

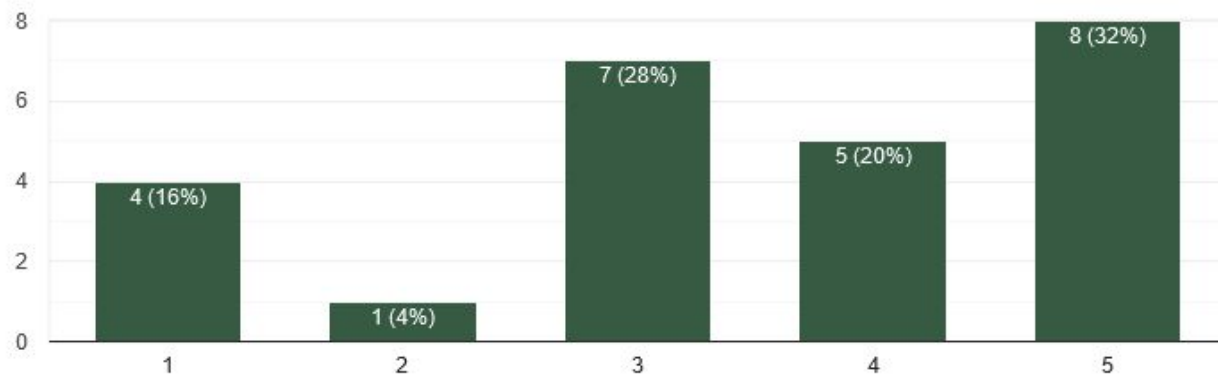
Project-based workflows with GitHub

Courtney Robichaud and Emma Hudgins
@cdrobich @emmajhudgins

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

What's your level of familiarity with R and RStudio?

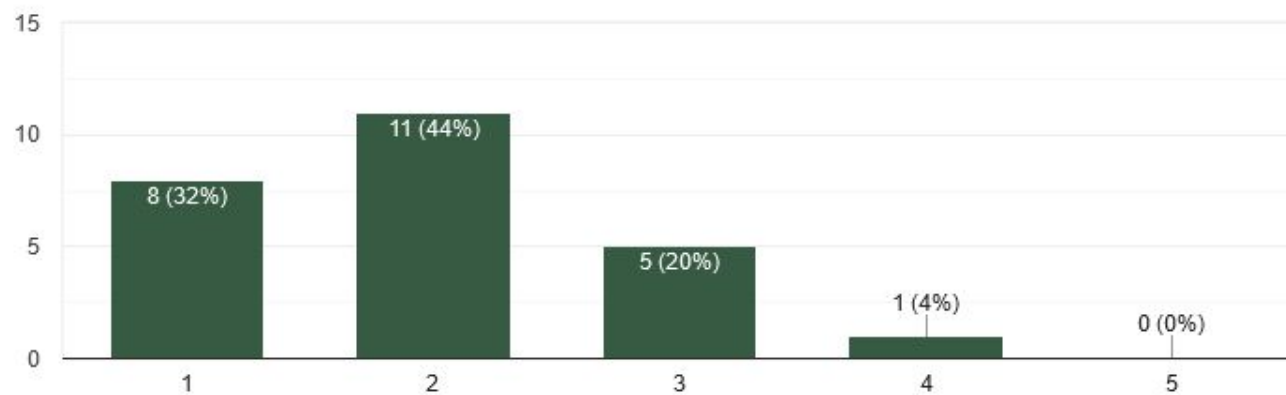
25 responses



What is your level of familiarity with Git/GitHub?



25 responses



What are your concerns about using or learning Git/GitHub?

Little coding experience

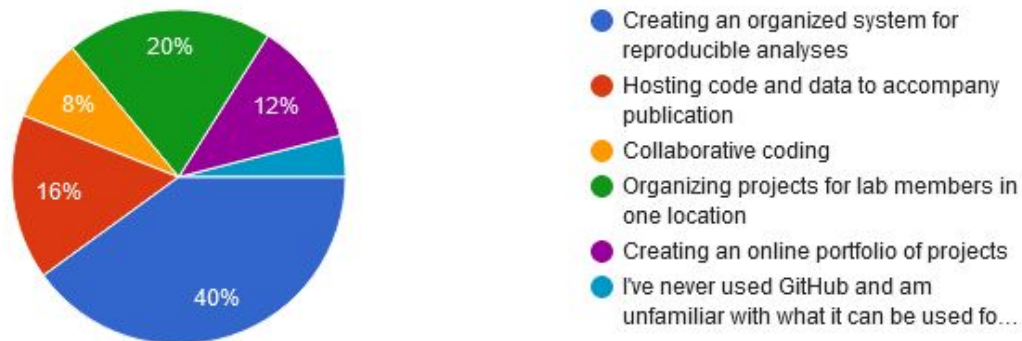
Never got the flow of it

The learning curve/difficulty for new users

I don't know how to use it in my research

What is your main priority when it comes to developing GitHub skills?

25 responses



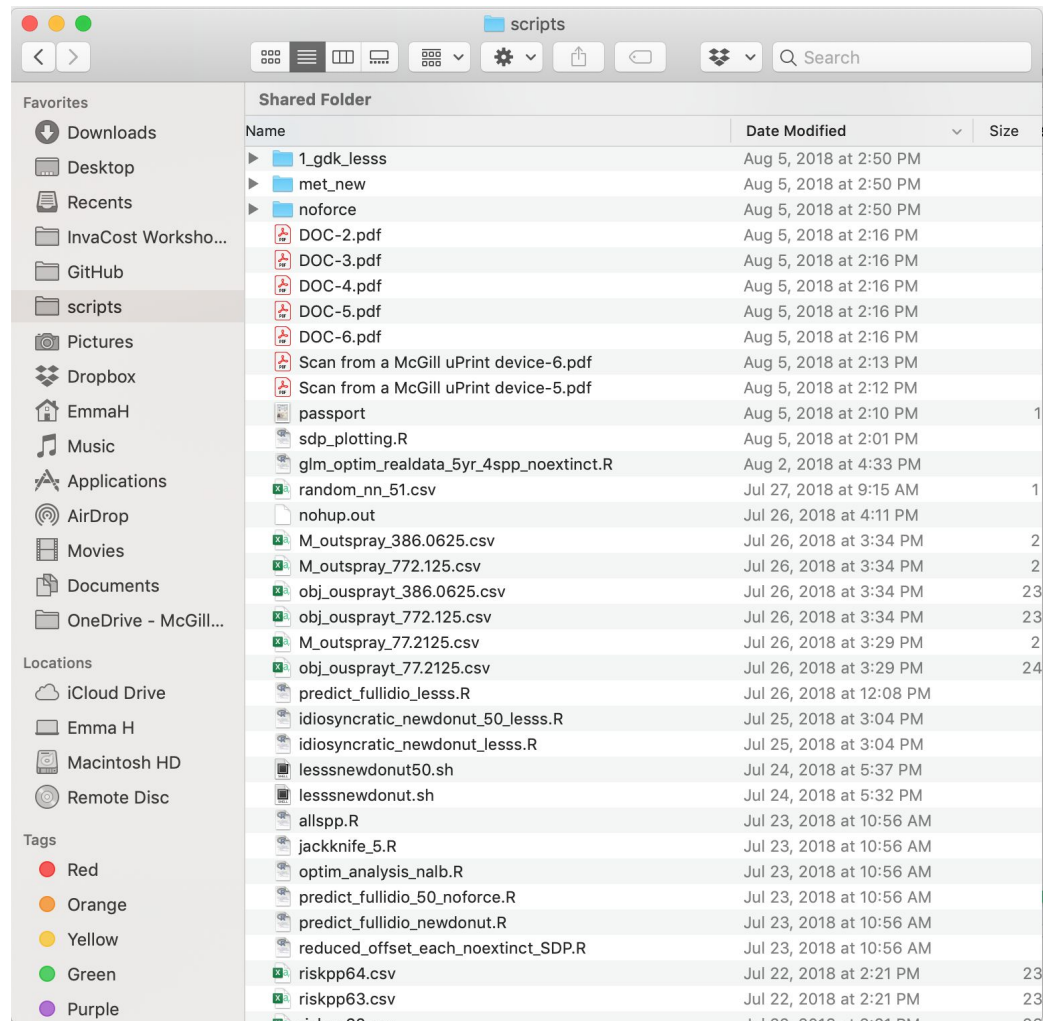
What we will cover:

- What are R projects, Git, and Github?
- Walking through how to use them
- Demo creating a repo, writing script, generating figures, and committing them to a repo
- You follow along with our demo and make your own!

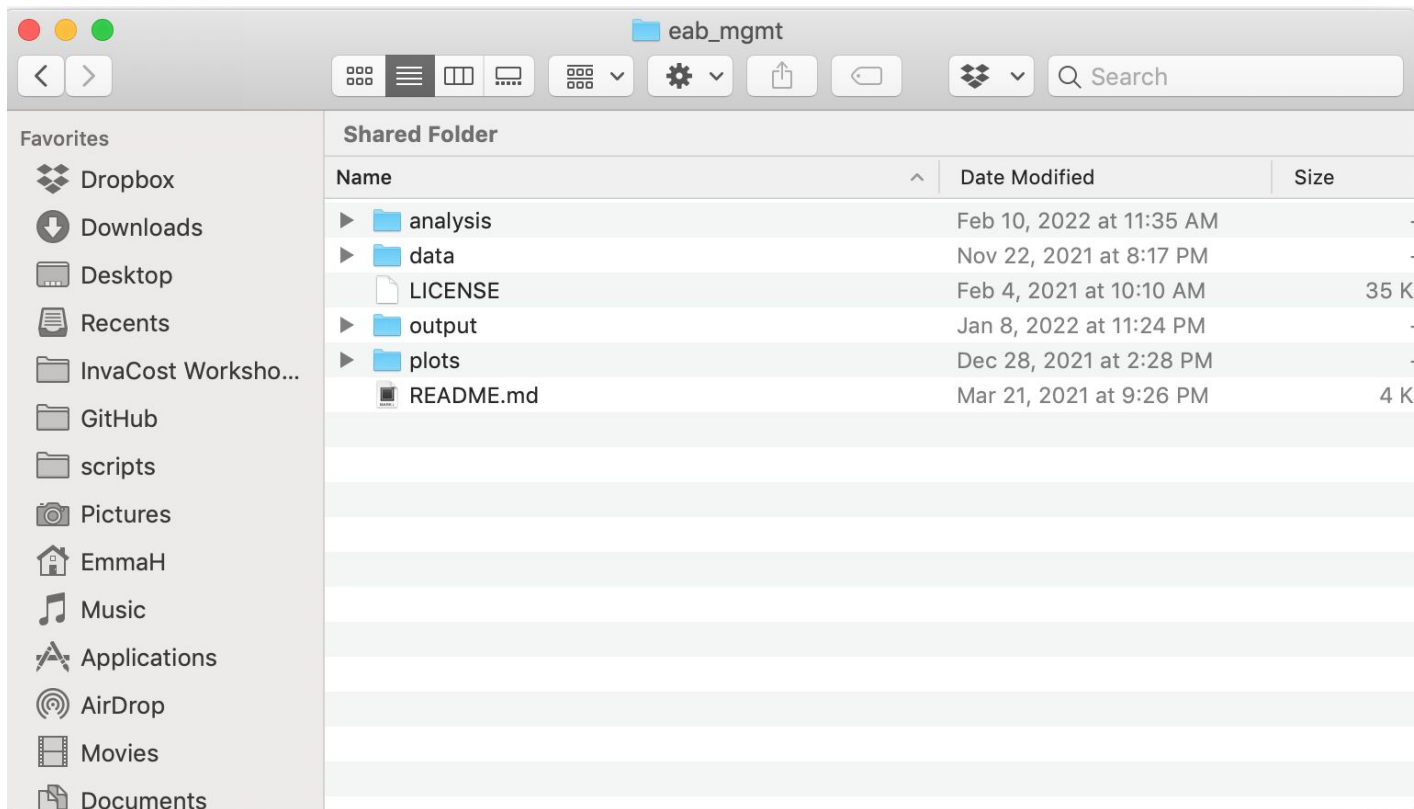
The power of projects, Git and GitHub

Your current organization

Could look something like this

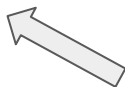


Your ideal organization





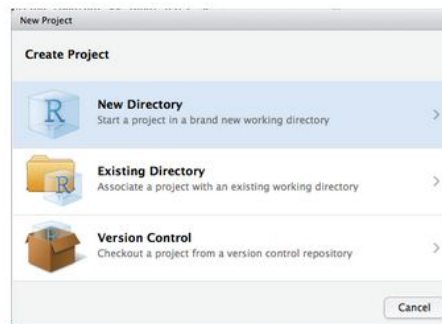
Push to GitHub



Commit often



R project



R Projects

```
setwd("C:/Users/cdrobich/Desktop/M.Sc/Birds")
```

code and data



Thursday, January 13, 2022 at 1:44 PM



nestinitdata_final.csv
77 KB



TRIG PR path linkag...
5.3 KB



[Download All](#) • [Preview All](#)

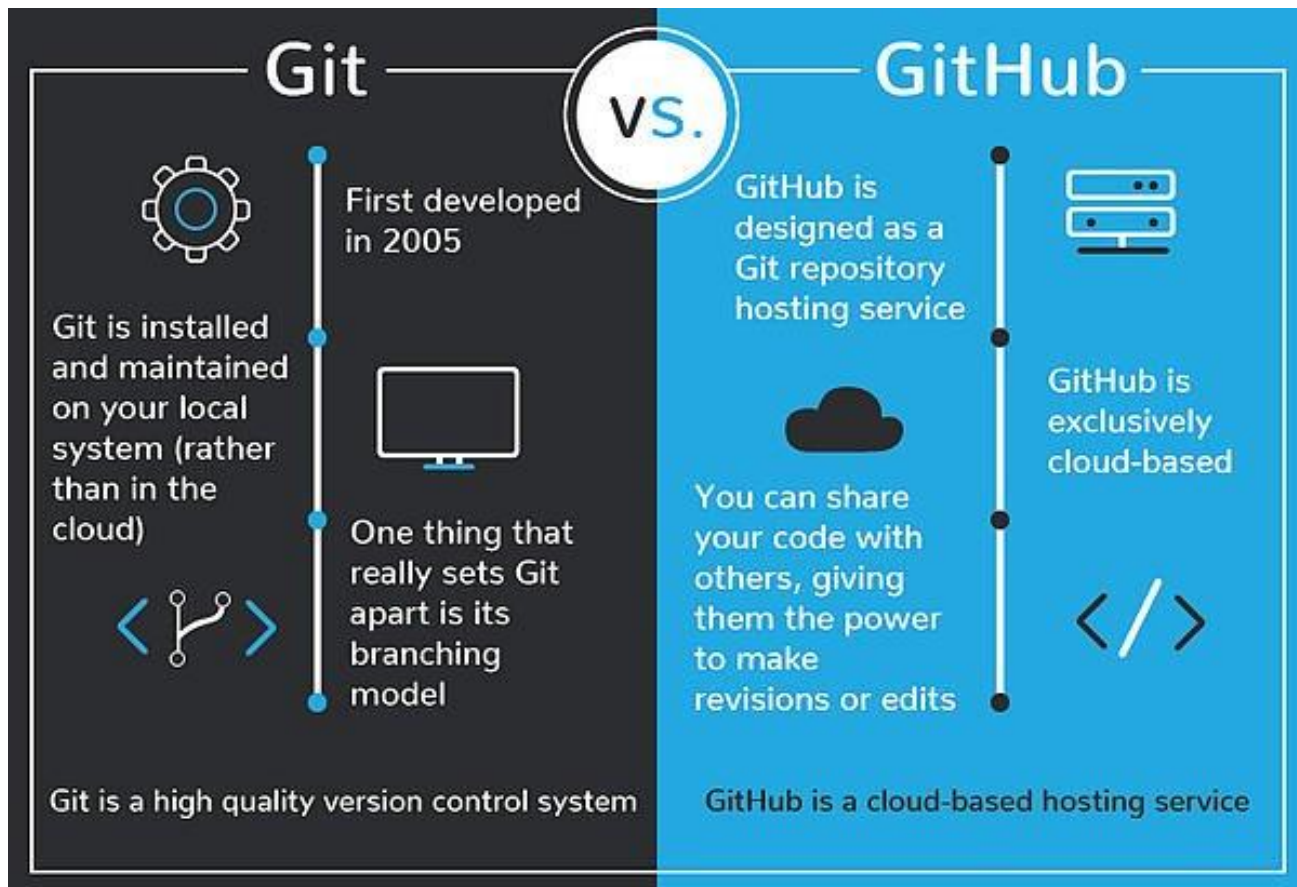
You replied to this message.

[External Email]

Hi Emma,
Thanks again for your help today! The code and data are attached if you're able to help with the code for that plot.
Thanks!

What is a project and why is it better?

- All your data (raw and manipulated), scripts, and output are saved in separate folders
- Does not call on anything system-specific
- Well commented so others (including future you) understand



What else can GitHub do?

- Can be a **cloud storage** service for any type of file
- “**Forking**” allows people to use others’ projects as **templates** for their own
- Provides a **hosting** service for web content
- Allows you to freeze your work at a given moment in time as a ‘**release**’
which can be linked to a DOI (Required by many journals/funders)
- Provides **integration** with other tools (e.g. OSF)

GitHub basics

Push, pull, commit, fork, clone

Create your project in GitHub and 'clone' to your machine. Then you will interact with Git by:

Repo(sitory) - one or more folders that have git functionality, GitHub repos are stored on the cloud

Push - send changes to the cloud

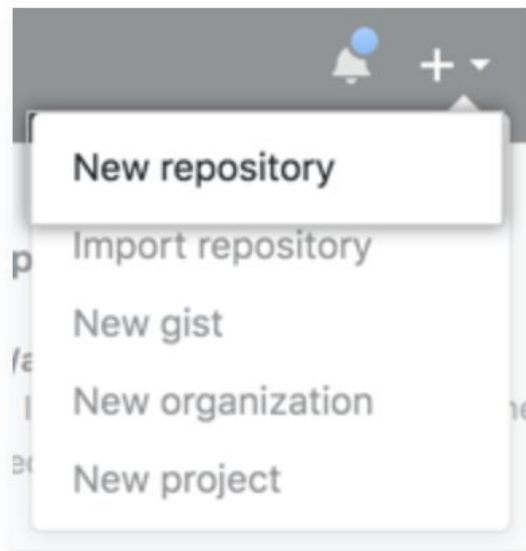
Pull - get changes from the cloud

Commit - create a named version of a set of one or more changes to the repo

Clone - copy an existing repo into your local github folder **such that it communicates with the original repo**

Fork - freeze an existing repo in time and copy it into your github folder **such that it does not communicate with the old repo**

- 1 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



emmajhudgins ▼

Recent Repositories

 **New**

- 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

Repository name

 octocat ▾ / 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.

Add .gitignore: None ▼

Add a license: None ▼



Create repository

6 Click **Create repository**.

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


Add .gitignore: None ▼

Add a license: None ▼



Create repository


Structure of a repo




 [emmajhudsons](#) / [WEN_github](#) Private


Unwatch 1 Fork 0 Star 0

[Code](#) [Issues](#) [Pull requests 1](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

main 2 branches 0 tags Go to file Add file Code

 **emmajhudsons** 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 Hudsons


see survey link [here](#) slides [here](#)


Ideas of what to cover:


• where to put the Gitignore so it always works [link](#)


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](#)

Go to file

Add file ▼

Code ▼

Create new file

Upload files

7 commits

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



Go to file

Create new file

Upload files

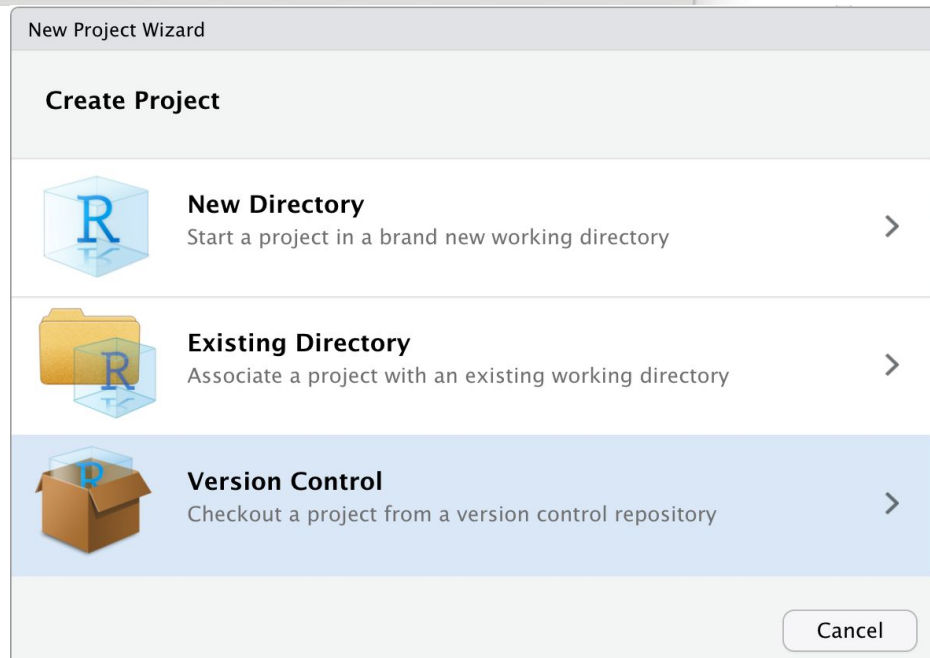
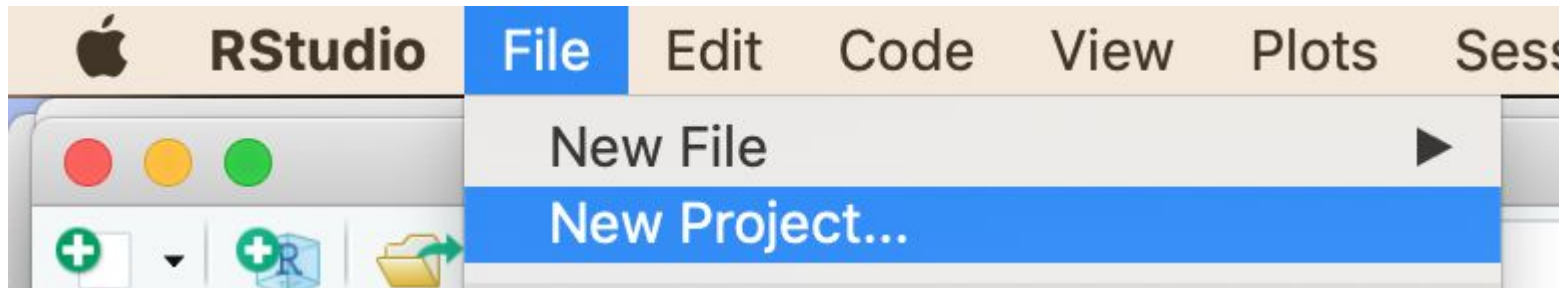


WEN_github /



Drag files here to add them to your repository

Or [choose your files](#)



New Project Wizard

Back

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



Cancel

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

Browse

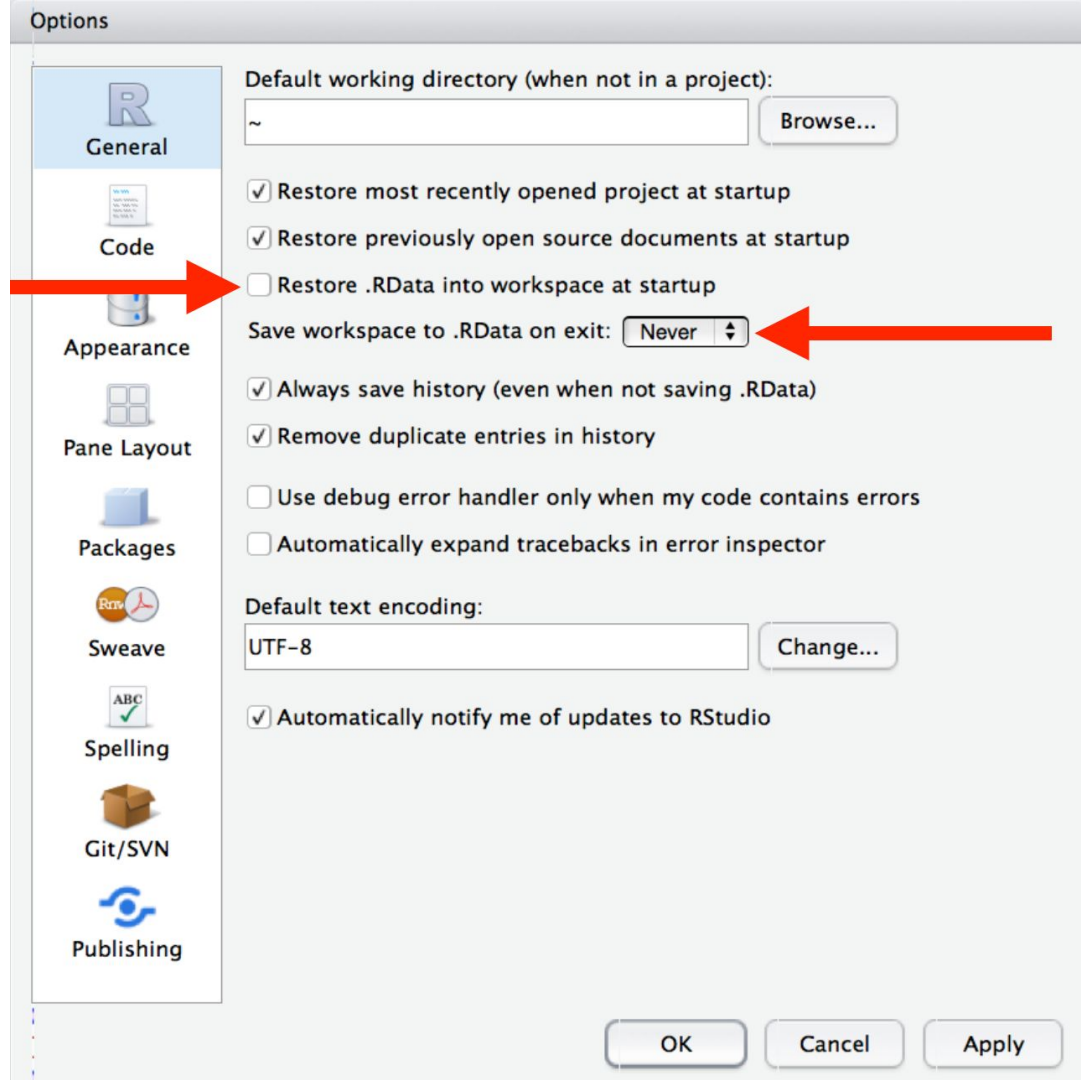
☐ Open in new session

Create Project

Cancel

| Files | Plots | Packages | Help | Viewer |
|--|-------|------------------|------|--------|
| + New Folder - Delete Rename More | | | | |
| OneDrive – McGill University > GitHub > WEN_github | | | | |
| Name | | | | Size |
| .. | | | | |
| <input type="checkbox"/> | | .gitignore | | 570 B |
| <input type="checkbox"/> | | Data | | |
| <input type="checkbox"/> | | LICENSE | | 1 KB |
| <input type="checkbox"/> | | Output | | |
| <input type="checkbox"/> | | Raw data | | |
| <input type="checkbox"/> | | README.md | | 2.7 KB |
| <input type="checkbox"/> | | Scripts | | |
| <input type="checkbox"/> | | WEN_github.Rproj | | 205 B |

Check/change your settings in R:



GO TO GITHUB

.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

Choosing the best license



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.

Ideal folder structure

Raw Data

Metadata includes date of download or collection, original source and re-use info

(Derived) Data

Data you transformed after downloading/collecting, e.g. merging 2 databases

Scripts

Code (can separate by language)

Output

Figures, tables, results

Every folder should contain a README!

Readme/Metadata best practices

- Include package version information and any external software used
- Describe files in a logical order
- Describe any column/variable names (especially units)

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 dashes

Clean coding

Be proactive

- Use `#####` to separate steps
- 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

More advanced functionality

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

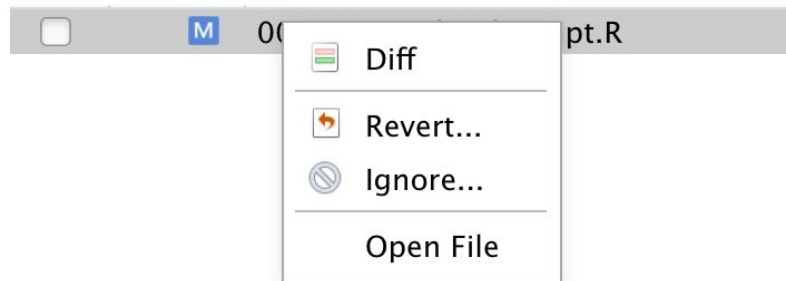
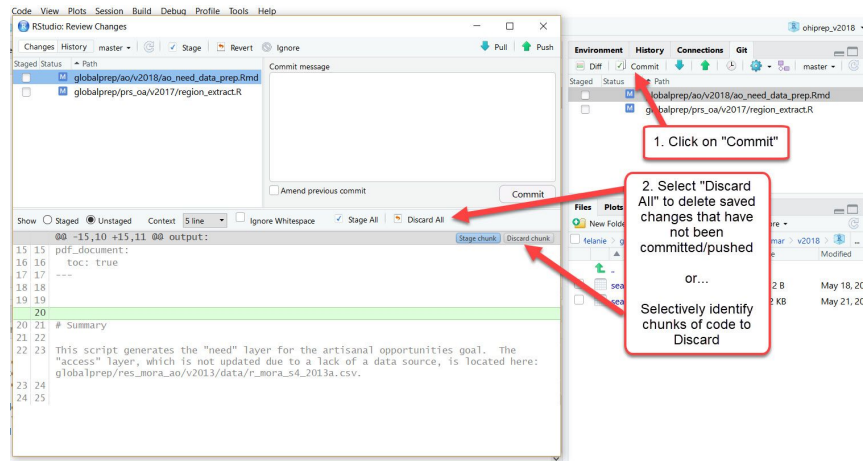
Pull often, commit after each change

Revert changes

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

Pre-commit:

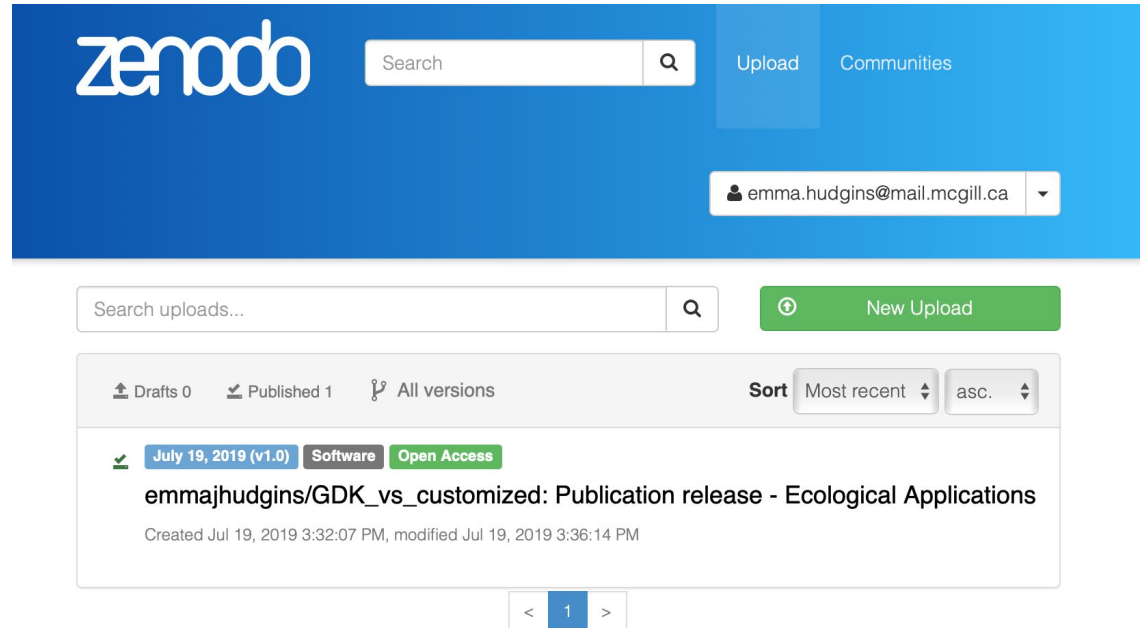
In RStudio, right click on a file and select 'revert'



Releases, Zenodo & DOI creation

Releases

No releases published
Create a new release



The screenshot displays the Zenodo website interface. At the top, the Zenodo logo is on the left, followed by a search bar and a magnifying glass icon. To the right are buttons for 'Upload' and 'Communities'. Below these is a user profile dropdown menu showing 'emma.hudgins@mail.mcgill.ca' with a downward arrow. Below the header is a section for 'Search uploads...' with a magnifying glass icon and a green 'New Upload' button. Below this is a navigation bar with links for 'Drafts 0', 'Published 1', and 'All versions', along with a 'Sort' dropdown menu set to 'Most recent' and 'asc.'. The main content area shows a list of uploads. The first entry is a green checkmark icon, followed by a blue badge 'July 19, 2019 (v1.0)', a grey badge 'Software', and a green badge 'Open Access'. The title of the upload is 'emmajhudgins/GDK_vs_customized: Publication release - Ecological Applications'. Below the title is the creation and modification timestamp: 'Created Jul 19, 2019 3:32:07 PM, modified Jul 19, 2019 3:36:14 PM'. At the bottom of the page is a pagination bar showing '< 1 >'.

zenodo

Search

Upload Communities

emma.hudgins@mail.mcgill.ca

Search uploads...

New Upload

Drafts 0 Published 1 All versions

Sort Most recent asc.

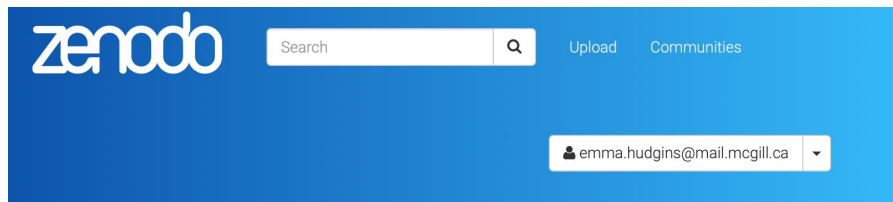
July 19, 2019 (v1.0) Software Open Access

emmajhudgins/GDK_vs_customized: Publication release - Ecological Applications

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

< 1 >

Releases, Zenodo & DOI creation



[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

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

OpenRefine



OpenRefine

A free, open source,
powerful tool for working
with messy data

<https://openrefine.org/>

Other helpful resources

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

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

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