07)	& (0.08)	\\
681	+	log(HHI)	& -0.05	& -0.05	& -0.13	& -0.13	\\
682	+		& (0.04)	& (0.04)	& (0.08)	& (0.08)	\\
683	+	Presidential	& 0.40	& 0.40	& 0.76	& 0.76	\\
684	+		& (0.34)	& (0.34)	& (0.59)	& (0.66)	\\

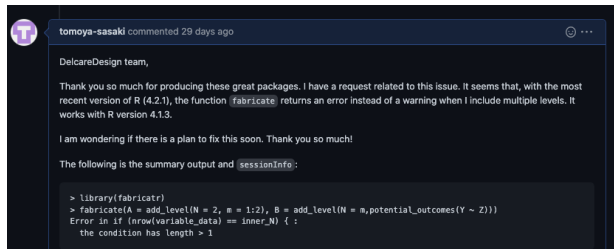
Other side benefits of Git/Github

- 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



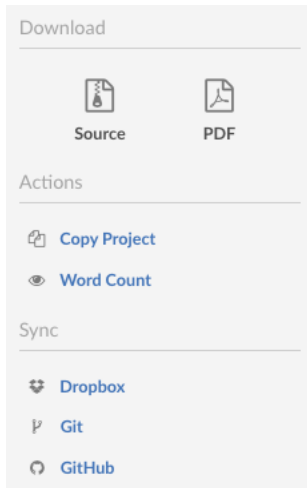
Other side benefits of Git/Github

- 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 limit:
 - 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
~> 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 not track “ignore”

```
*.csv # ignore csv files  
/data/ # ignore data folder
```
- Include huge files as well as sensitive files that contain password, API key etc in `.gitignore`

Limitations: not great if you want 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
→ even though you update a figure in PDF or PNG format, Git/Github might not recognize it unless the file size changes...



How Git/Github tracks files

Comparison to alternatives (Dropbox, Google Drive)

- Unlike Dropbox or Google Drive, Git/Github does not automatically track changes
 \leadsto you need to take (small) action