*IPR Methods Workshop*

# Using GitHub to Facilitate Reproducible Research

Jason Thomas
thomas.3912@osu.edu

Thursday, September 10th, 2020
12:30pm - 1:30pm

# Kick Off

- **WELCOME!!** to our 1st IPR Methods Workshop
  - Draw on the collective knowledge at IPR to share tips & best practices for conducting research.
  - several affiliates sharing/providing demos; tutorials; and ...
    - trade-off: productive vs. Jason inducing Zoom comas
  - Many thanks to John Casterline for help with planning & organizing (and presenting?)
- Topics
  - Reproducible research (GitHub, dynamic documents)
  - responsive to demand
  - please feel free to reach out with ideas

# Agenda

- Reproducible Research (Open Science)
  - motivation
  - pieces of the puzzle
- GitHub & GitHub Desktop
  - intersection with RR
  - basic concepts
  - a few examples

# Reproducible Research: why?

- General benefits for science
  - additional/efficient testing of hypotheses and models
  - difficult to reproduce results from an article's *Data & Methods* section
- Personal benefits
  - dealing with R&Rs; or returning to a project after a few days
  - collaborators, students/advisors
- Negative consequences?
  - more work. . . (maybe at first)
  - competition. . . (citations?, 2-way street)

# Reproducible Research: a few pieces of the puzzle

- Project Organization
  - folder structure: all data in the Data folder; all data preparation & EDA code in the Data-Prep folder; all the code for fitting models in the Models folder; all Figures in the Figures folder; etc.
  - scripts (and scripts for scripts $\longrightarrow$ automation)
  - documention / Read Me files (and markdown)
    - seriously? YES! (think about future you in December)
- Accessibility of files (code, data?, figures, tables, manuscript)
- Track changes (for code too!)
  - models.do; new_models.do; new_models_2.do; new_models_2b.do **YIKES!**

# Enter GitHub

- **GitHub** is an on-line service for making your project files available
- Folders and files can be organized just like on your local computer
  - supports markdown for Read Me files / documentation
- Global accessibility or restricted to in-house
  - how much space?
  - unlimited repositories (public and private); "abundant storage"; try to keep it under 1GB per repo
- Version control
  - developing code (differences between versions)
  - branches (e.g., sensitivity analyses; exploratory work)

# GitHub: additional features

- websites
  - directly: openva.net
  - through service (e.g., Netlify): Peter Choi
- markdown for tutorials
  - (also tab for wiki)
- hooks into coding software (R Studio, Sublime, Atom)
- large user base (easy to find answers/support on the web)
  - a lot of software/packages made available on GitHub (submitting issues/finding support)

# **GitHub**: signing up

Sign up for a free GitHub account:

- https://github.com/join
  - you will need to create a new username and password
  - the *Free* subscription will work just fine
    - there is also an educational account that provides more resources
- Verify your email address
  - click the little picture in the top-right corner (this is your profile picture, which you can change)
  - select the `Settings` option
  - select `Emails` in the panel on the left-hand side of the page
  - click on the button labeled `Send verification email`
  - open your email, view the message from GitHub, and click on the link to verify your email address
  - this will enable most of the useful feature of your GitHub account

# GitHub: concepts and actions

- Repository - a project where code (and file/folder structure) are stored
- Branch - a specific version of the project
- Basic Logic: TSA runs Dropbox
  - 2 copies of project files: local & GitHub
  - update files locally, then upload them to repository; GitHub tracks all the changes (and stores accessible copies of old versions)
  - you can also update files directly on GitHub
  - handles multiple users updating files (within reason)

# GitHub: concepts and actions (cont.)

Actions & Workflow...

- Clone - copy a repository to your computer for the first time
- Pull - update your local copy of the repository
- Stage (add/remove) - select files that you want to change in the a repository (on GitHub)
- Commit - finalize your changes
  - typically include a message describing the commit
- Push - send your changes to GitHub

# Git

- **GitHub Desktop** is software for interacting with GitHub repositories (pushing, pulling, etc.)
  - actually, it is really more of a friendly interface
- There is a second program **Git** which does most of the work (figuring out differences between files, tracking changes, transfering files, etc.)
  - GitHub Desktop uses Git behind the curtain
  - Both pieces of software are free (and open source)
- Download and install Git from
  - https://git-scm.com/downloads

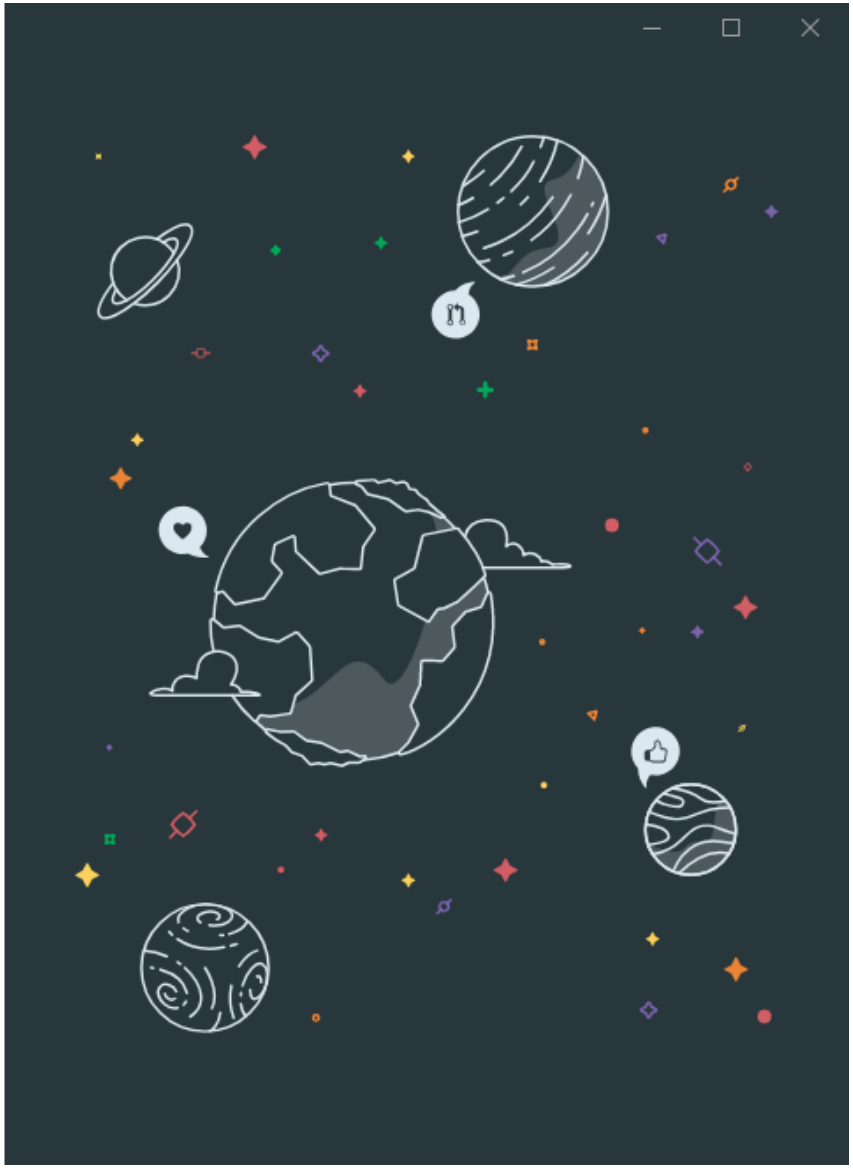
# GitHub Desktop

- Download and install the GitHub Desktop application
- https://desktop.github.com/
  - for Windows & Mac OS
- GitHub Desktop will present a Welcome Screen
  - sign in using the username and password you created previously

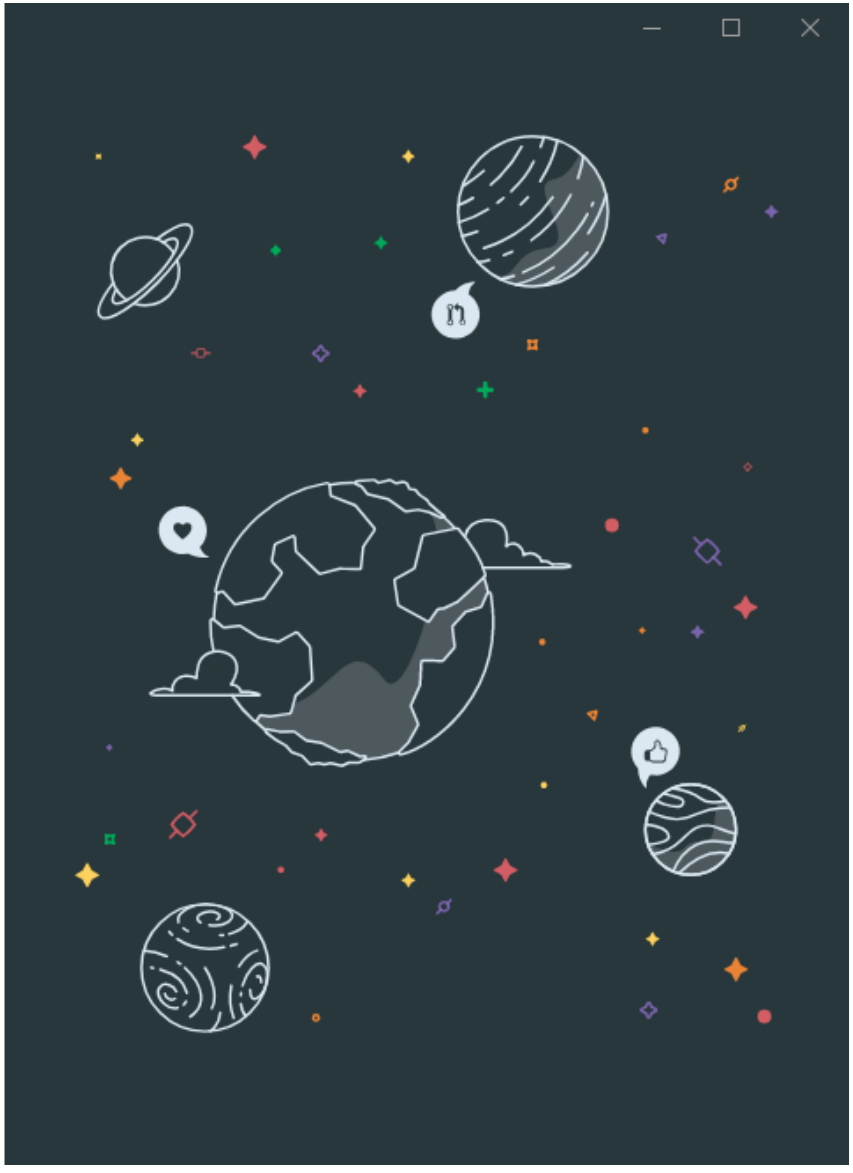
# Example Screenshots

GitHub Desktop Screenshots

# GitHub Desktop: welcome



# GitHub Desktop: sign in



# GitHub Desktop: configure

## Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

Name

Jason Thomas

Email

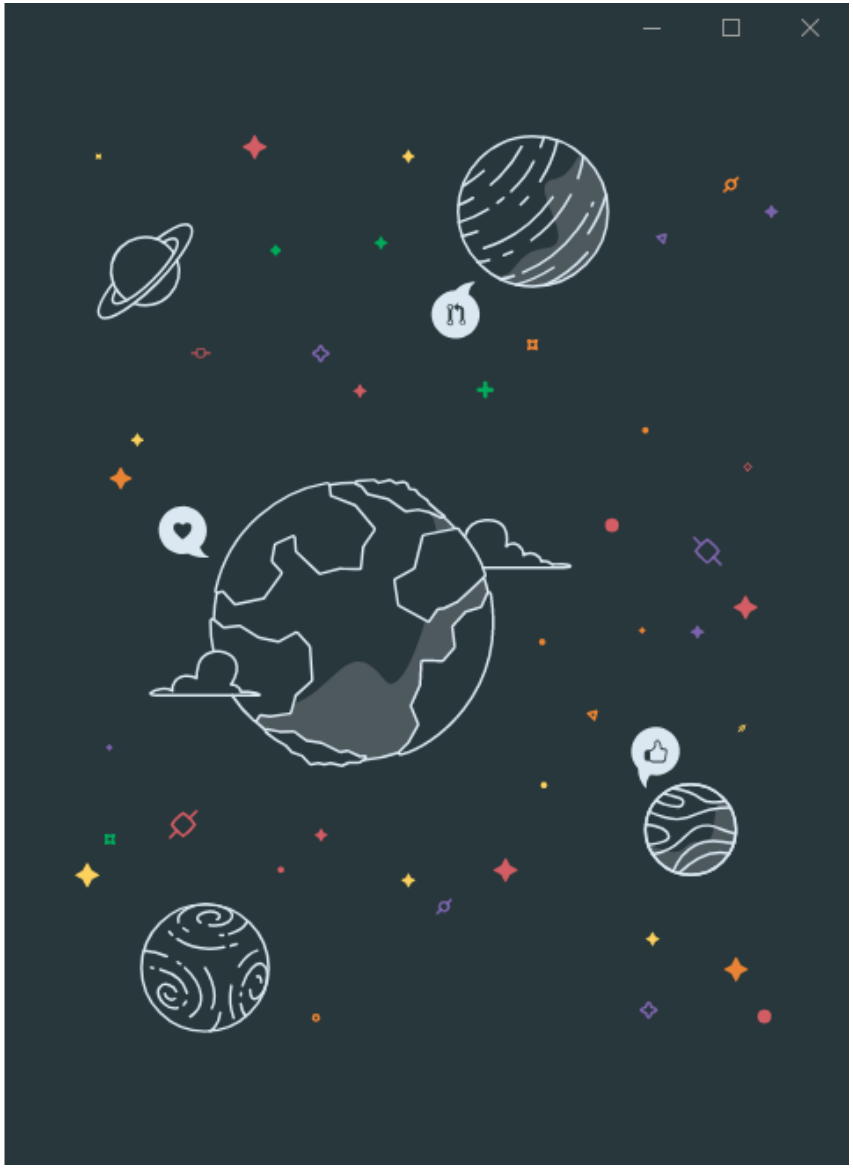
jarathomas@gmail.com

Continue Cancel

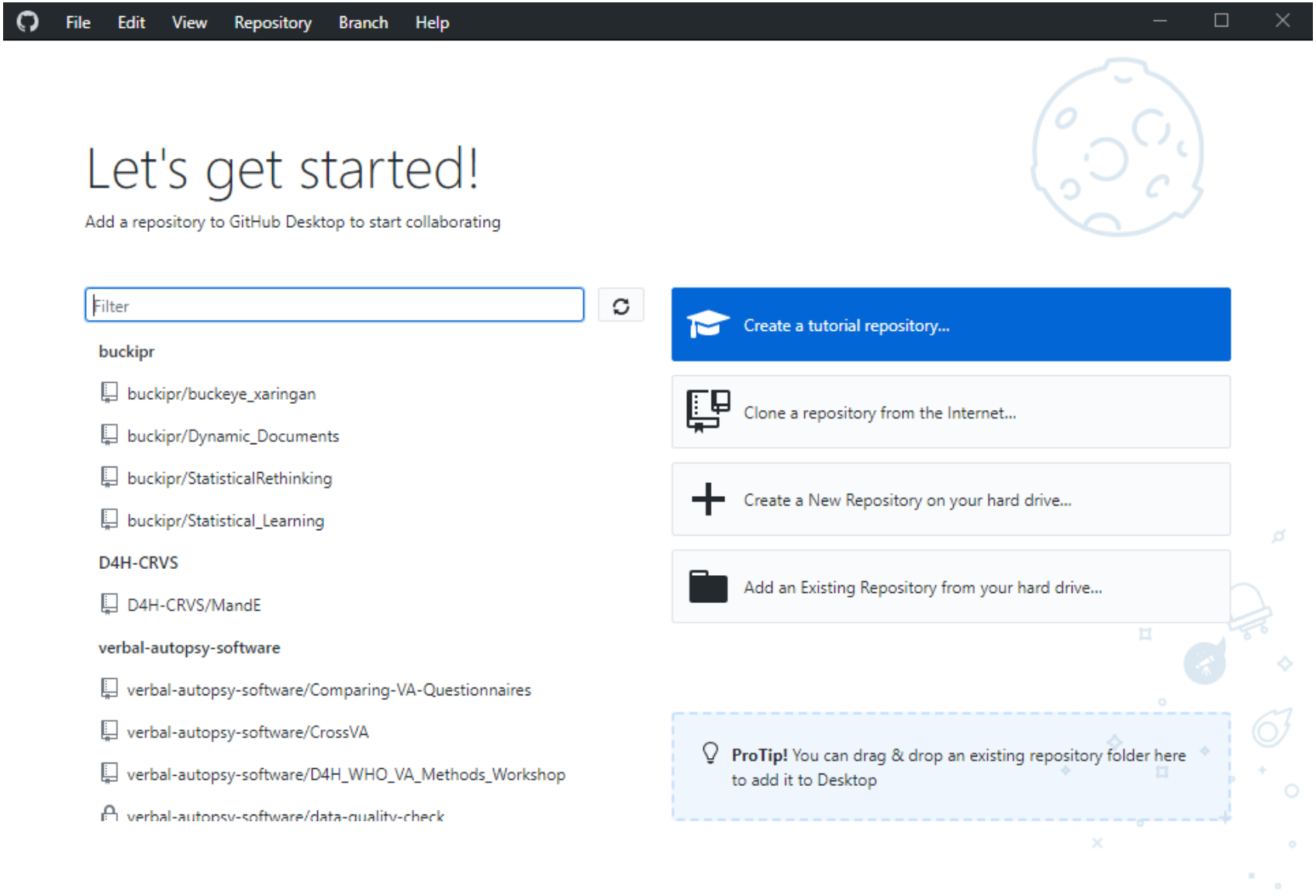
Example commit

**Fix all the things**

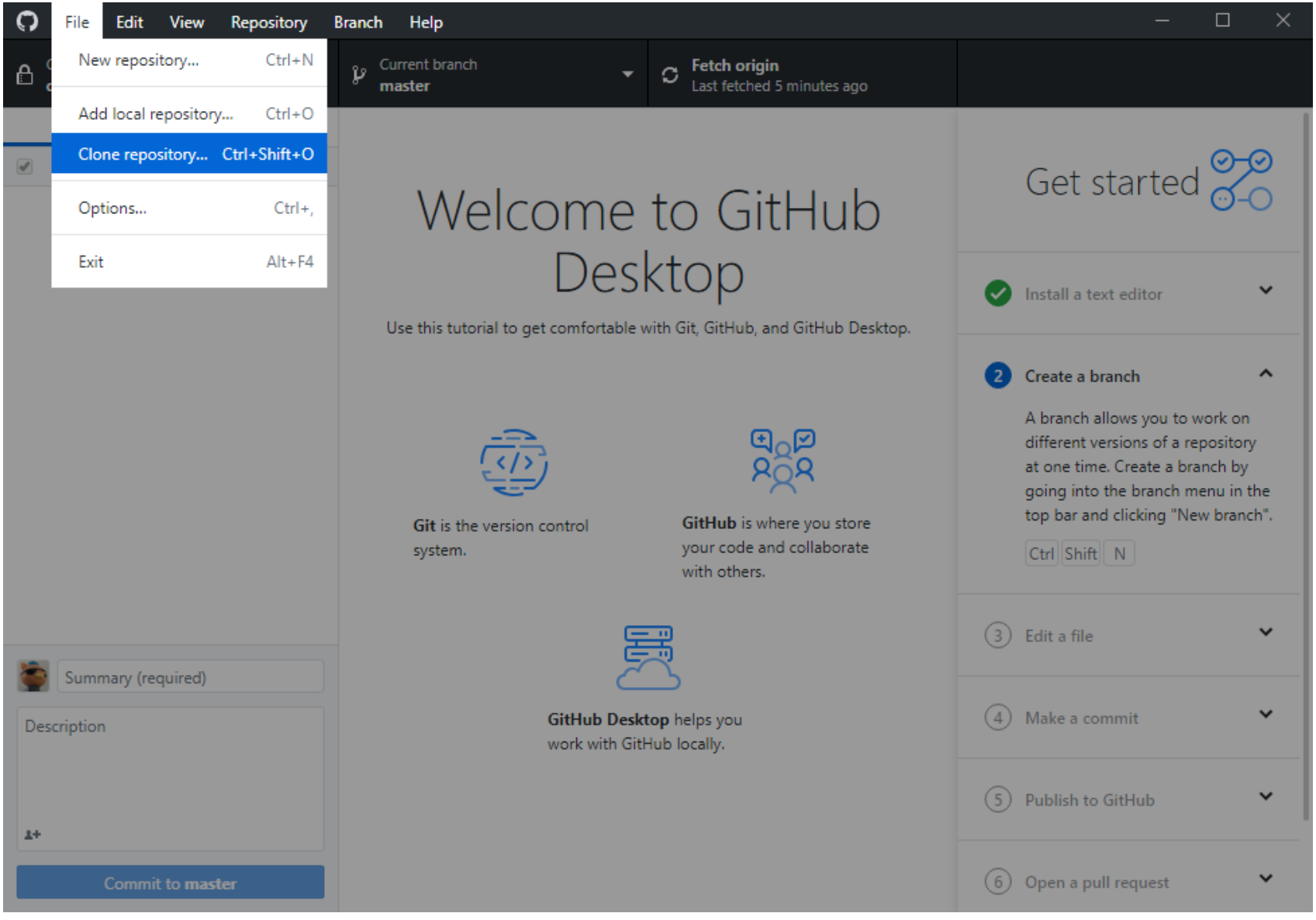
Jason Thomas committed 30 minutes ago



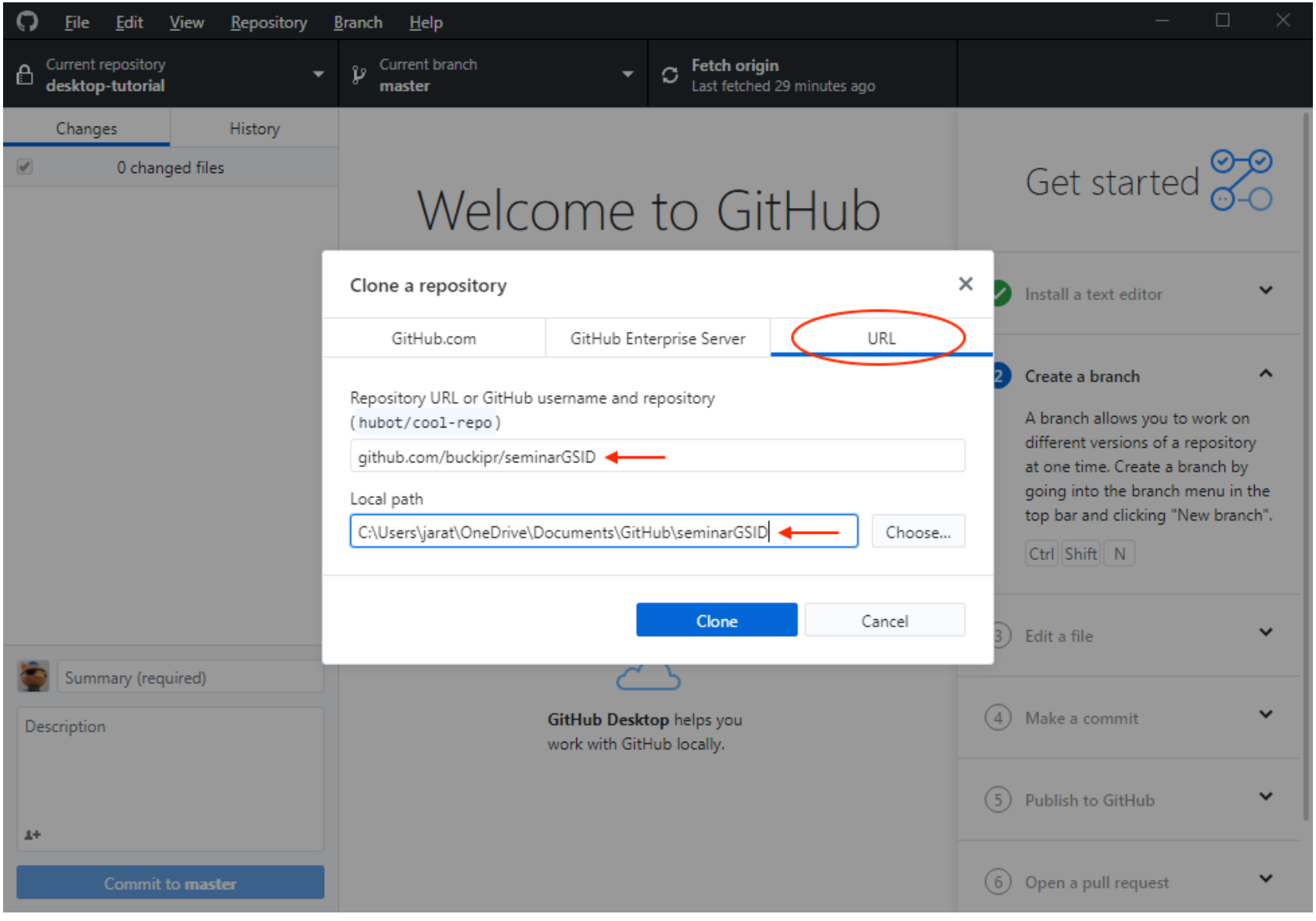
# GitHub Desktop: get started



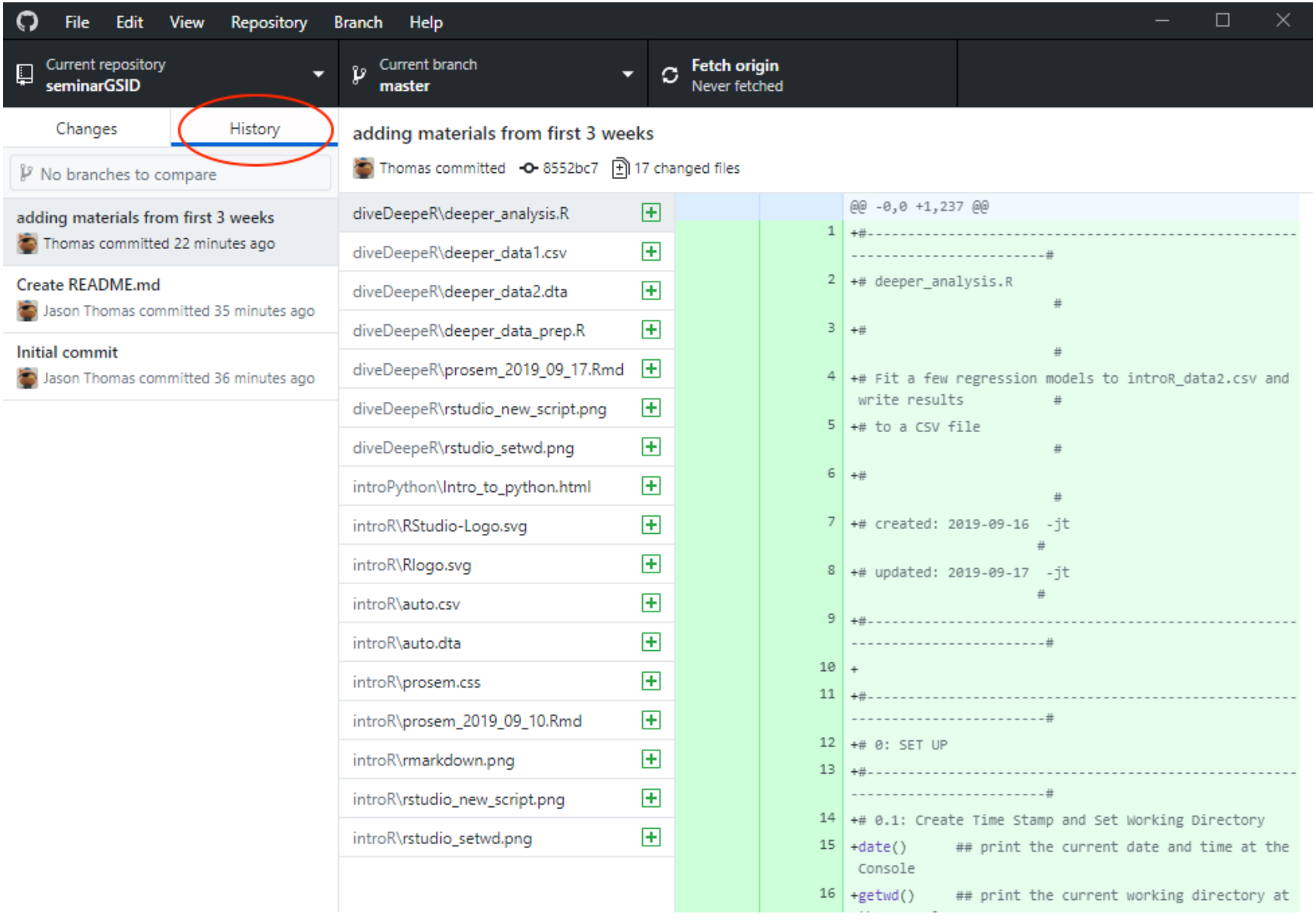
# GitHub Desktop: clone



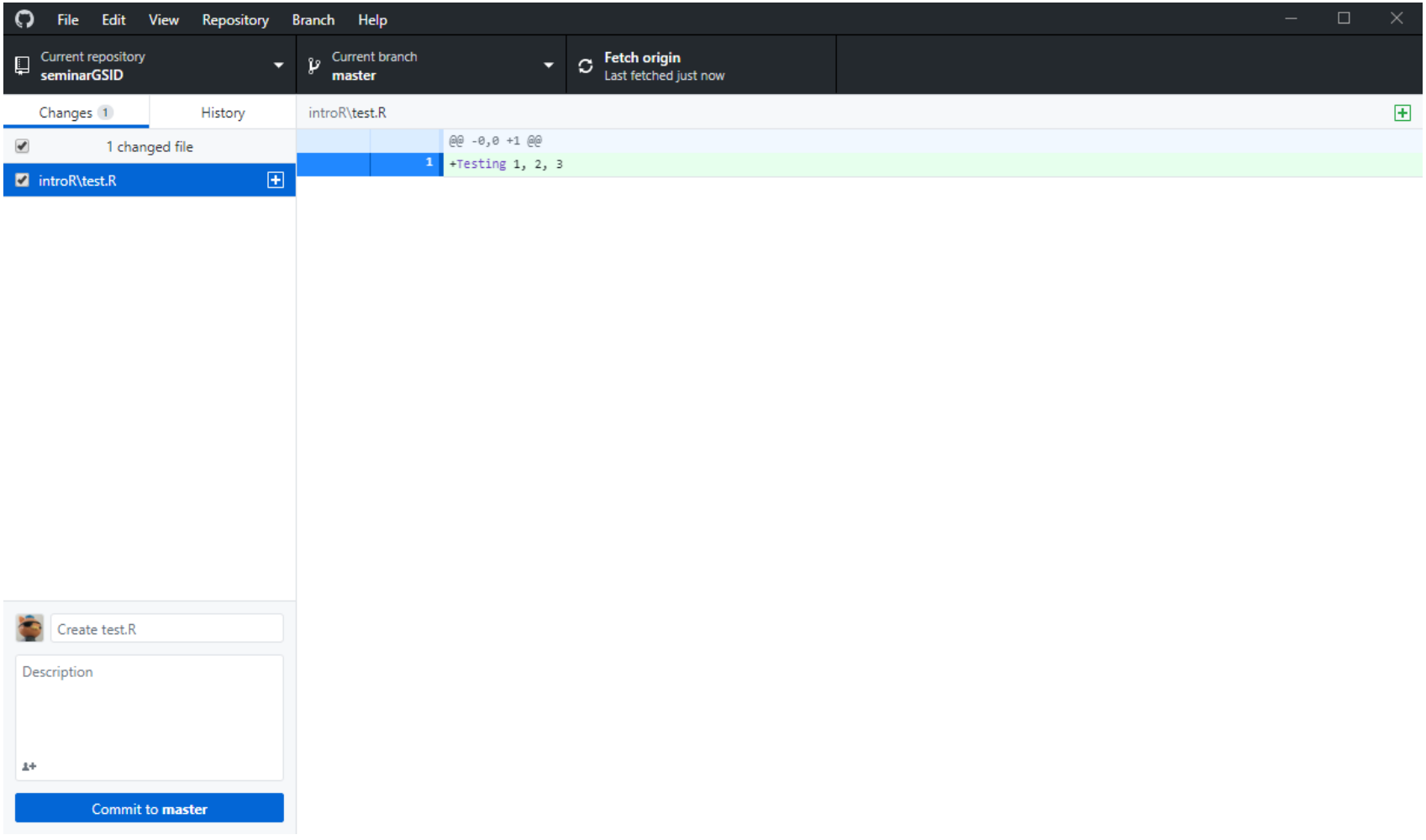
# GitHub Desktop: select repository



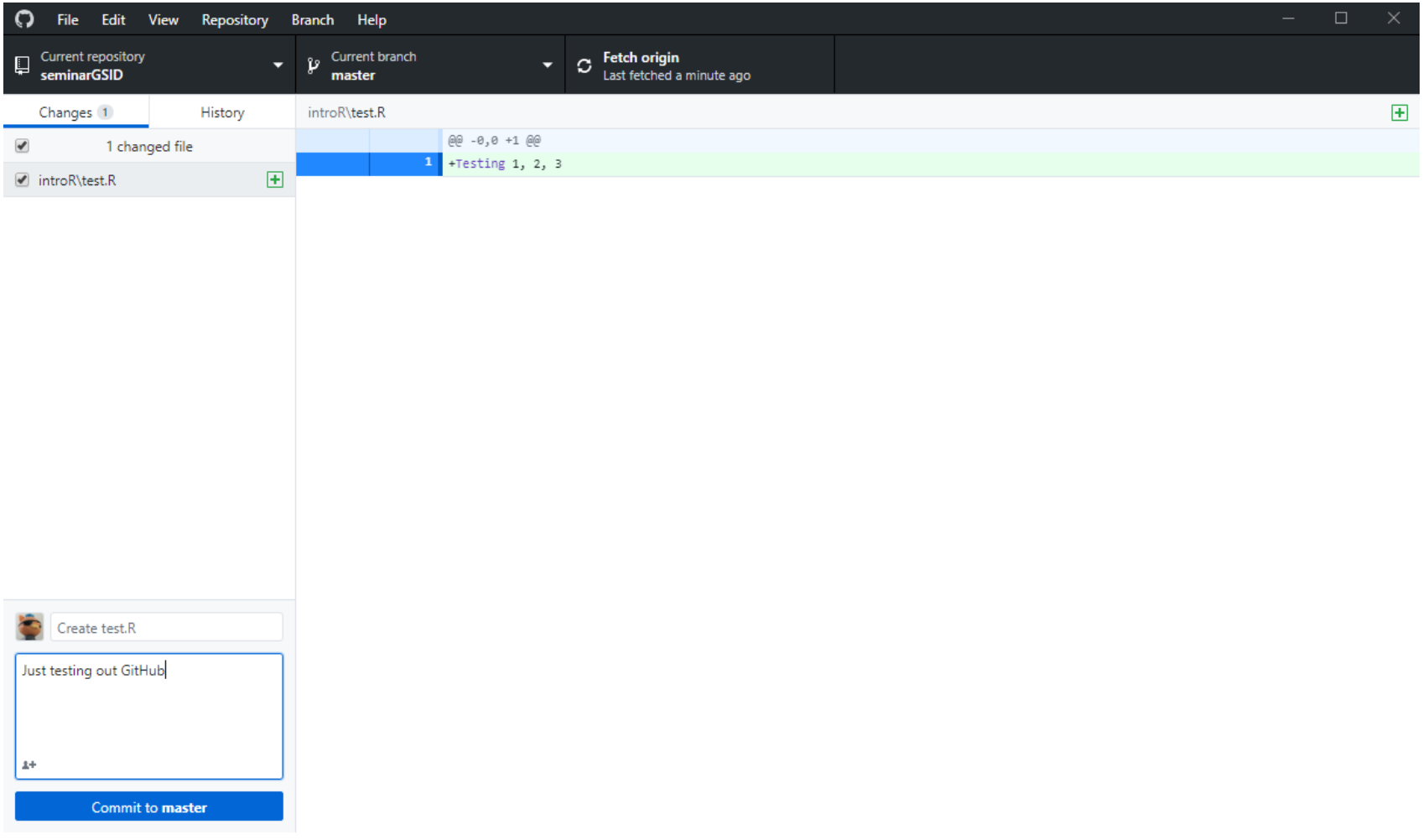
# GitHub Desktop: history



# GitHub Desktop: stage



# GitHub Desktop: commit



# GitHub Desktop: push