

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 → automation)
 - ▶ documentation / 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 features 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, transferring 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

Welcome to GitHub Desktop

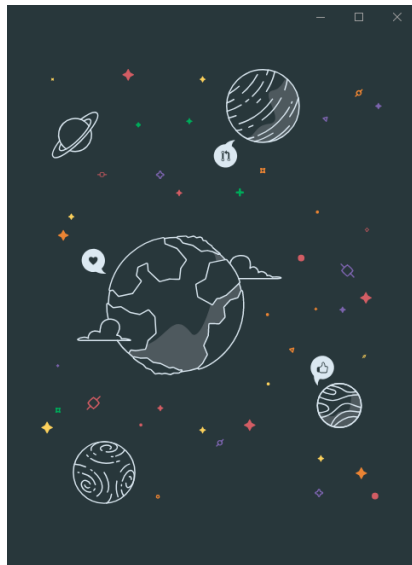
GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise Server. Sign in below to get started with your existing projects.

New to GitHub? [Create your free account.](#)

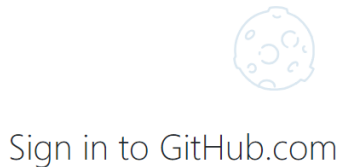
[Sign in to GitHub.com](#)

[Sign in to GitHub Enterprise Server](#)

[Skip this step](#)



GitHub Desktop: sign in



Username or email address

jarathomas@gmail.com

Password

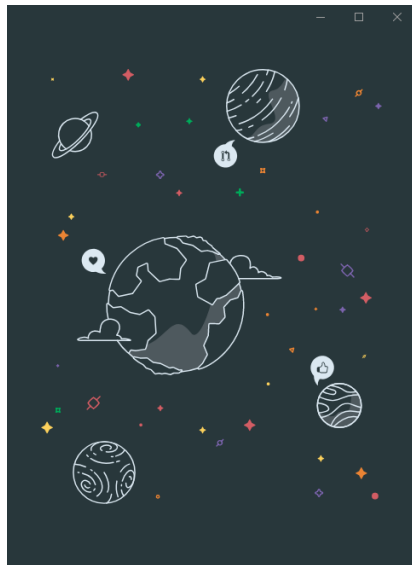
.....

Sign in

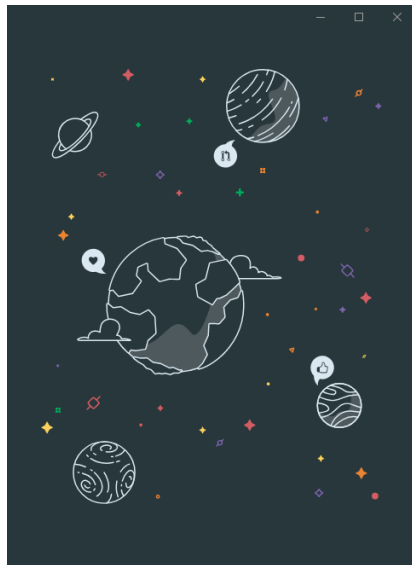
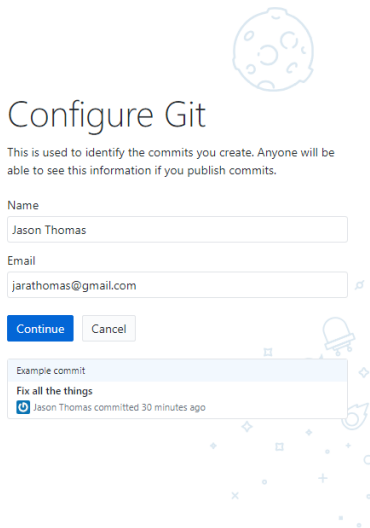
Cancel

[Forgot password?](#)

[Sign in using your browser](#)



GitHub Desktop: configure



GitHub Desktop: get started





Let's get started!


Add a repository to GitHub Desktop to start collaborating




buckipr


 buckipr/buckeye_xaringan

 buckipr/Dynamic_Documents


 buckipr/StatisticalRethinking


 buckipr/Statistical_Learning


D4H-CRVS


 D4H-CRVS/MandE

verbal-autopsy-software

 verbal-autopsy-software/Comparing-VA-Questionnaires

 verbal-autopsy-software/CrossVA

 verbal-autopsy-software/D4H_WHO_VA_Methods_Workshop

 verbal-autopsy-software/data-quality-check



Create a tutorial repository...



Clone a repository from the Internet...



Create a New Repository on your hard drive...

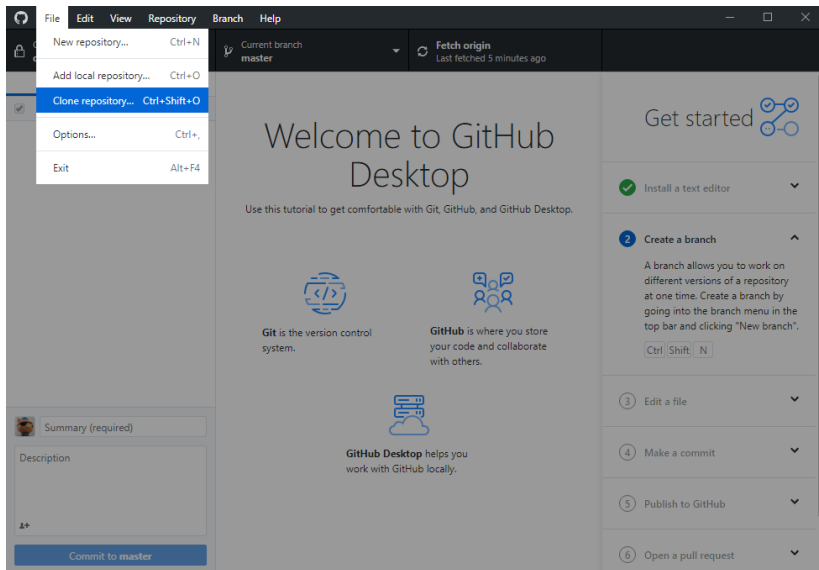


Add an Existing Repository from your hard drive...

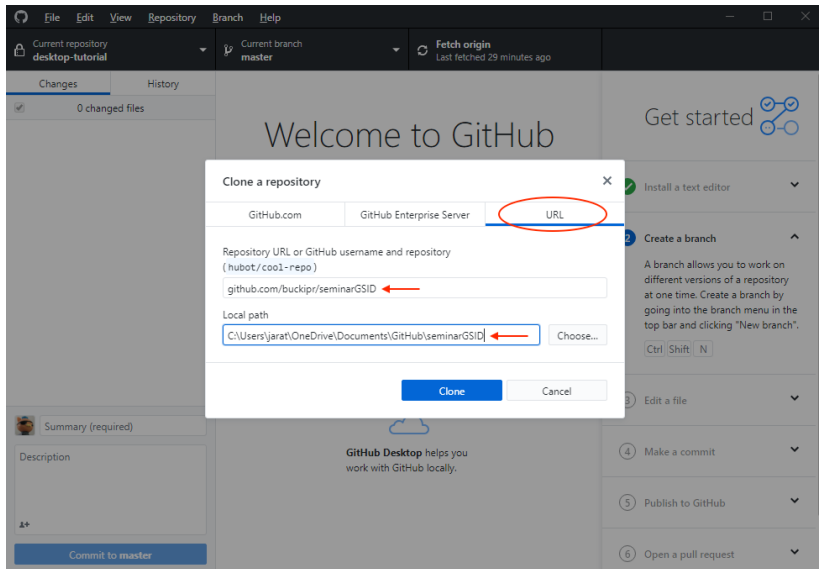


ProTip! You can drag & drop an existing repository folder here to add it to Desktop

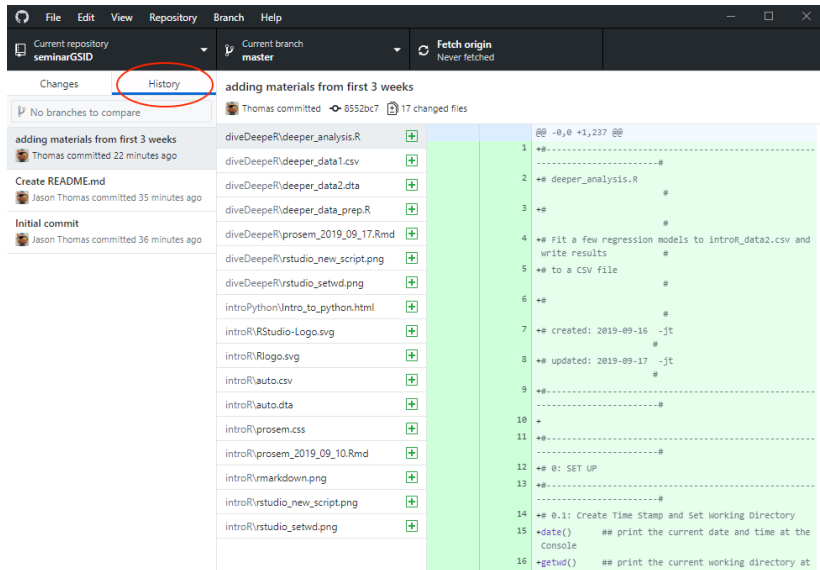
GitHub Desktop: clone



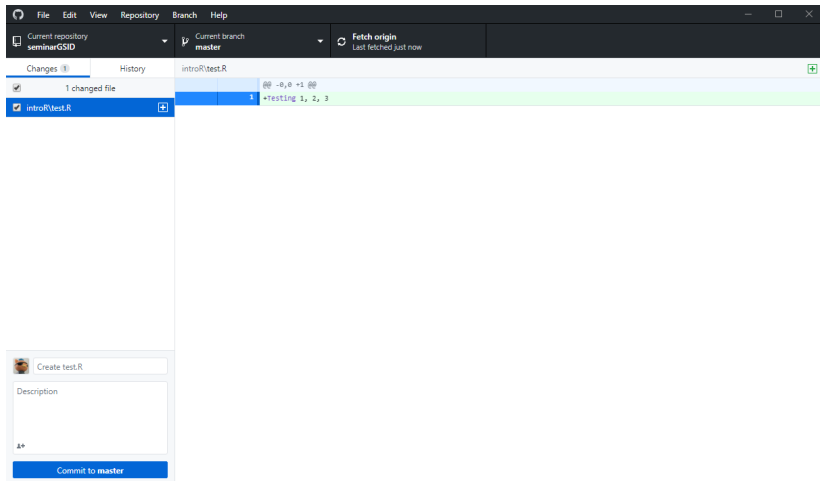
GitHub Desktop: select repository



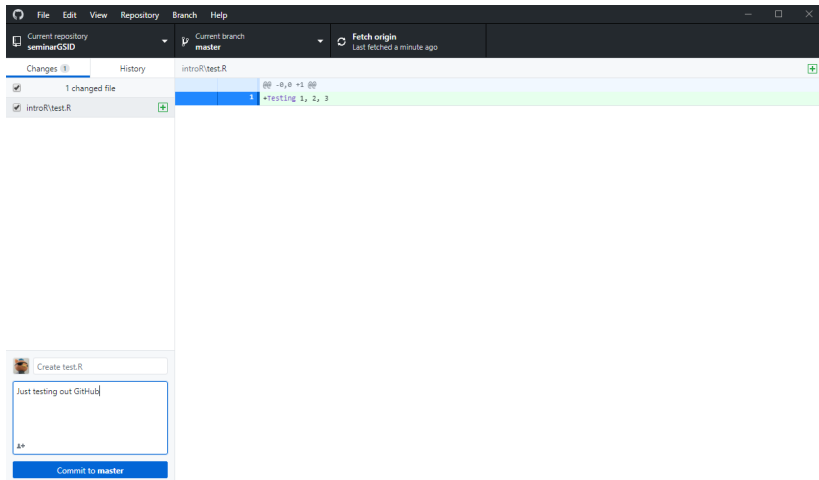
GitHub Desktop: history



GitHub Desktop: stage



GitHub Desktop: commit



GitHub Desktop: push

The screenshot shows the GitHub Desktop application window. The top menu bar includes File, Edit, View, Repository, Branch, and Help. Below the menu bar, the 'Current repository' is set to 'seminarGSID' and the 'Current branch' is 'master'. The 'Push origin' button is visible in the top right of the toolbar, indicating the last fetch was 2 minutes ago.

The main area displays the message 'No local changes' with a subtext: 'There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.' To the right of this text is a small illustration of a building with a flag.

On the left sidebar, the 'Changes' tab is selected, showing '0 changed files'. Below this, there is a section for committing changes, including a 'Summary (required)' field, a 'Description' field, and a 'Commit to master' button. At the bottom of the sidebar, it shows 'Committed just now' and 'Create test.R' with an 'Undo' button.

The main area contains three suggestions:

- Push 1 commit to the origin remote**: You have one local commit waiting to be pushed to GitHub. Always available in the toolbar when there are local commits waiting to be pushed or `Ctrl + P`. A **Push origin** button is present.
- Open the repository in your external editor**: Select your editor in [Options](#). Repository menu or `Ctrl + Shift + A`. An **Open in Atom** button is present.
- View the files of your repository in Explorer**: Repository menu or `Ctrl + Shift + F`. A **Show in Explorer** button is present.
- Open the repository page on GitHub in your browser**: Repository menu or `Ctrl + Shift + G`. A **View on GitHub** button is present.