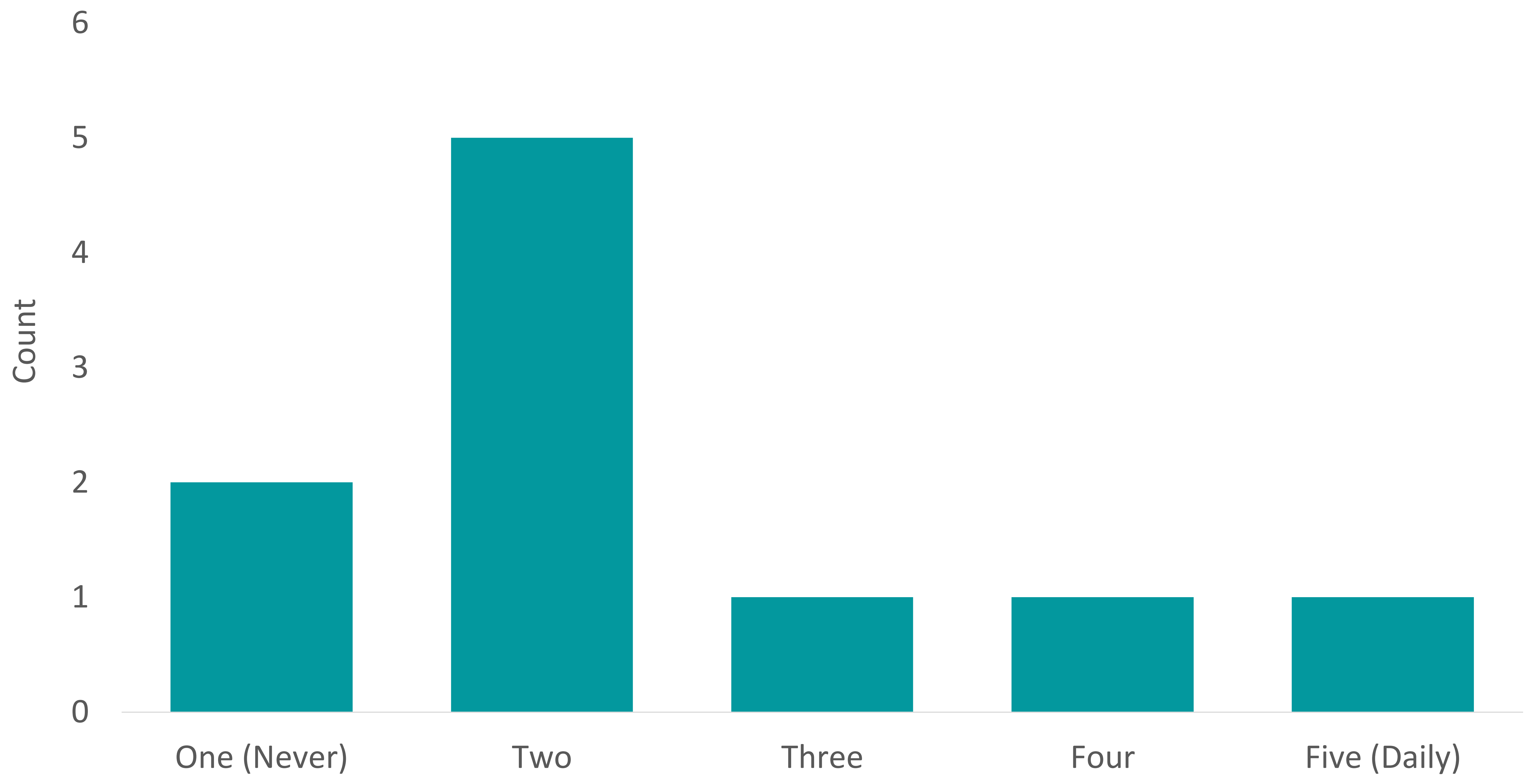
Project-based workflows with GitHub

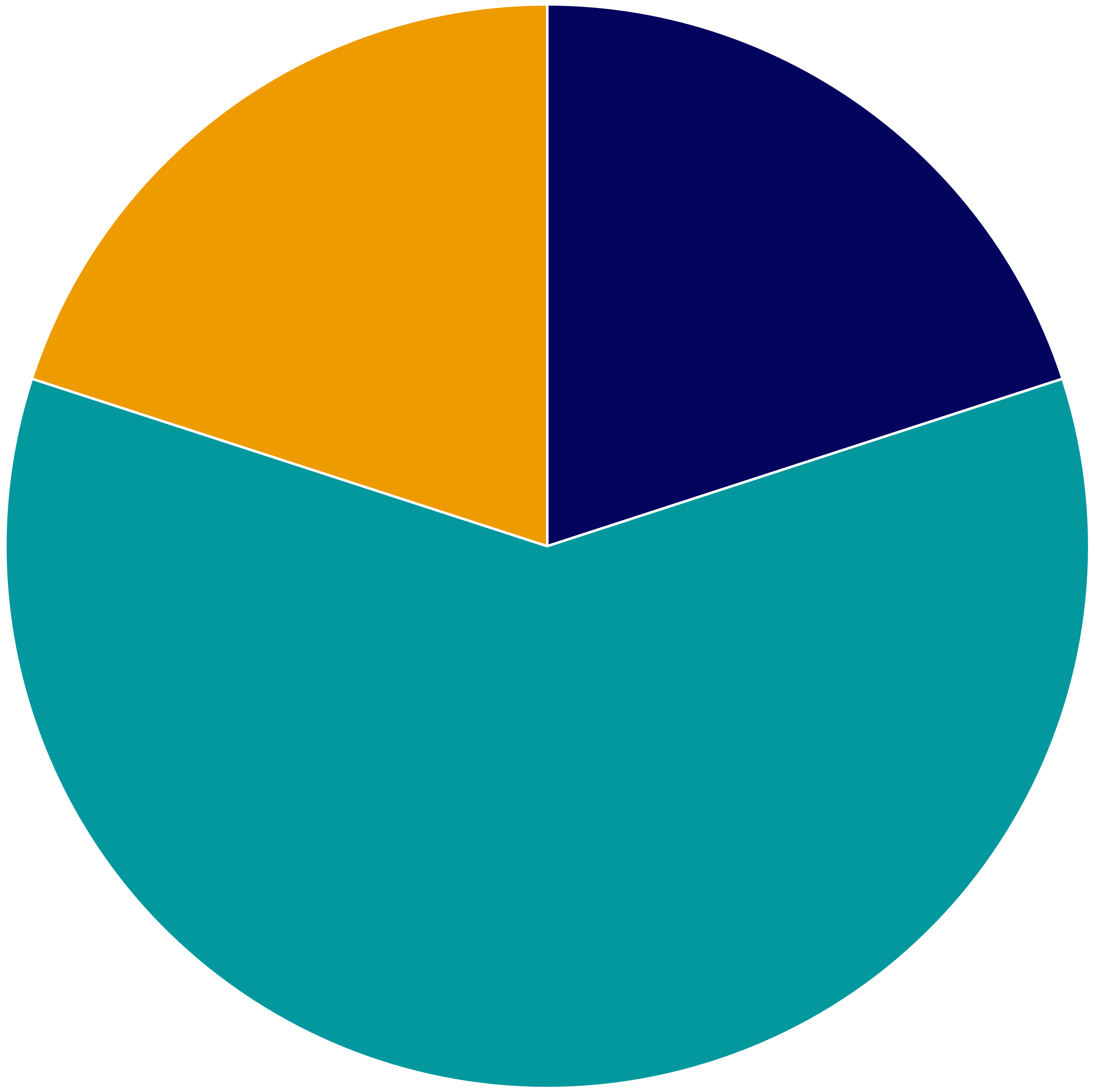
Courtney Robichaud & Emma Hudgins

Goal

You walk away confident using Git/GitHub for version control with your (R-based) projects

What is your level of familiarity with GitHub?



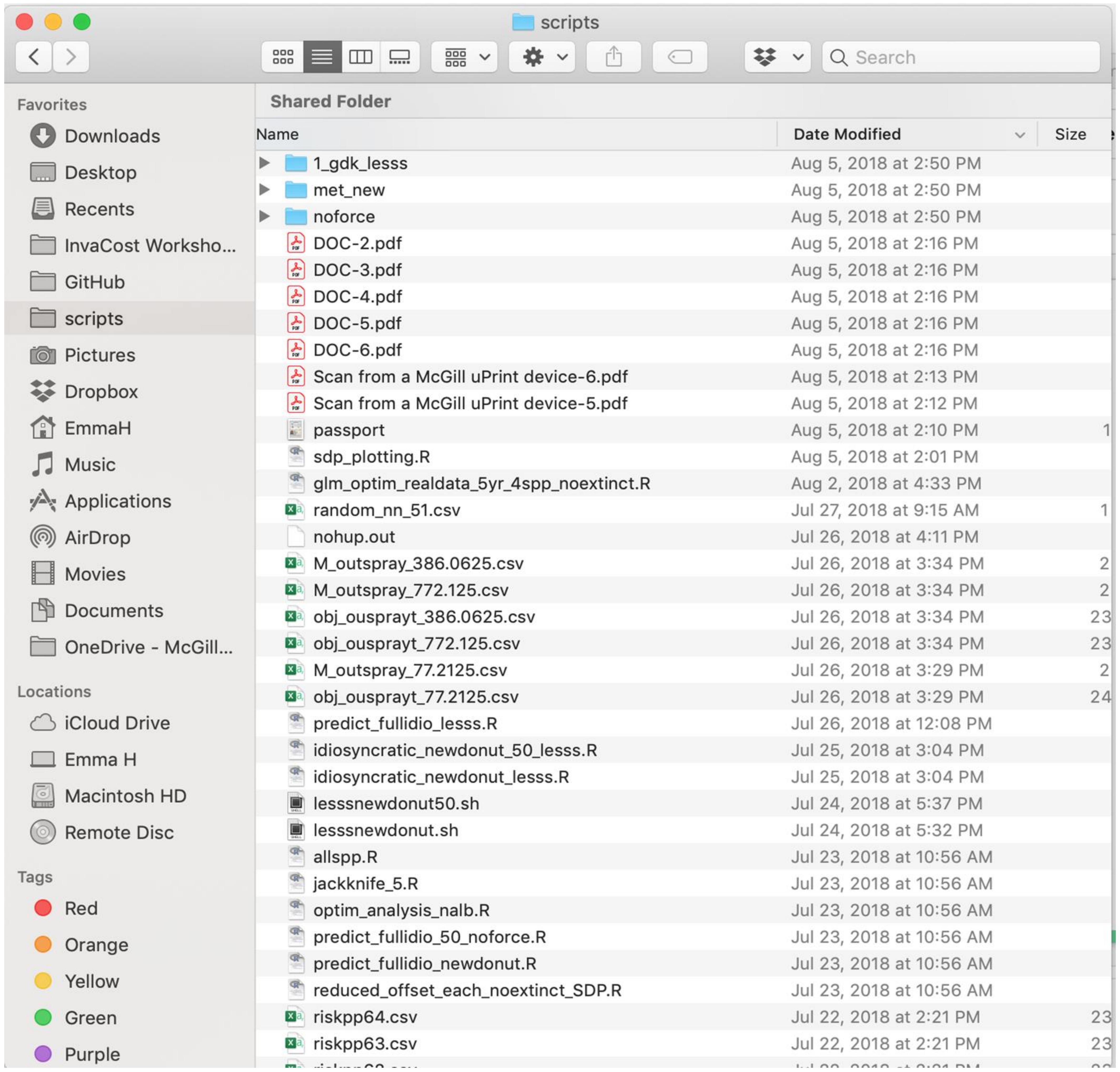


■ Organizing projects for lab members in one location

■ Creating an organized system for reproducible analyses

■ All of the above

Your current organization



scripts

Search

Favorites
- Downloads
- Desktop
- Recents
- InvaCost Worksho...
- GitHub
- scripts
- Pictures
- Dropbox
- EmmaH
- Music
- Applications
- AirDrop
- Movies
- Documents
- OneDrive - McGill...

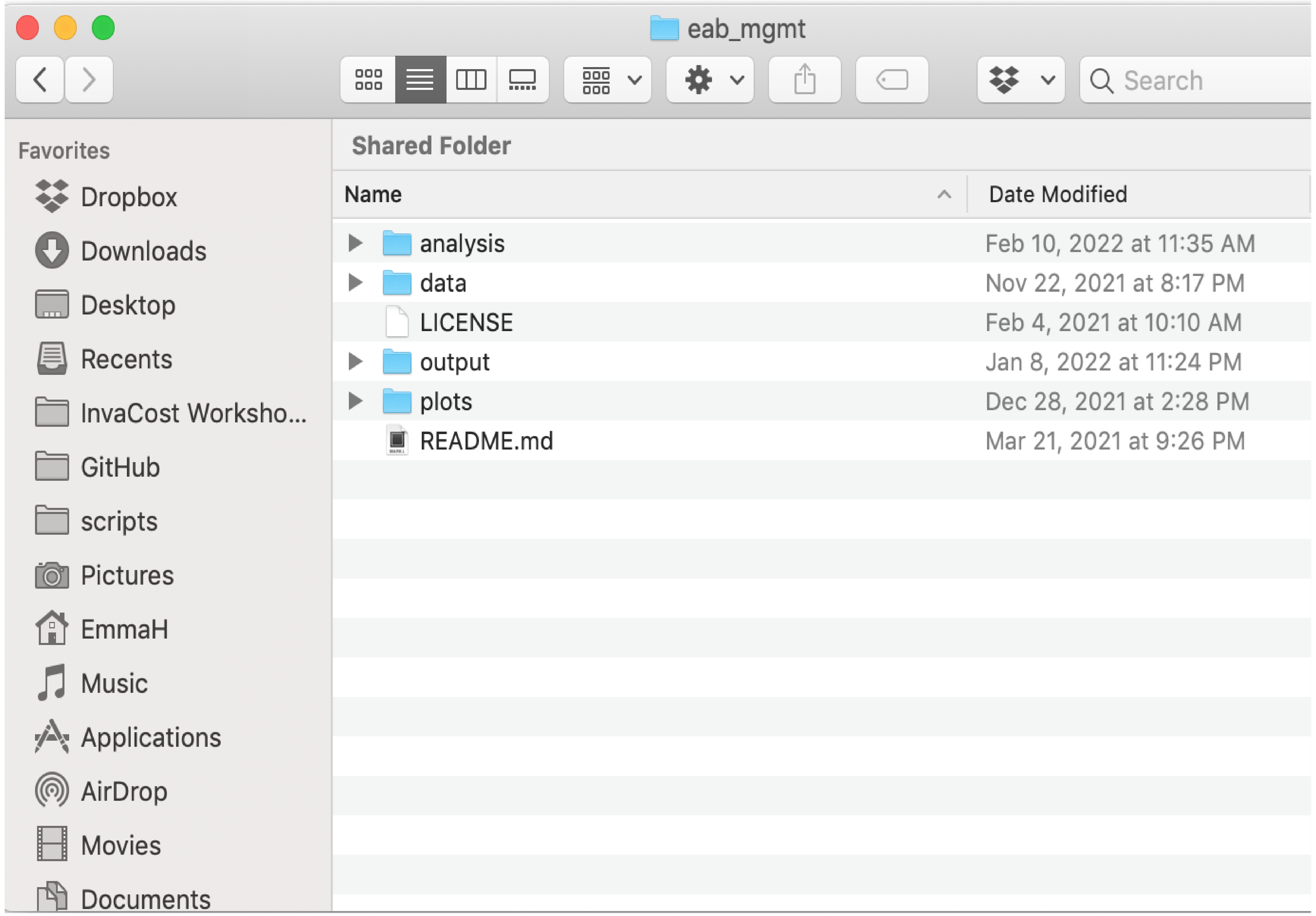
Locations
- iCloud Drive
- Emma H
- Macintosh HD
- Remote Disc

Tags
- 🔴 Red
- 🟠 Orange
- 🟡 Yellow
- 🟢 Green
- 🟣 Purple

Shared Folder

Name	Date Modified	Size
▶ 1_gdk_lesss	Aug 5, 2018 at 2:50 PM	
▶ met_new	Aug 5, 2018 at 2:50 PM	
▶ noforce	Aug 5, 2018 at 2:50 PM	
DOC-2.pdf	Aug 5, 2018 at 2:16 PM	
DOC-3.pdf	Aug 5, 2018 at 2:16 PM	
DOC-4.pdf	Aug 5, 2018 at 2:16 PM	
DOC-5.pdf	Aug 5, 2018 at 2:16 PM	
DOC-6.pdf	Aug 5, 2018 at 2:16 PM	
Scan from a McGill uPrint device-6.pdf	Aug 5, 2018 at 2:13 PM	
Scan from a McGill uPrint device-5.pdf	Aug 5, 2018 at 2:12 PM	
passport	Aug 5, 2018 at 2:10 PM	1
sdp_plotting.R	Aug 5, 2018 at 2:01 PM	
glm_optim_realdata_5yr_4spp_noextinct.R	Aug 2, 2018 at 4:33 PM	
random_nn_51.csv	Jul 27, 2018 at 9:15 AM	1
nohup.out	Jul 26, 2018 at 4:11 PM	
M_outspray_386.0625.csv	Jul 26, 2018 at 3:34 PM	2
M_outspray_772.125.csv	Jul 26, 2018 at 3:34 PM	2
obj_ousprayt_386.0625.csv	Jul 26, 2018 at 3:34 PM	23
obj_ousprayt_772.125.csv	Jul 26, 2018 at 3:34 PM	23
M_outspray_77.2125.csv	Jul 26, 2018 at 3:29 PM	2
obj_ousprayt_77.2125.csv	Jul 26, 2018 at 3:29 PM	24
predict_fullidio_lesss.R	Jul 26, 2018 at 12:08 PM	
idiosyncratic_newdonut_50_lesss.R	Jul 25, 2018 at 3:04 PM	
idiosyncratic_newdonut_lesss.R	Jul 25, 2018 at 3:04 PM	
lesssnewdonut50.sh	Jul 24, 2018 at 5:37 PM	
lesssnewdonut.sh	Jul 24, 2018 at 5:32 PM	
allspp.R	Jul 23, 2018 at 10:56 AM	
jackknife_5.R	Jul 23, 2018 at 10:56 AM	
optim_analysis_nalb.R	Jul 23, 2018 at 10:56 AM	
predict_fullidio_50_noforce.R	Jul 23, 2018 at 10:56 AM	
predict_fullidio_newdonut.R	Jul 23, 2018 at 10:56 AM	
reduced_offset_each_noextinct_SDP.R	Jul 23, 2018 at 10:56 AM	
riskpp64.csv	Jul 22, 2018 at 2:21 PM	23
riskpp63.csv	Jul 22, 2018 at 2:21 PM	23

Your ideal organization


eab_mgmt

Favorites
- Dropbox
- Downloads
- Desktop
- Recents
- InvaCost Worksho...
- GitHub
- scripts
- Pictures
- EmmaH
- Music
- Applications
- AirDrop
- Movies
- Documents

Shared Folder

Name	Date Modified
analysis	Feb 10, 2022 at 11:35 AM
data	Nov 22, 2021 at 8:17 PM
LICENSE	Feb 4, 2021 at 10:10 AM
output	Jan 8, 2022 at 11:24 PM
plots	Dec 28, 2021 at 2:28 PM
README.md	Mar 21, 2021 at 9:26 PM

Why use R projects?

1. Keep all your associated files for each research project together in one place

2. Never setwd() again

3. Make it easy to work across machines, return to work after long absences, and share your work

Why use Git & GitHub?

1. Tracks changes made to a project
2. GitHub provides cloud storage/backup for your projects and files
3. Makes working across different machines easy (individual or with collaborators)
4. Can easily share your work (however you are comfortable)

Git vs GitHub

Git

- Installed locally on your computer
- Version control of code
- Minimal additional features

GitHub

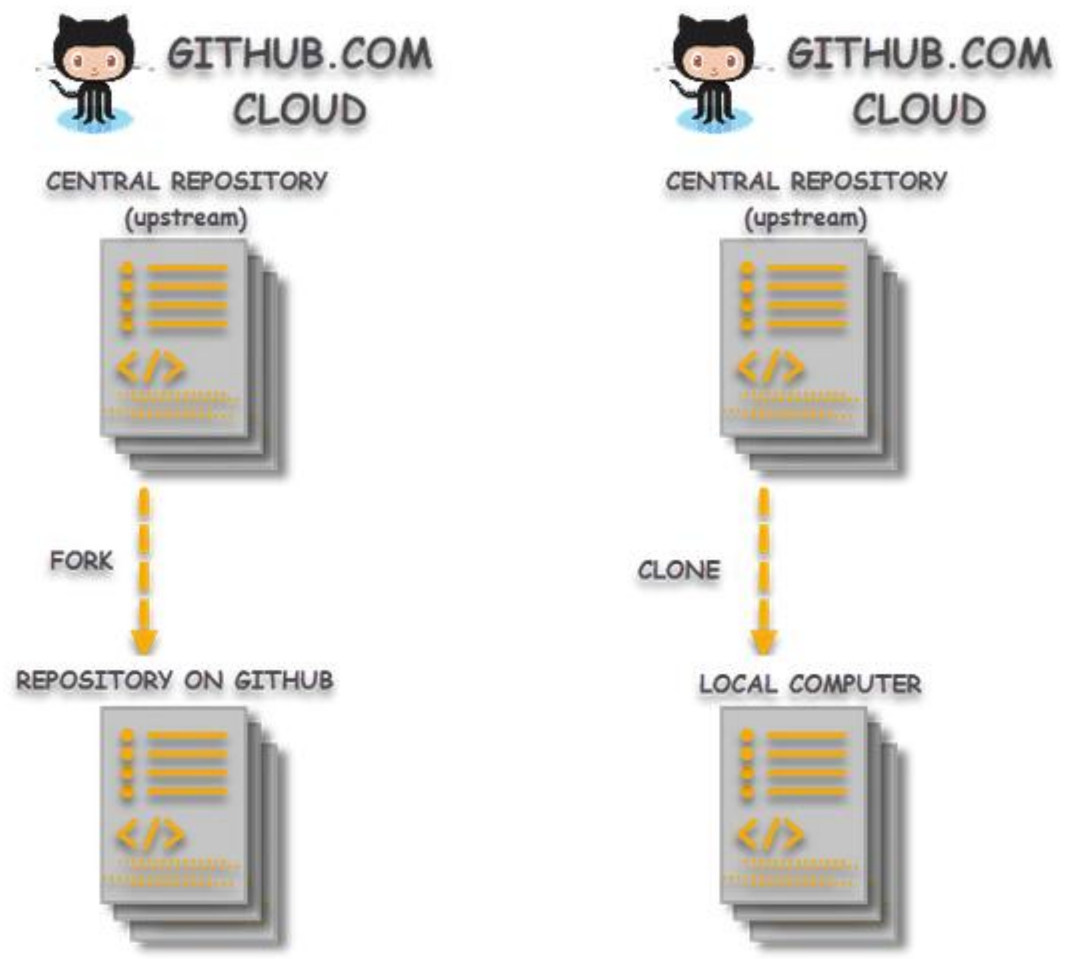
- Cloud-based source code repository
- Built around Git technology
- Services for collaboration, editing, tracking etc.

Quick glossary

Repository: contains all your project files and version history

Clone: Copy an existing repo onto your **local machine**

Fork: Freeze an existing repo and create an **independent copy** on your GitHub account



https://www.toolsqa.com/git/difference-between-git-clone-and-git-fork/

Preferred Sequence

"GitHub first, then RStudio"

1. We will set up the project in GitHub first by making a repository

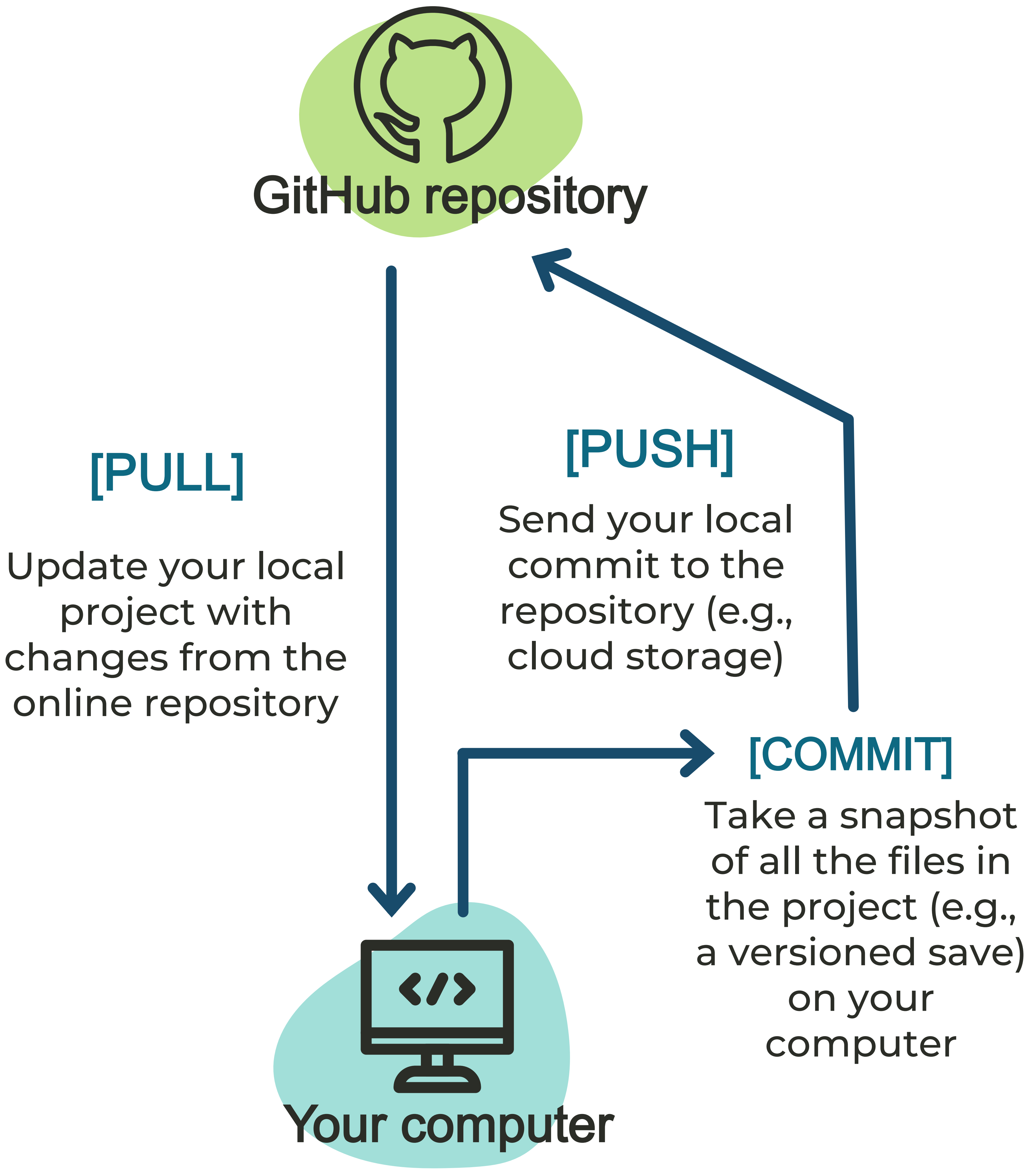
2. Then we will make a project in Rstudio, and clone the repository to our local computer

Your moves

Commit: Capture a ‘snapshot’ of the project (repository). Commits shape the project history.

Push: Upload local repository content (commits) to the remote repository (on GitHub)

Pull: Download content from the remote repository and update the repository on your computer


GitHub repository
[PULL]
Update your local project with changes from the online repository
[PUSH]
Send your local commit to the repository (e.g., cloud storage)
[COMMIT]
Take a snapshot of all the files in the project (e.g., a versioned save) on your computer
Your computer

Large Files

50MB: Git warns you that this is a big file

100MB: hard limit on pushes

GitHub Large File Storage: Alternative storage for files up to 2GB (free)

Releases: Can store large compressed files (.tar.gz)

Common Issues/Tricks

Push > Pull conflicts: If you are not up to date with the repo, your push will be rejected. Try pulling, then push your work

Don't push intermediate work – what is in your repo should always "work"

Fork vs Clone

Forking:

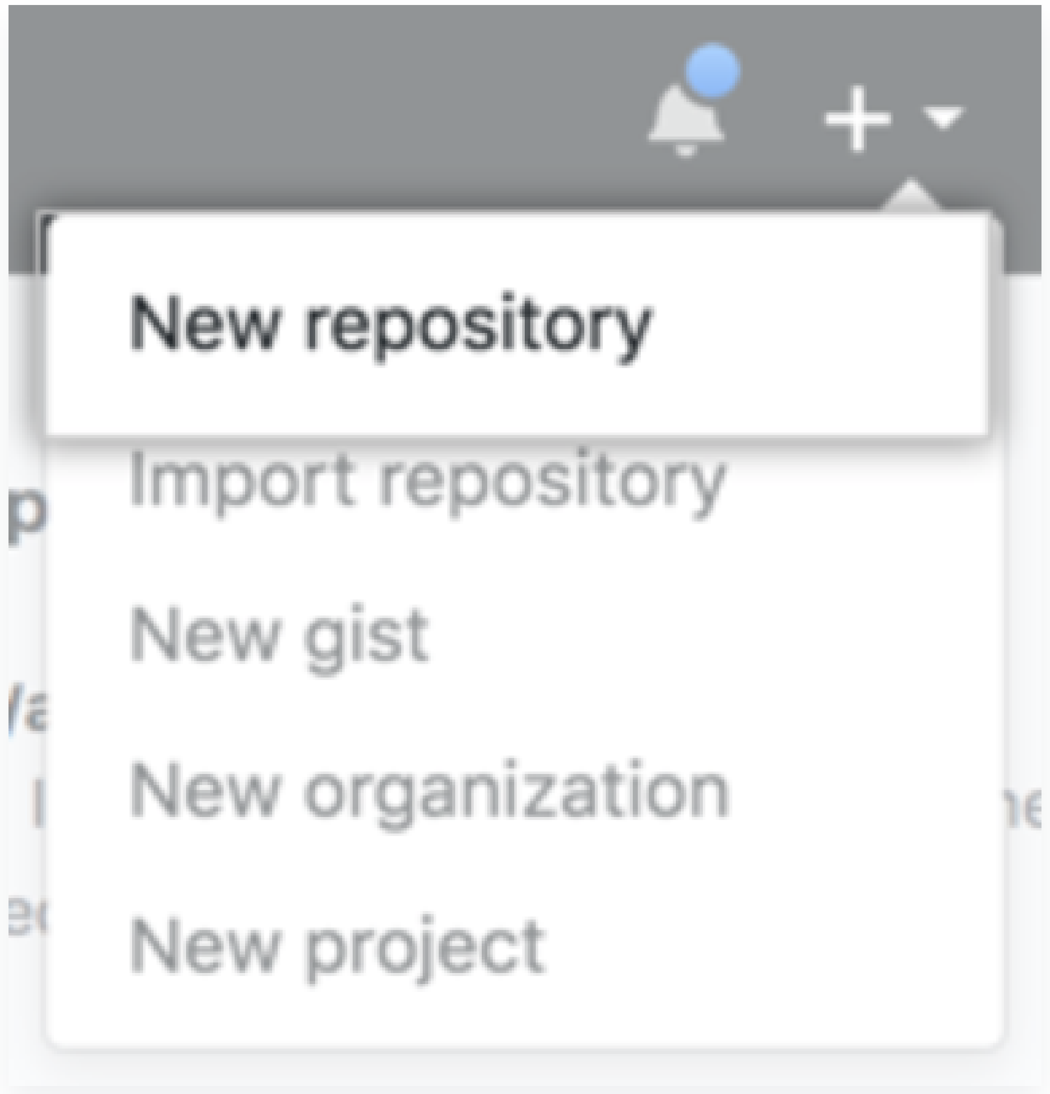
- never alters the original repo (without a Pull Request)
- great for: using a repo as a template, refining someone else's approach, safe collaboration

Cloning:

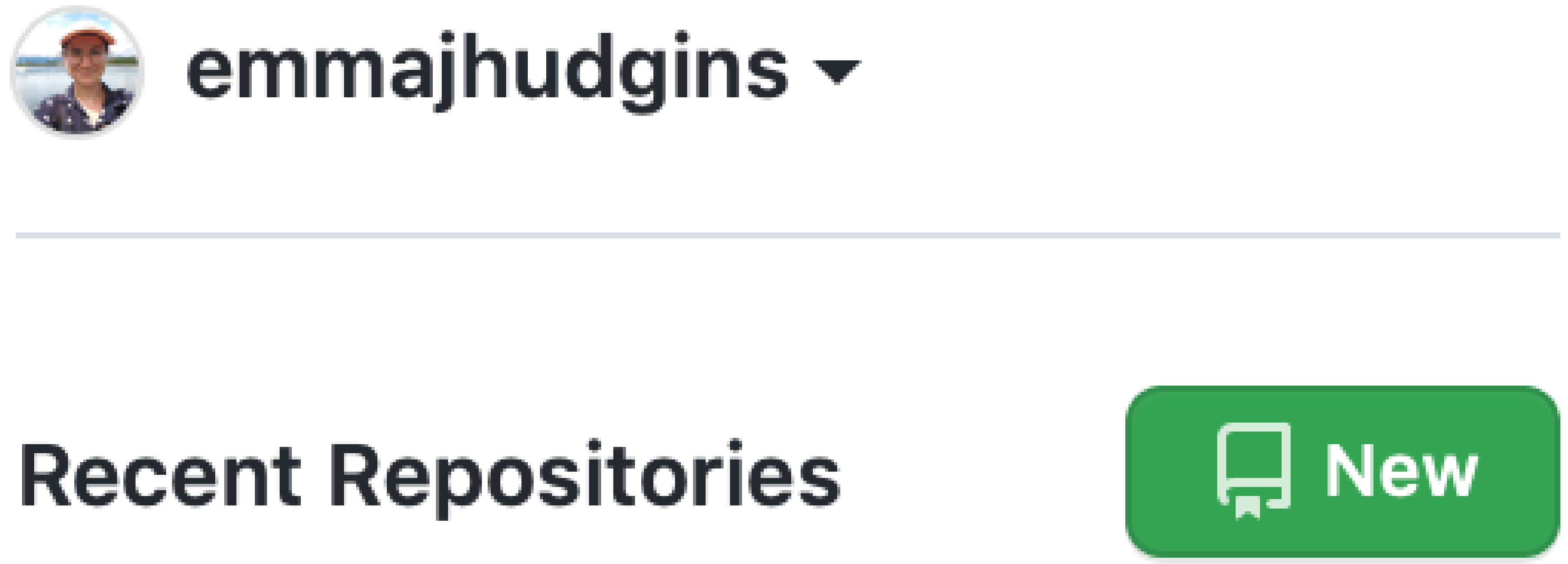
- Depending on collaborator permissions, you may be able to alter the main repo
- Can be used for collaboration (most often by working in a branch and submitting Pull Requests)
- Can also just be a way to download code

Screenshots of the steps

❶ In the upper-right corner of any page, use the + drop-down menu, and select **New repository**.

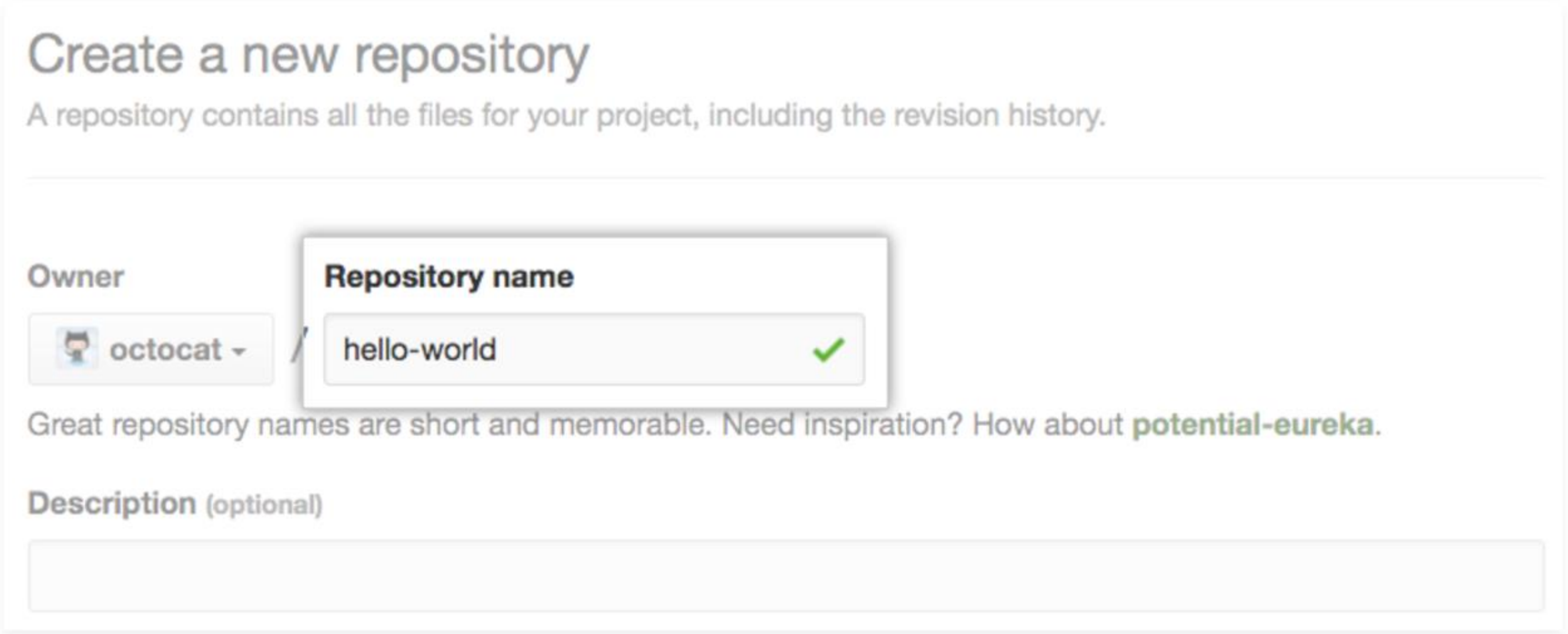


OR click the green button in the left pane



github.com

2 Type a short, memorable name for your repository. For example, "hello-world".



Create a new repository

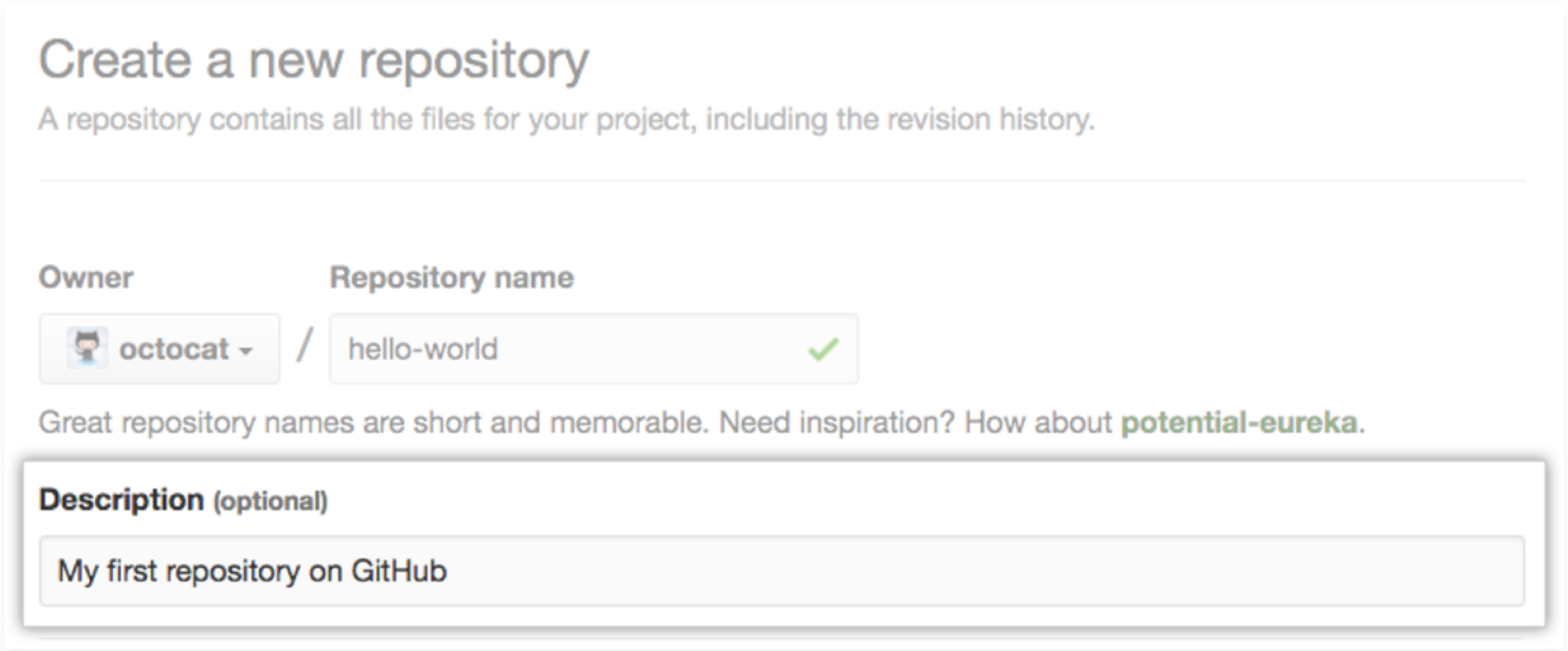
A repository contains all the files for your project, including the revision history.

Owner: octocat / Repository name: hello-world

Great repository names are short and memorable. Need inspiration? How about potential-eureka.

Description (optional)

3 Optionally, add a description of your repository. For example, "My first repository on GitHub."



Create a new repository

A repository contains all the files for your project, including the revision history.

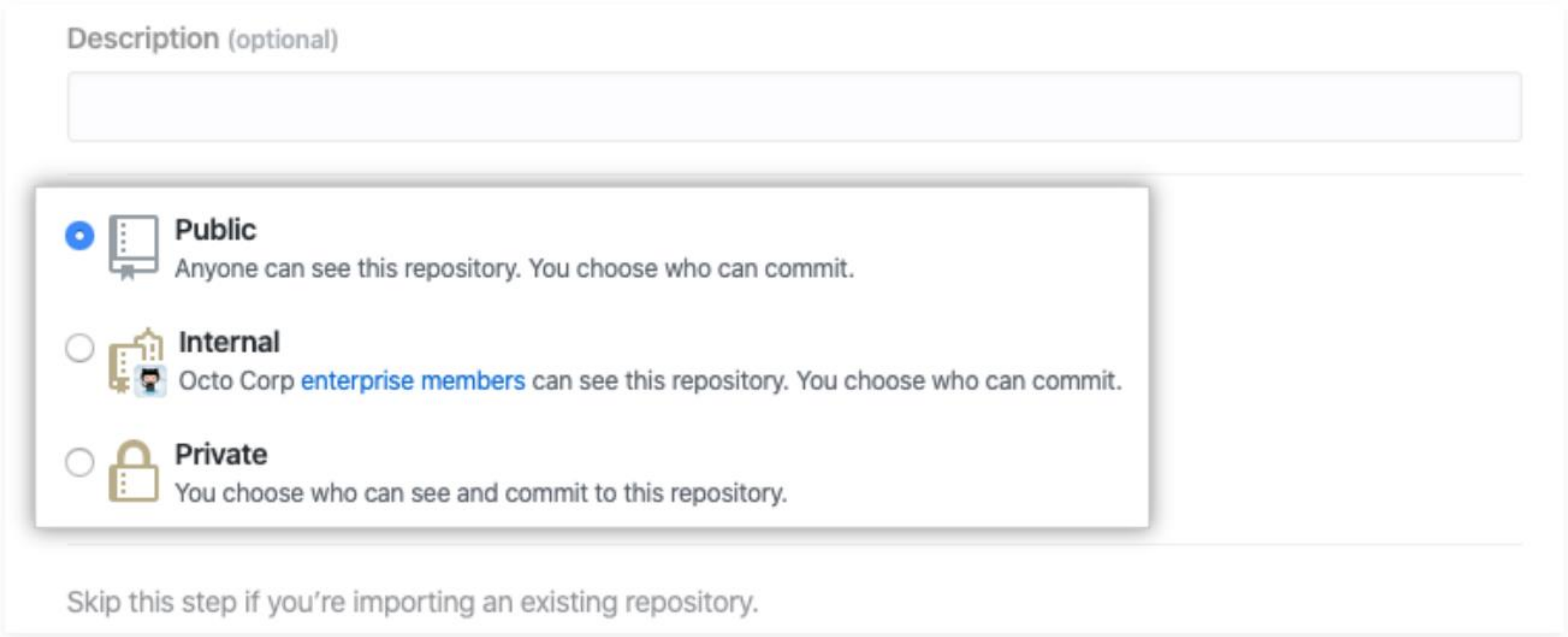
Owner: octocat / Repository name: hello-world

Great repository names are short and memorable. Need inspiration? How about **potential-eureka**.

Description (optional)

My first repository on GitHub

4 Choose a repository visibility. For more information, see "About repositories."


Description (optional)

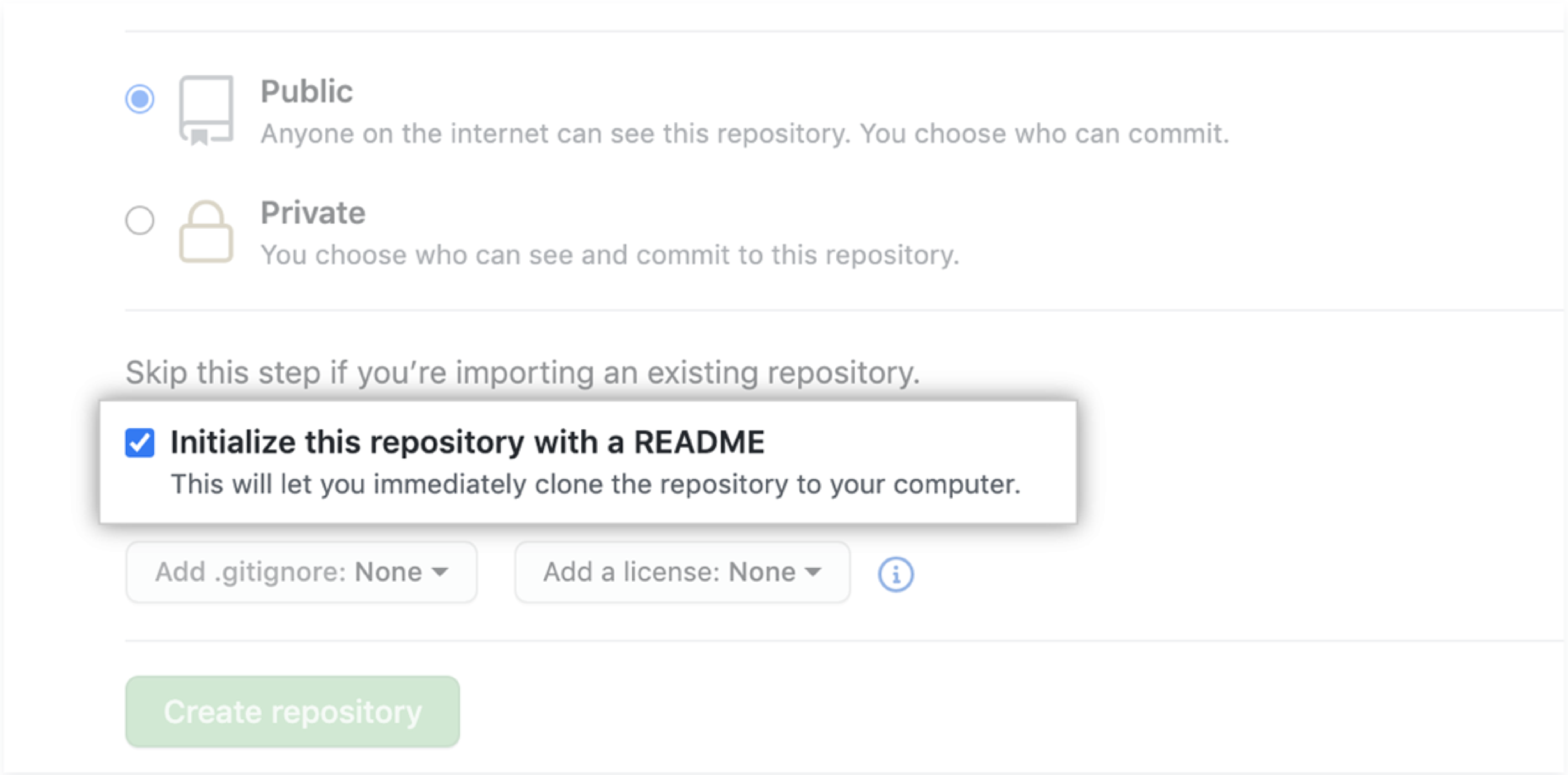
[✓ Public] Anyone can see this repository. You choose who can commit.

Internal Octo Corp enterprise members can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

5 Select **Initialize this repository with a README.**



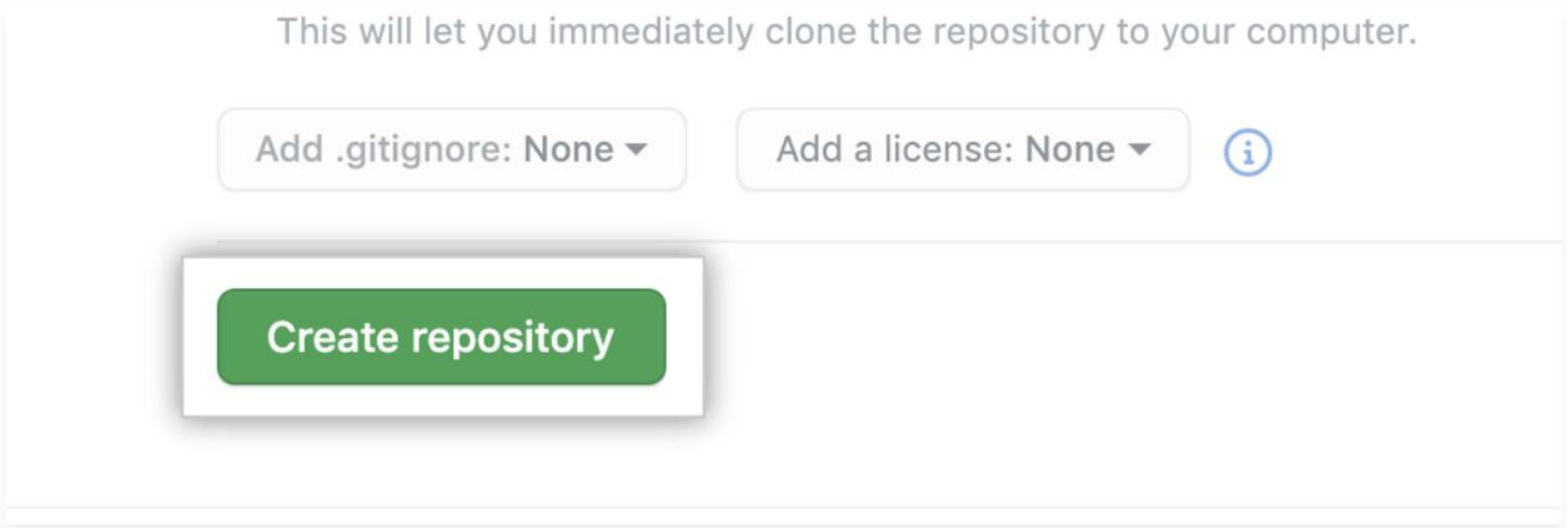
● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

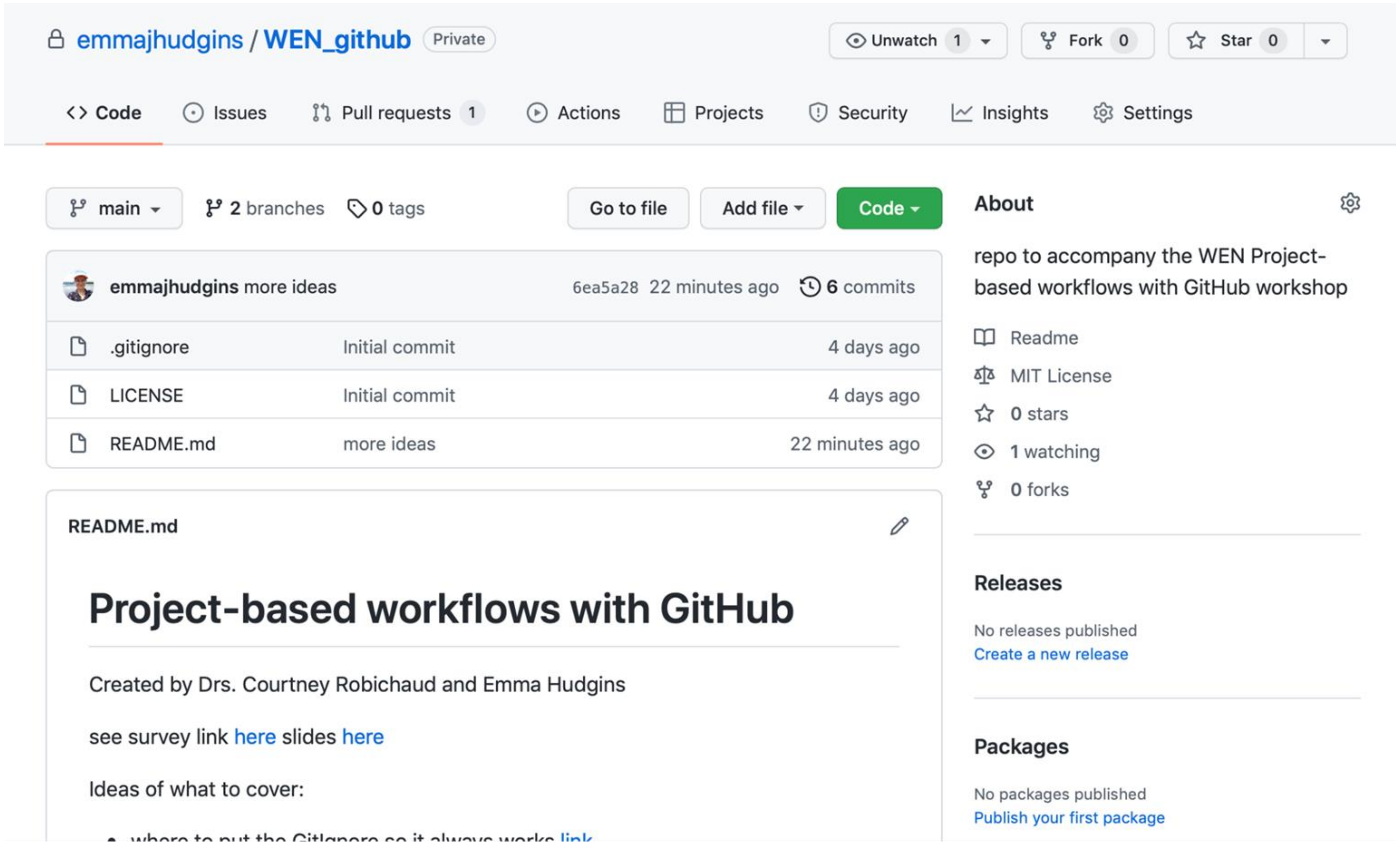
Skip this step if you're importing an existing repository.

✓ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

❻ Click **Create repository**.


This will let you immediately clone the repository to your computer.

Structure of a repo



emmajhudgins more ideas 6ea5a28 22 minutes ago 6 commits

.gitignore	Initial commit	4 days ago
LICENSE	Initial commit	4 days ago
README.md	more ideas	22 minutes ago

README.md

Project-based workflows with GitHub

Created by Drs. Courtney Robichaud and Emma Hudgins

see survey link here slides here

Ideas of what to cover:

About

repo to accompany the WEN Project-based workflows with GitHub workshop

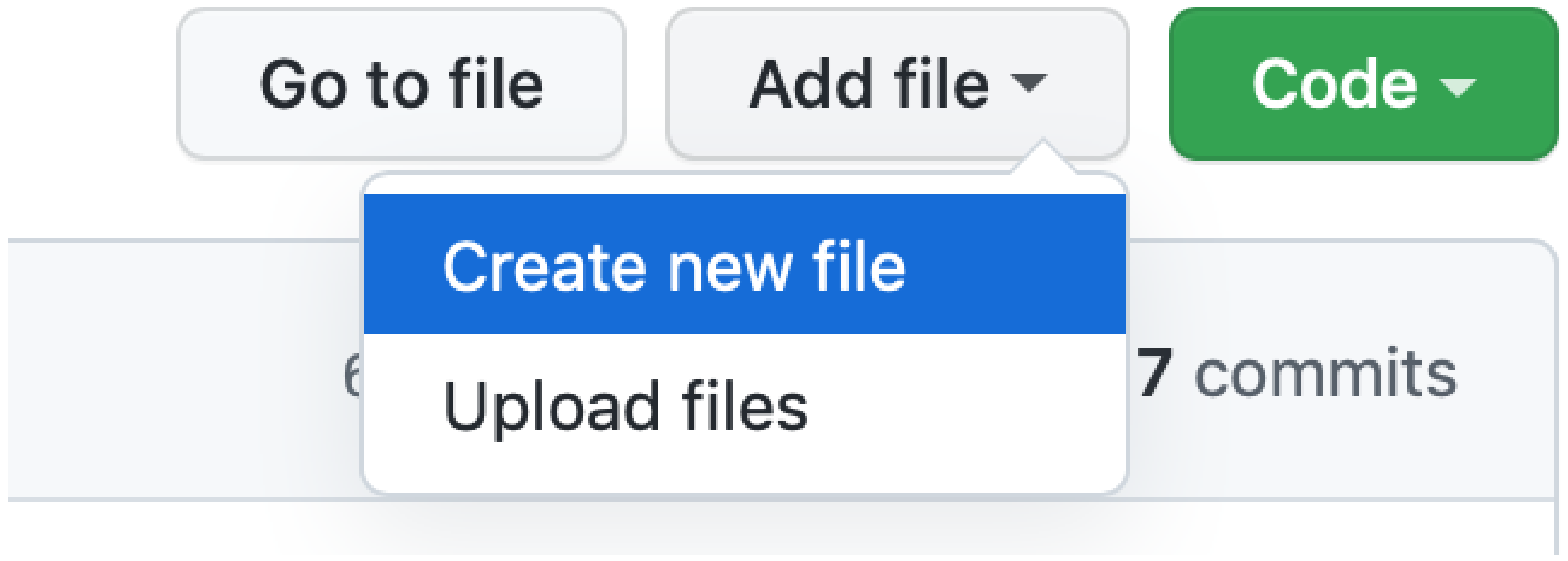
Readme
MIT License
0 stars
1 watching
0 forks

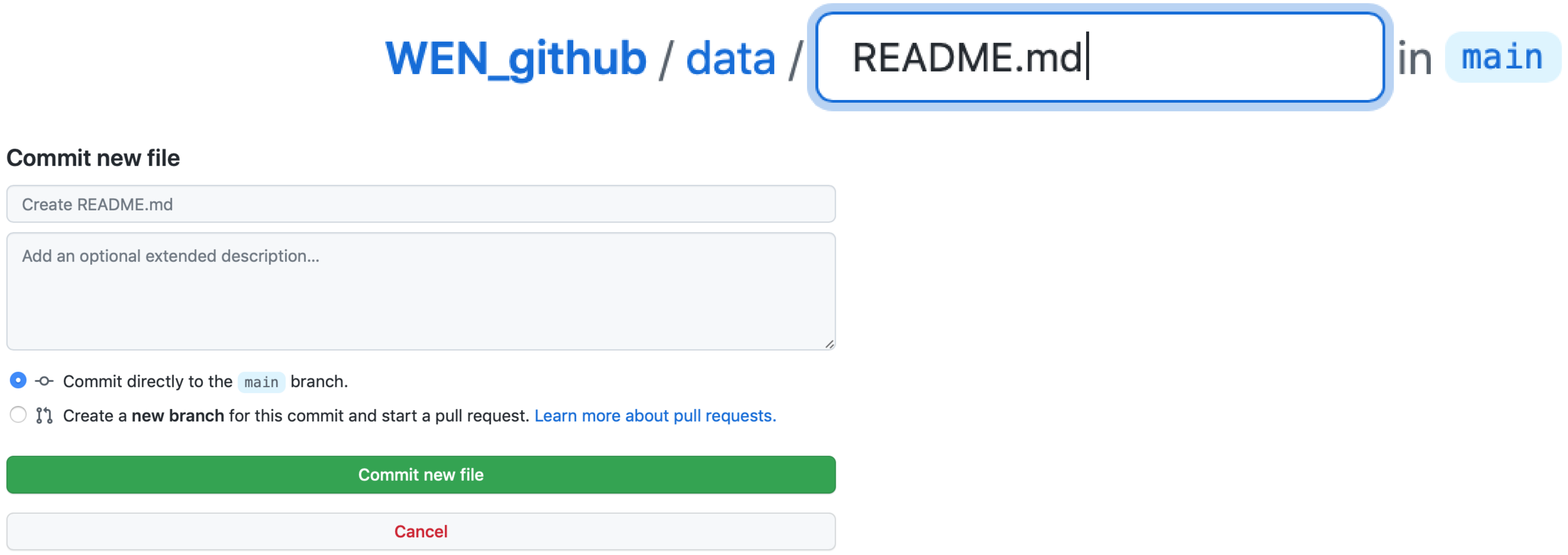
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package




WEN_github / data / README.md in `main`

Commit new file

Create README.md

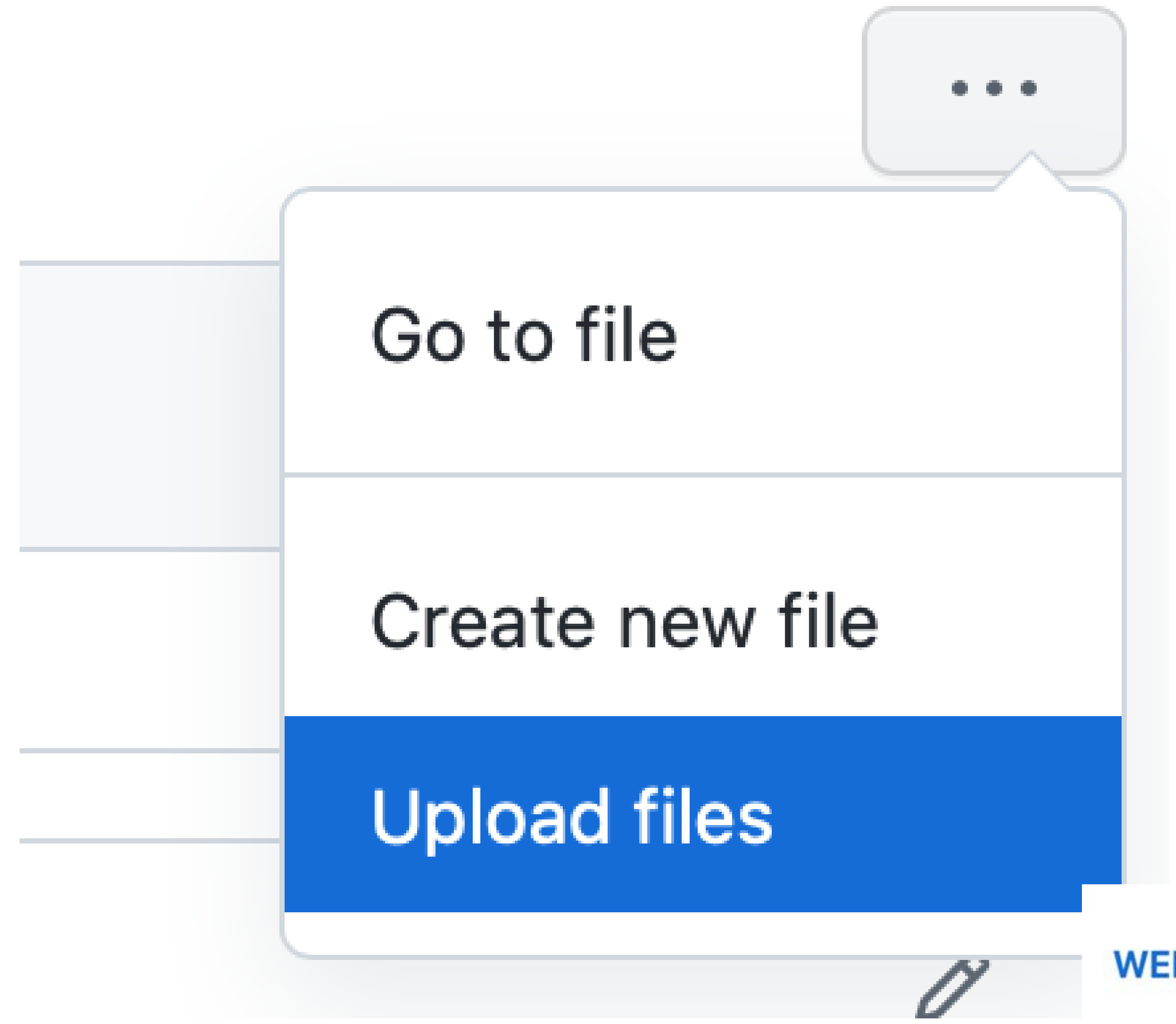
Add an optional extended description...

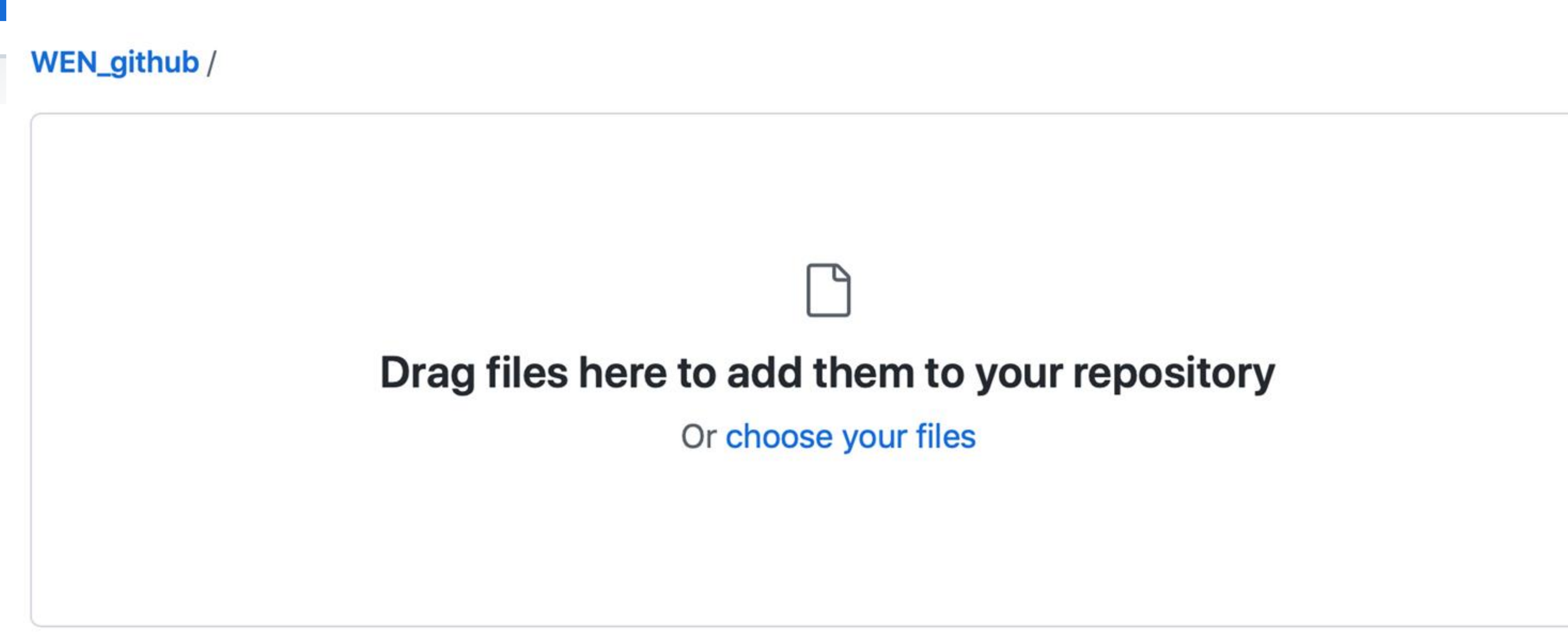
[✓ Commit directly to the `main` branch.]

Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Commit new file

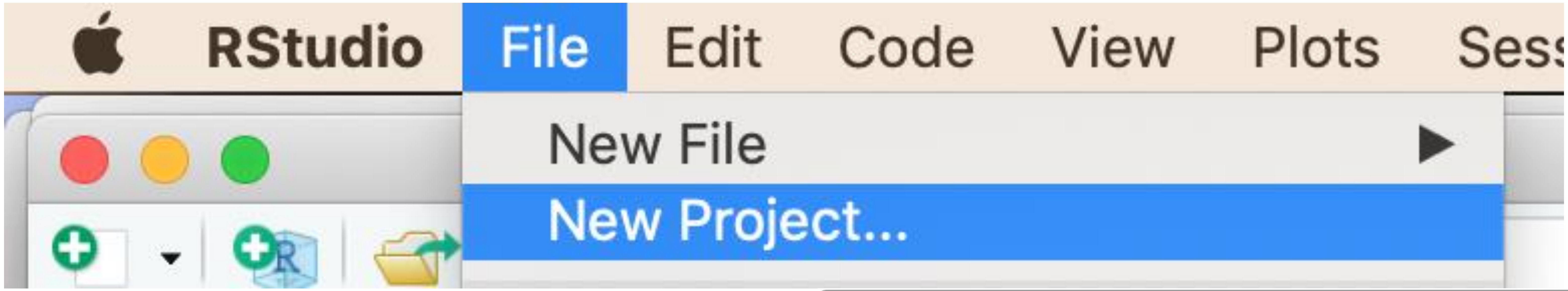
Cancel

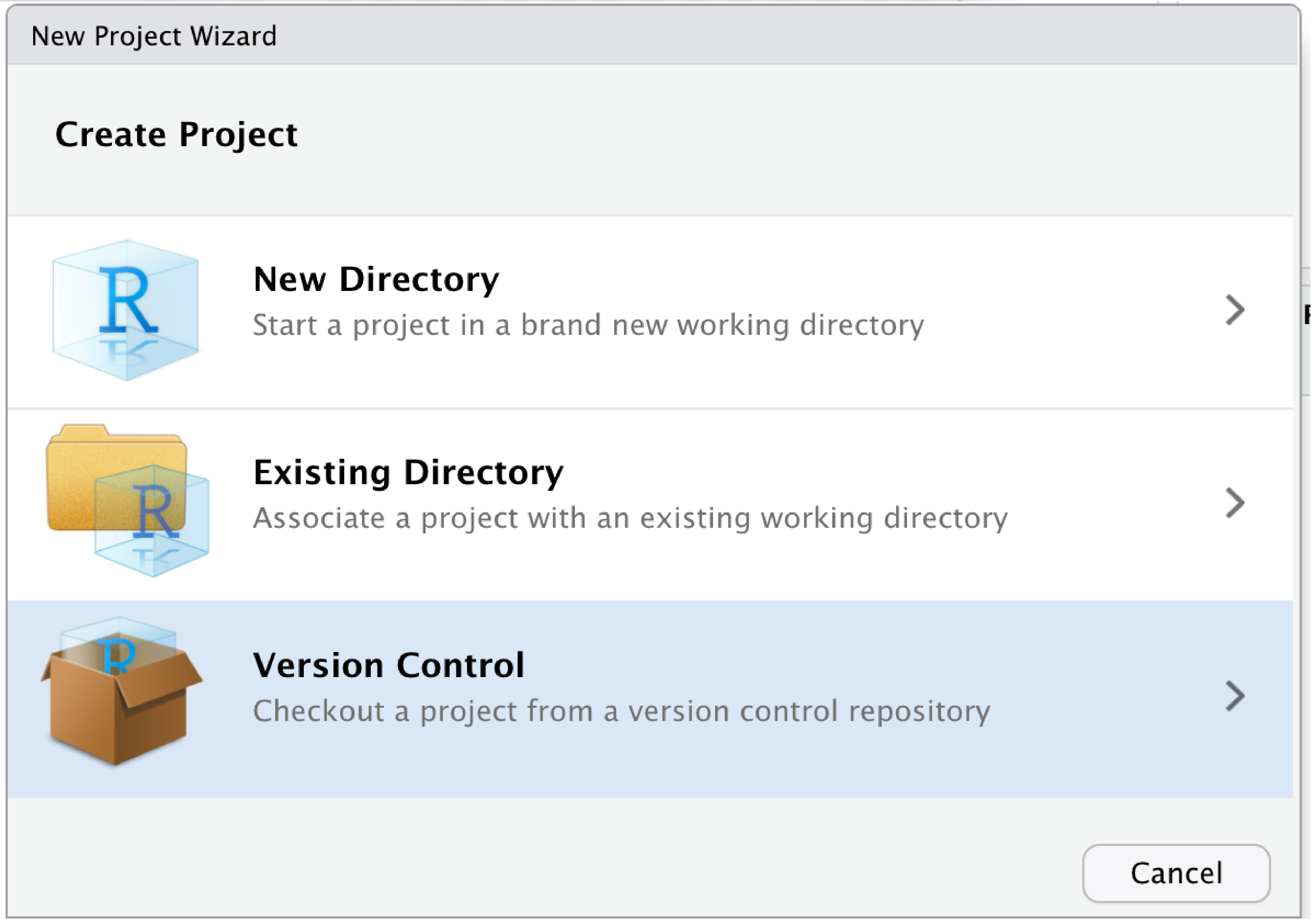



WEN_github /

Drag files here to add them to your repository

Or choose your files




New Project Wizard

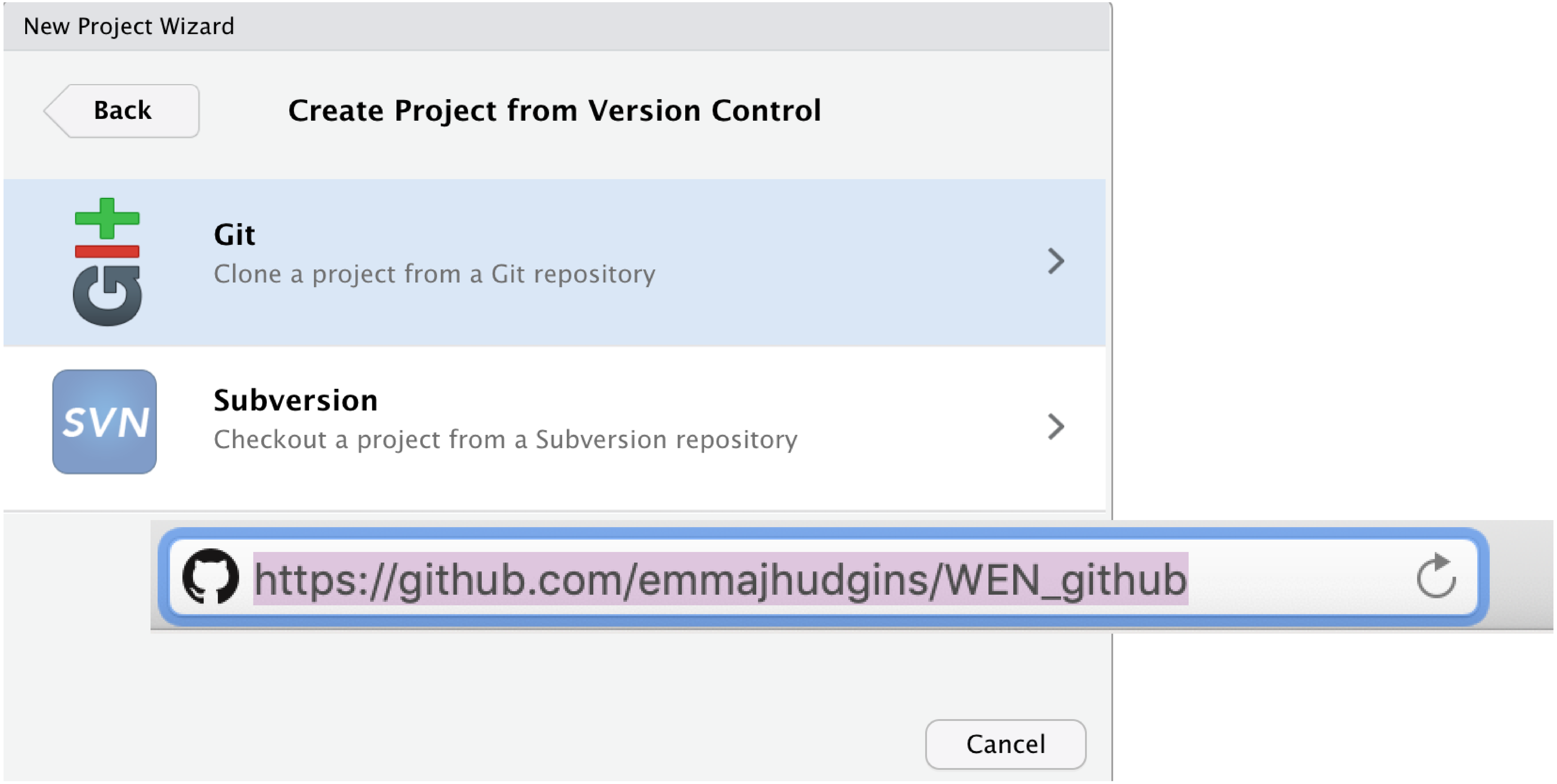
Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel



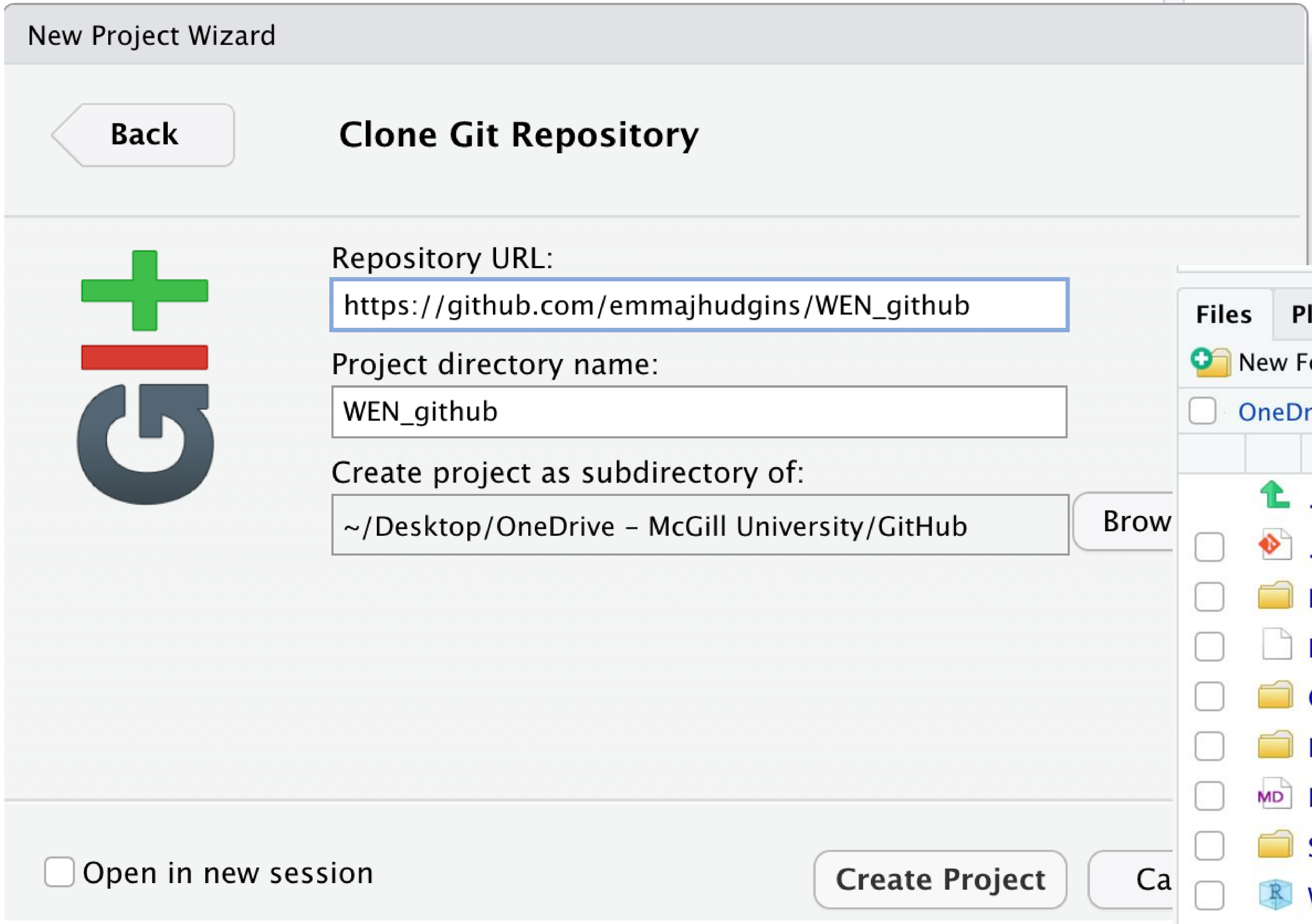
New Project Wizard

Create Project from Version Control

Git
Clone a project from a Git repository

Subversion
Checkout a project from a Subversion repository

https://github.com/emmajhudgins/WEN_github


New Project Wizard

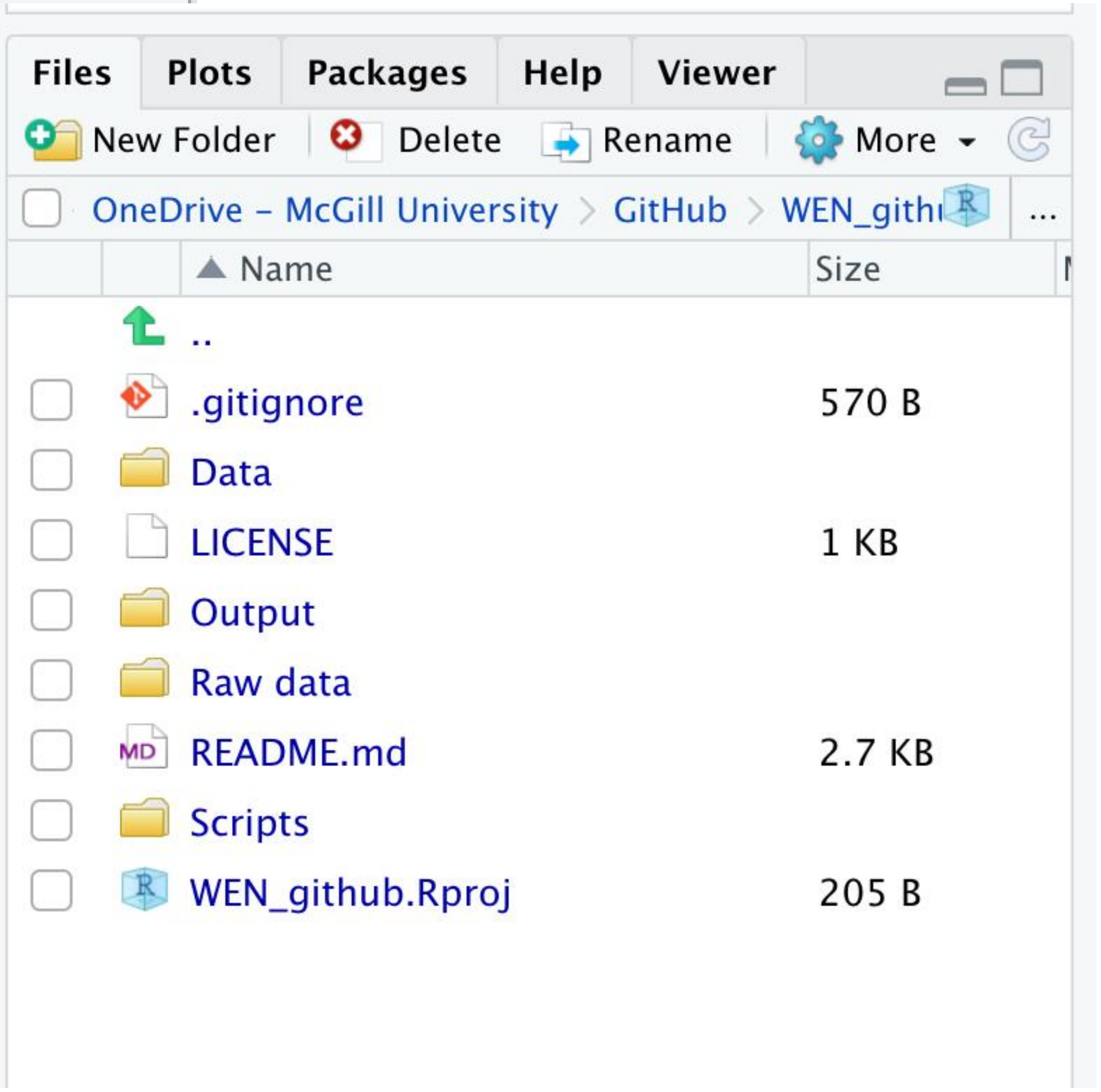
Back

Clone Git Repository

Repository URL:
https://github.com/emmajhudgins/WEN_github

Project directory name:
WEN_github

Create project as subdirectory of:
~/Desktop/OneDrive - McGill University/GitHub

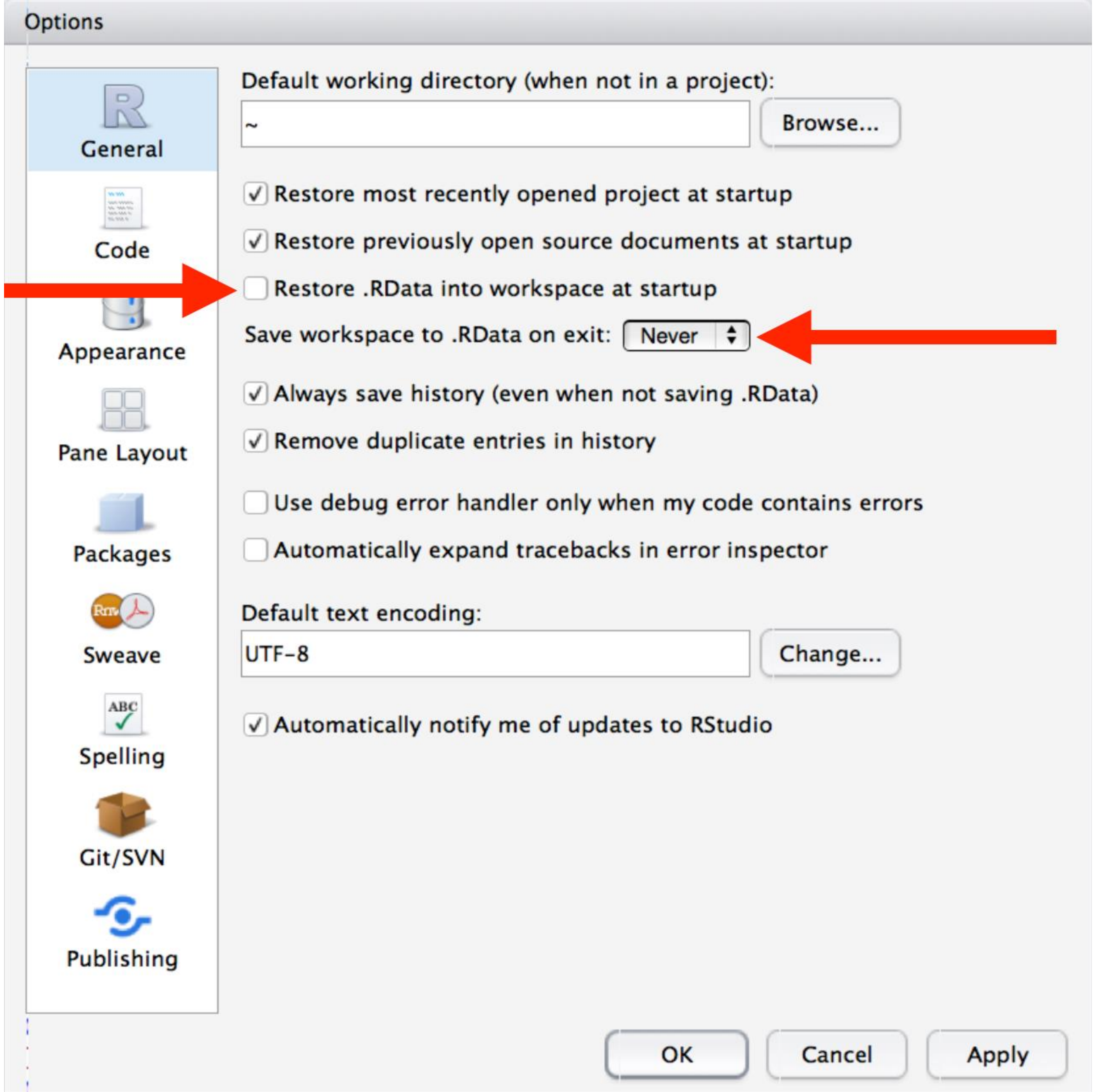


OneDrive – McGill University > GitHub > WEN_githu

Name	Size
..	
.gitignore	570 B
Data	
LICENSE	1 KB
Output	
Raw data	
README.md	2.7 KB
Scripts	
WEN_github.Rproj	205 B

Additional Information

Check/change your settings in R:



Default working directory (when not in a project):

~

[✓] Restore most recently opened project at startup

[✓] Restore previously open source documents at startup

[ ] Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

[✓] Always save history (even when not saving .RData)

[✓] Remove duplicate entries in history

[ ] Use debug error handler only when my code contains errors

[ ] Automatically expand tracebacks in error inspector

Default text encoding:

UTF-8

[✓] Automatically notify me of updates to RStudio

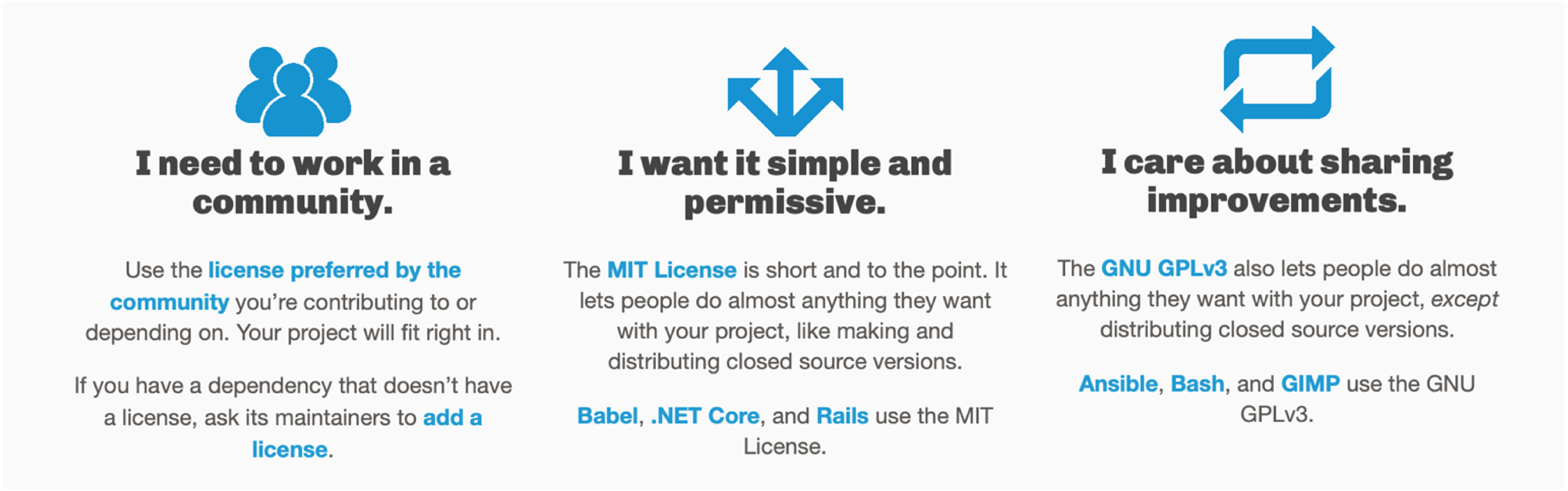
Git Ignore (.gitignore)

Choose a template based on your main programming language (R template ignores files like `.RHistory`)

Some examples of files you probably want to ignore:

- Sensitive information (e.g. passwords)
- Binary files such as `.Rdata`.
- **Files > 50MB**. Git is specifically made for **code** (e.g. `.R`) and does not intend to track all changes in large data files (these can be uploaded in ‘releases’ with DOIs through Zenodo).
- *temporary files/folders* with ‘disposable’ content

Licenses



I need to work in a community.

Use the **license preferred by the community** you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to **add a license**.

I want it simple and permissive.

The **MIT License** is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

Babel, **.NET Core**, and **Rails** use the MIT License.

I care about sharing improvements.

The **GNU GPLv3** also lets people do almost anything they want with your project, *except* distributing closed source versions.

Ansible, **Bash**, and **GIMP** use the GNU GPLv3.

choosealicense.com

Ideal Project Structure

Raw Data: Data imported into your project. Metadata includes date of download or collection, original source and re-use info

(Derived) Data: Data you have altered e.g., merging databases, cleaning up data, subsets, etc.

Scripts: Code (can separate by language)

Output: Figures, tables, results

Every folder should contain a README to explain what it contains

Meta-data/ReadMe best practices

- Include package version information and any external software used
- Describe files in a logical order
- Describe any column/variable names (especially units)
- Include which scripts outputs come from (helpful for new users and future you)

File Naming

- Be as descriptive as possible
- Can add leading numbers to scripts that indicate order they should be run e.g.
  - 01-data_processing.R
  - 02-model_fitting.R
- Avoid dates/overly generic names
- Name output similarly to script that generated it
- Use hyphens and underscores, not spaces

Clean Coding

Be proactive

- Use #### #### or ctrl + shift + r to separate sections within scripts
- Describe each major step and why it’s done
- Put yourself in the shoes of the person reading the code for the first time
- Include code author names, software versions

More advanced GitHub

Collaboration

Branch **-** one set of version histories for a repo, including the ‘**main**’ original branch, and additional branches used to suggest changes, test out new ideas that may not work etc.

Pull request - a suggested commit (created in another branch or from a fork) that must be approved by the owner of the main branch

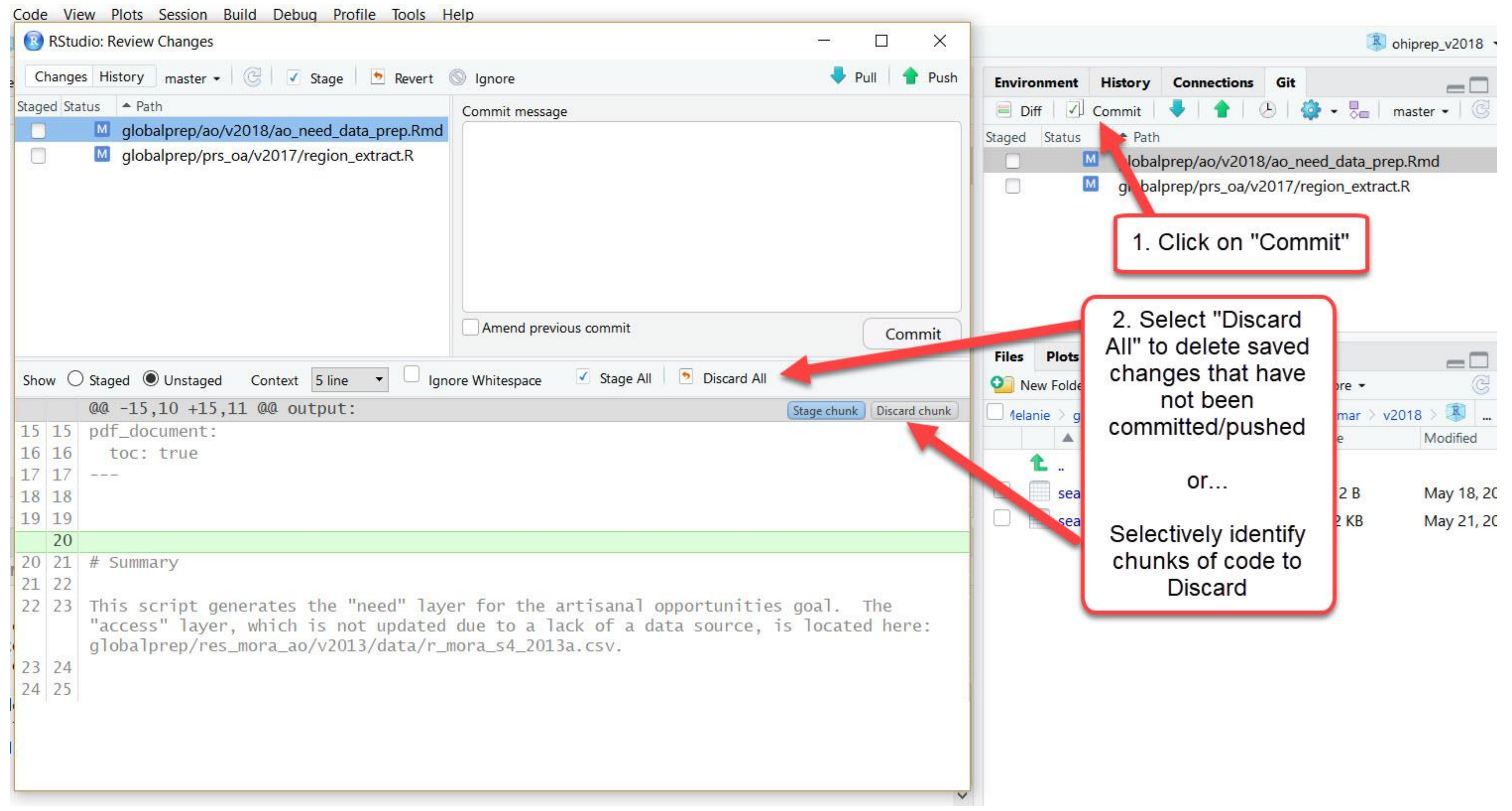
Pull often, commit after each change

Revert changes

Easier pre-commit, but possible post-commit too.

Pre-commit:

In RStudio, right click on a file and select ‘revert’



Code View Plots Session Build Debug Profile Tools Help
RStudio: Review Changes
Changes History master Stage Revert Ignore Pull Push
Staged Status Path
Commit message
Amend previous commit
Commit
Show Staged Unstaged Context 5 line Ignore Whitespace Stage All Discard All
Stage chunk Discard chunk
ohiprep_v2018
Environment History Connections Git
Diff Commit master
Files Plots
New Folder
Modified

globalprep/ao/v2018/ao_need_data_prep.Rmd
globalprep/prs_oa/v2017/region_extract.R

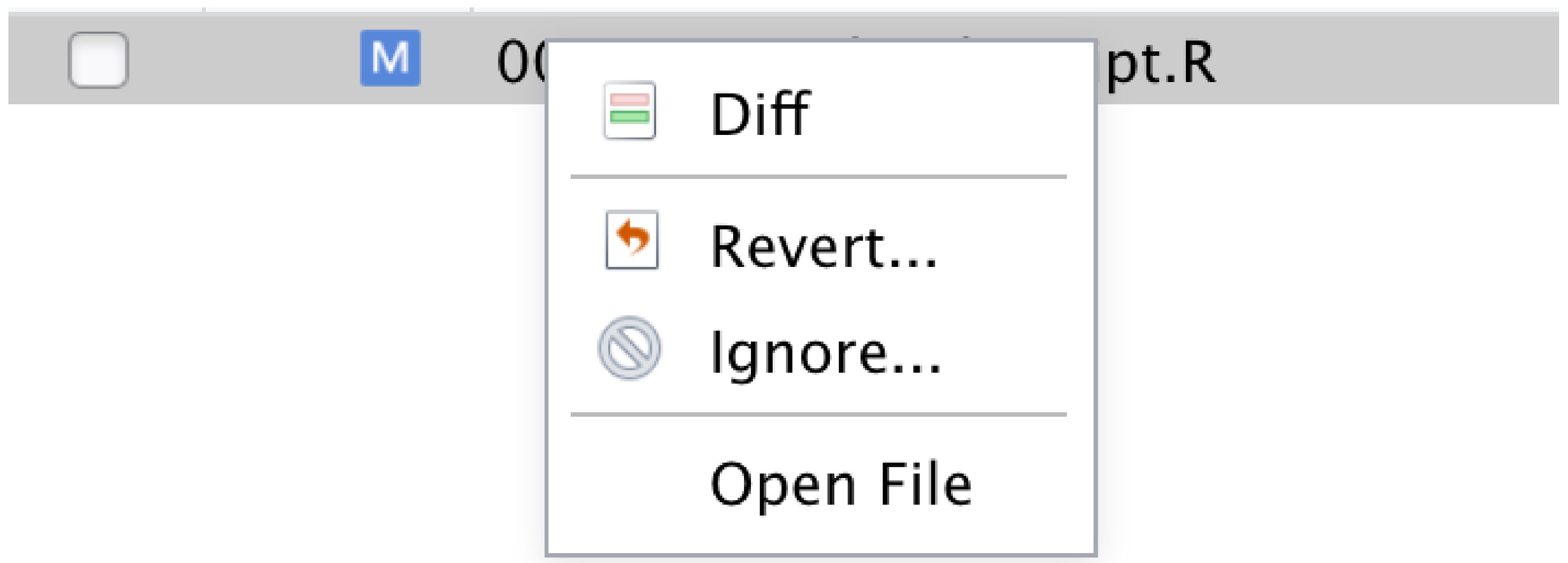
```
@@ -15,10 +15,11 @@ output:
pdf_document:
  toc: true
---



# Summary

This script generates the "need" layer for the artisanal opportunities goal.  The
"access" layer, which is not updated due to a lack of a data source, is located here:
globalprep/res_mora_ao/v2013/data/r_mora_s4_2013a.csv.


```

1. Click on "Commit"

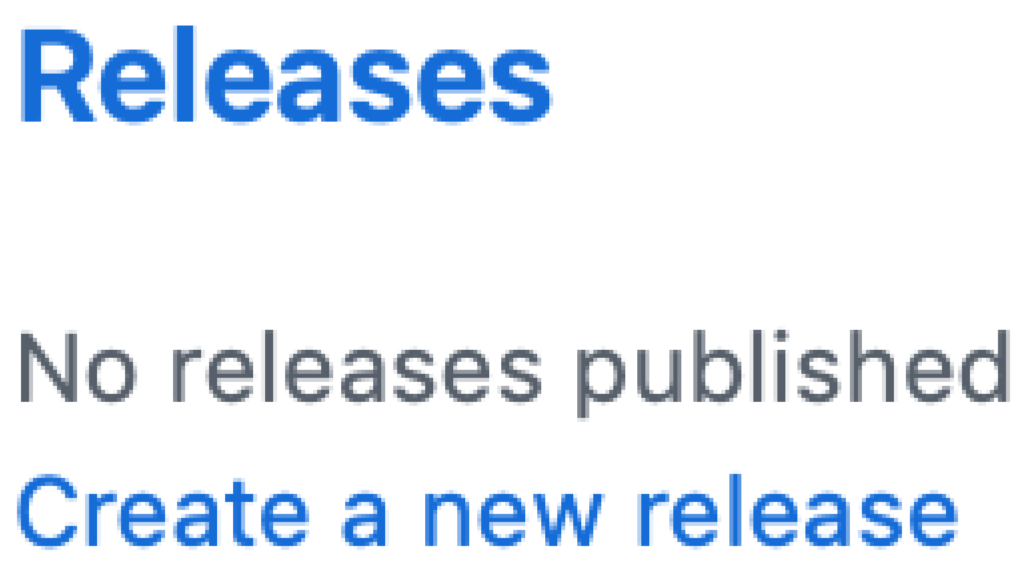
2. Select "Discard All" to delete saved changes that have not been committed/pushed

or...

Selectively identify chunks of code to Discard



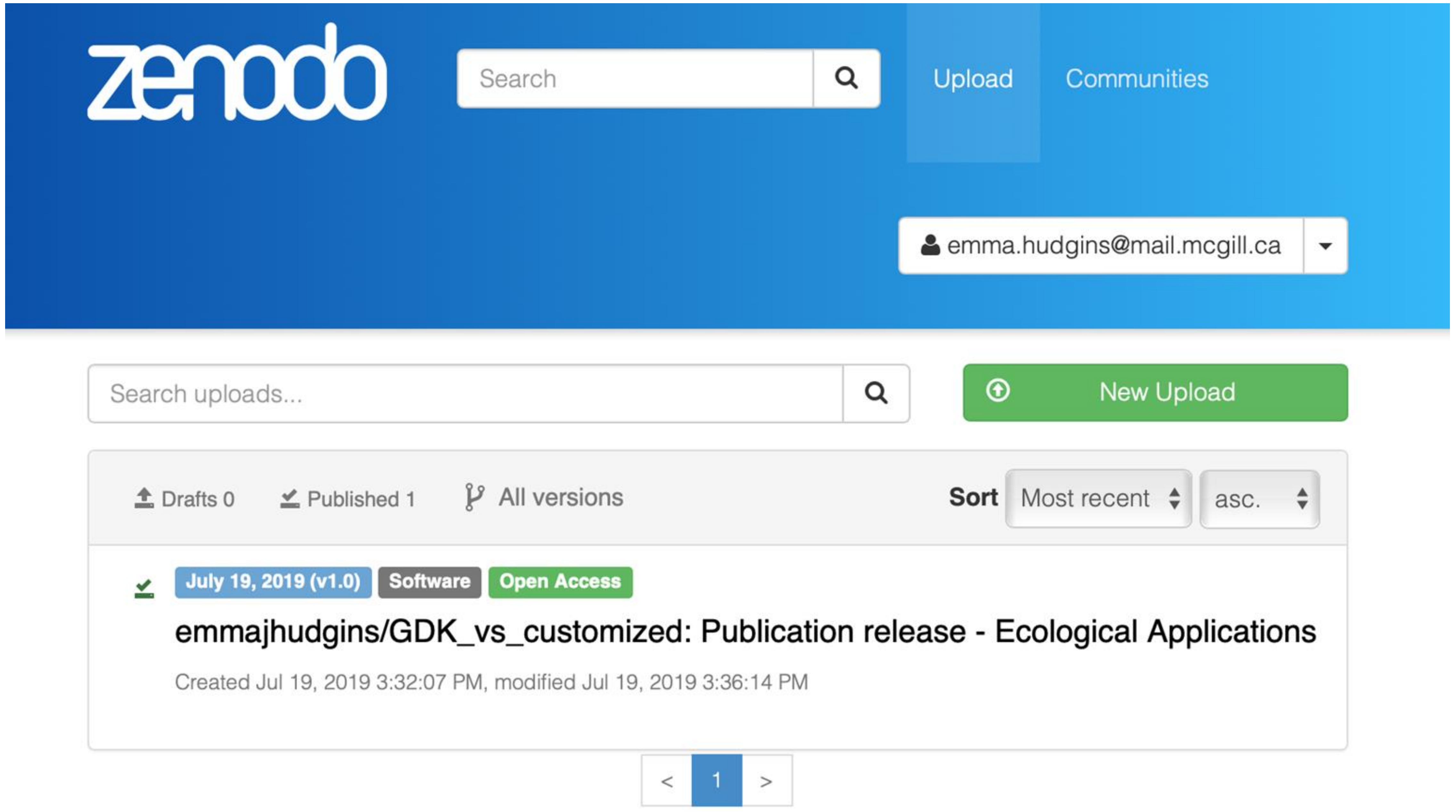
Releases, Zenodo & DOI creation



Releases

No releases published

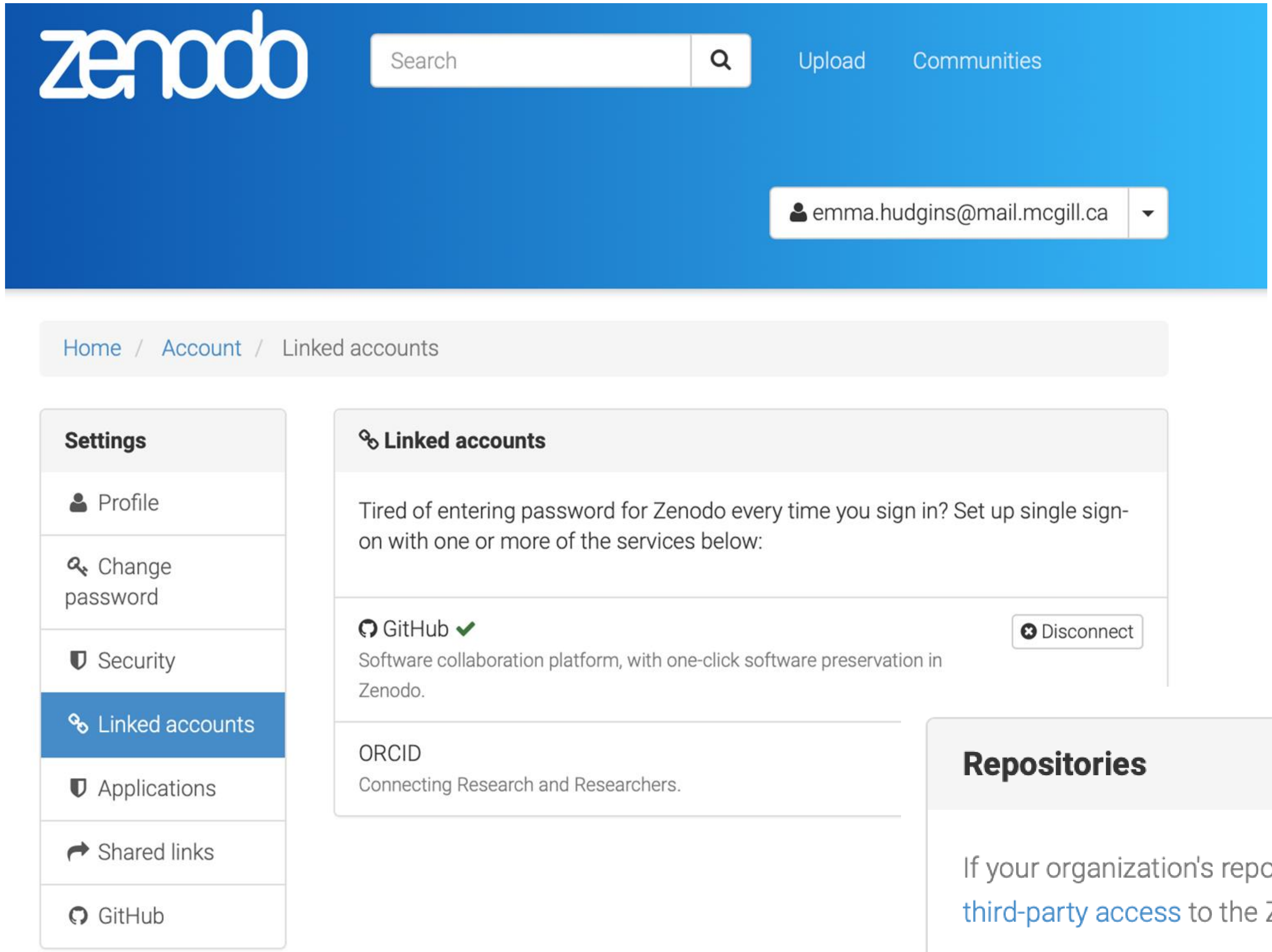
Create a new release



July 19, 2019 (v1.0) Software Open Access

emmajhudgins/GDK_vs_customized: Publication release - Ecological Applications

Created Jul 19, 2019 3:32:07 PM, modified Jul 19, 2019 3:36:14 PM



Home / Account / Linked accounts

Settings

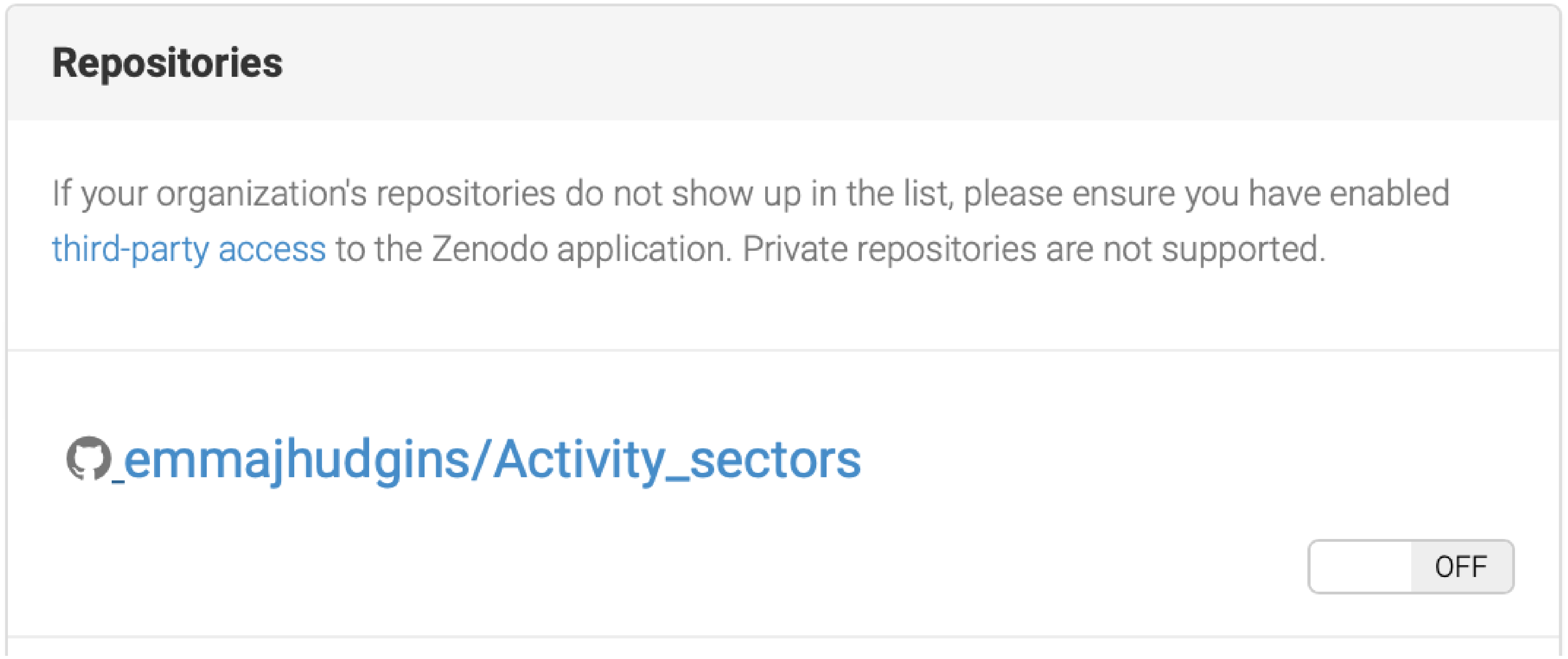
- Profile
- Change password
- Security
- Linked accounts
- Applications
- Shared links
- GitHub

Linked accounts

Tired of entering password for Zenodo every time you sign in? Set up single sign-on with one or more of the services below:

GitHub ✓ Disconnect
Software collaboration platform, with one-click software preservation in Zenodo.

ORCID
Connecting Research and Researchers.



Repositories

If your organization's repositories do not show up in the list, please ensure you have enabled third-party access to the Zenodo application. Private repositories are not supported.

emmajhudgins/Activity_sectors

OFF

Other helpful resources

https://datacarpentry.org/rr-version-control

https://carpentries-incubator.github.io/git-Rstudio-course/

https://www.markdownguide.org/basic-syntax/