

Task and Project Management Primer

(this deck and links to materials are at benlockwood.com/training)

Benjamin Lockwood

Pre-doc orientation, July 2025

The philosophy of reproducibility: 3 parts

1. Reproducible *inputs*
2. Reproducible *analyses*
3. Reproducible *writeups*

The philosophy of reproducibility: 3 parts

1. Reproducible *inputs*

- `input` folders contain nothing but raw data and a `README` file
- “Raw data” means that no formatting, cleaning, etc. has been done
- `README` files can be brief, but should contain everything necessary for a future researcher to obtain all files in the folder

2. Reproducible *analyses*

3. Reproducible *writeups*

The philosophy of reproducibility: 3 parts

1. Reproducible *inputs*

2. Reproducible *analyses*

- Any analysis output should be reproducible at a future date, by another researcher.
- This includes final output in papers, but also any intermediate outputs (“let’s see what this looks like if we winsorize”)
- Litmus test: any time you are about to post or send some output: “if I were abducted by aliens immediately after sending this, would another researcher be able to pick up this project, read the project files, and produce exactly this output?”

3. Reproducible *writeups*

The philosophy of reproducibility: 3 parts

1. Reproducible *inputs*

2. Reproducible *analyses*

3. Reproducible *writeups*

- In a paper, every statistic and claim should be easily sourced.
- All calculated statistics should be imported from automated calculations, rather than typed by hand.
- Textual claims sometimes benefit from additional detail for easy sourcing, e.g., “Chetty (2013) finds a preferred value of 0.33”; in this case a LaTeX comment should be entered adjacent to the claim with further detail for the researcher: `% See first paragraph of conclusion, p. 1083`

References

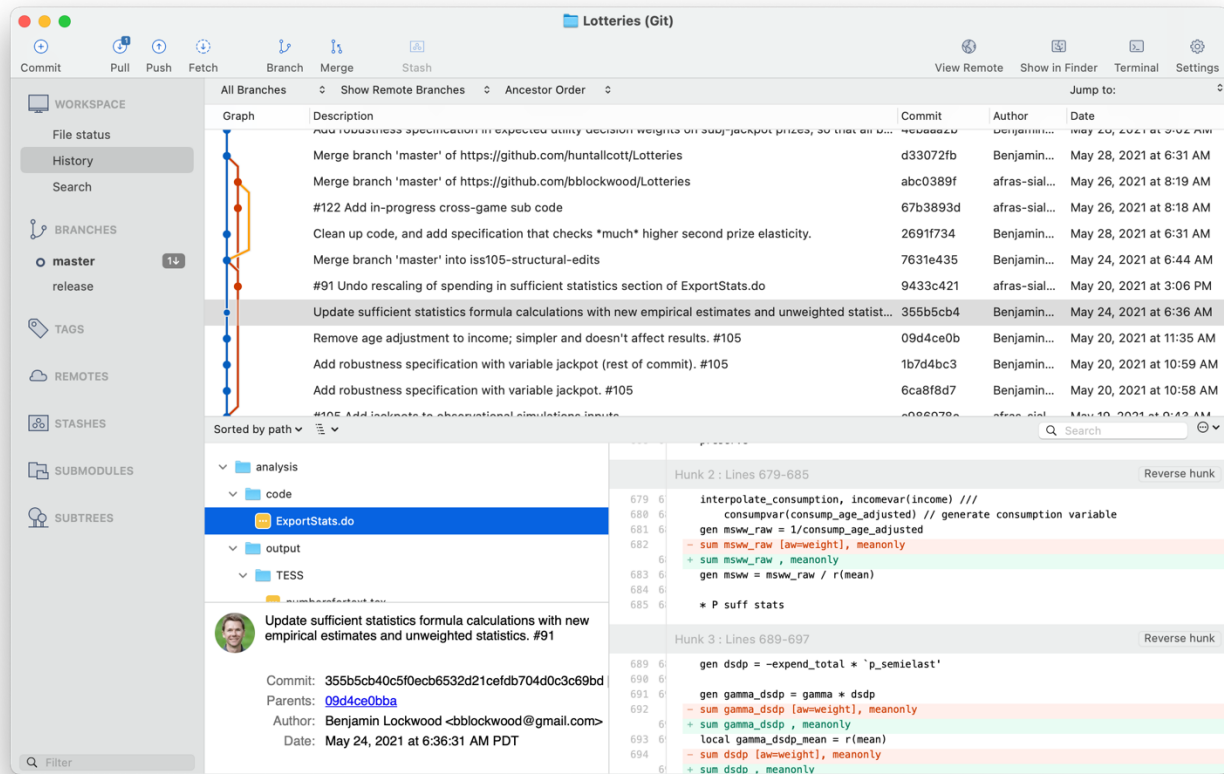
- Gentzkow and Shapiro have paved the way on many best practices.
- Starting point: their 2014 RA manual
 - See link at <https://github.com/gslab-econ/ra-manual/wiki/Introduction>
 - It is easy, quick, and enjoyable reading.
 - But references to specific programs and packages are dated. (GitHub did not exist in 2014!)
- They keep their GitHub wiki (above) updated with specifics.
- We will follow many of their best practices, though sometimes simplified to reflect our smaller teams.
- See <https://github.com/bblockwood/lab/wiki> for a guide to in-house best practices.

Project/task management with GitHub

Overview

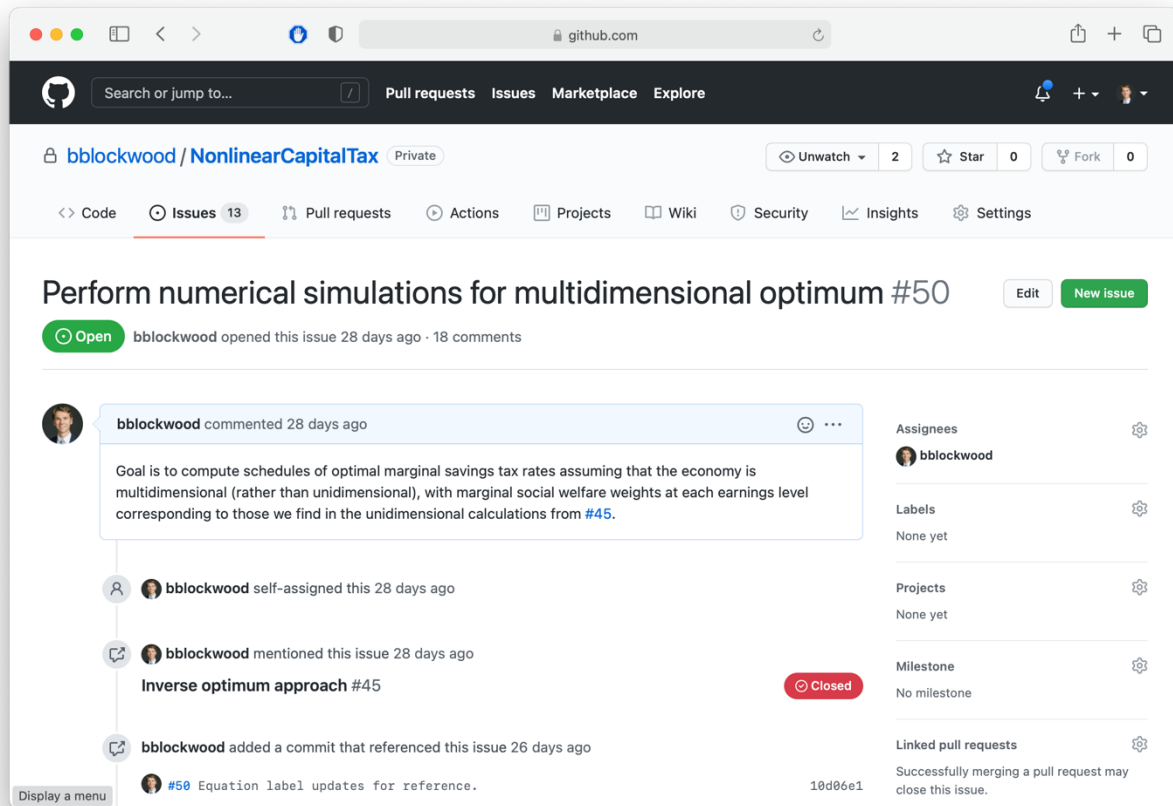
- Git: a version control software package
 - Keeps a snapshot of what a project folder (AKA “repository” or “repo”) looked like at any time.
 - Baseline repo structure: <https://github.com/bblockwood/lab/wiki/Project-Organization>
 - Easily revert entire repo to any desired date. (Note how this facilitates our reproducibility goals.)
 - Facilitates jointly working on a project while managing potential conflicts.
 - Recommended reading: the *Git Pro* ebook (<https://git-scm.com/book/en/v2>), ch. 1–3, 5
 - This will help you understand key concepts like *commit*, *push*, *branch*, *merge*, etc.
 - Self-tutoring with ChatGPT works well too—I use that a lot.
- GitHub: a website that integrates with Git to provide project management tools.
 - Cloud-hosted repositories that can be shared by collaborating researchers.
 - You can view (and link to!) any file in your repo on github.com.
 - “Issues”: a feature that provides task management and tracking alongside each repo.

Intro to Git



- Everything *can* be done from the command line.
- I instead recommend using a graphical management tool. I use *SourceTree*.
- Example: historical commits from one of my projects.
- See wiki: <https://github.com/bblockwood/lab/wiki/GitHub-b-Best-Practices>

Intro to GitHub



- Every issue starts with a “goal” (upon opening) and ends with a “summary” (upon closing).
- Any git commit tagged #50 auto-appears here.
- Issue-specific output gets saved to the folder `github/issues/issZZ`. This folder can be deleted when the issue is closed—it is always accessible via commit history!
- See <https://github.com/bblockwood/lab/wiki/Task-Management>