

Reproducible data science with GitHub



Emma Hudgins
@emmajhudgins

Postdoctoral Fellow / Course Instructor
Biology Department, Carleton University

Objective: Develop confidence in using GitHub for version control with your R-based projects

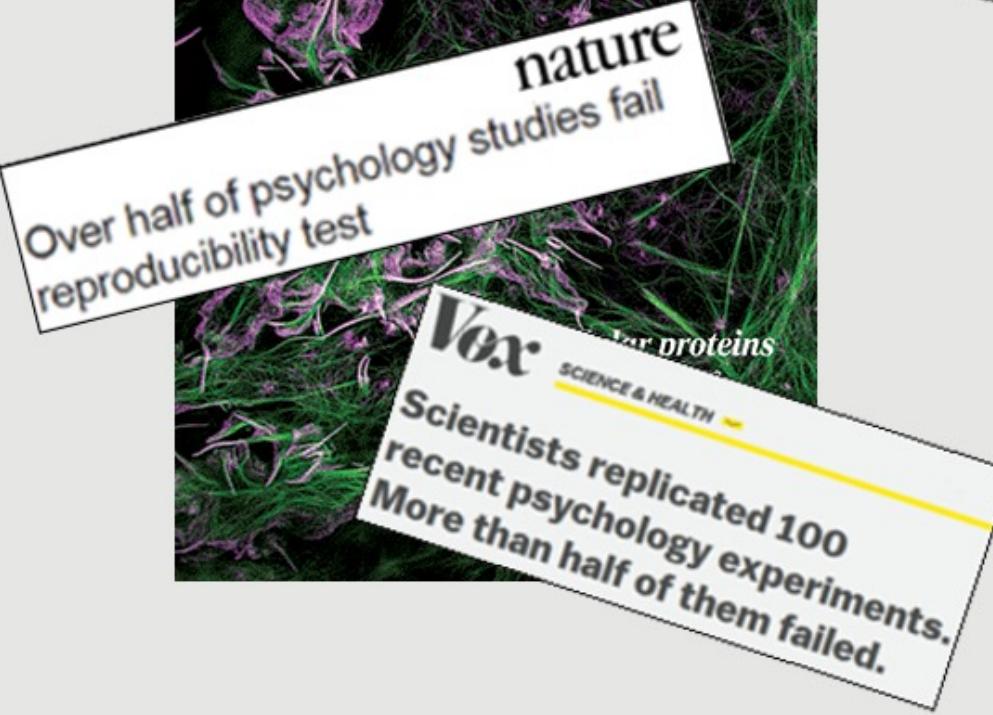
Plan for today

- Why do open, reproducible science?
- What are R projects, Git, and Github?
- Demo creating a repo, writing script, generating figures, and committing them to a repo
- You follow along with our demo and make your own!

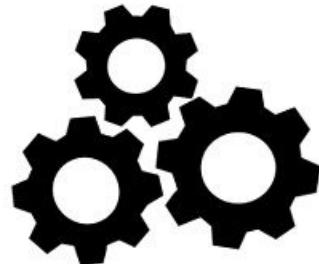
Why do open, reproducible science?

Have you heard of

- The replication crisis?
- FAIR data principles?



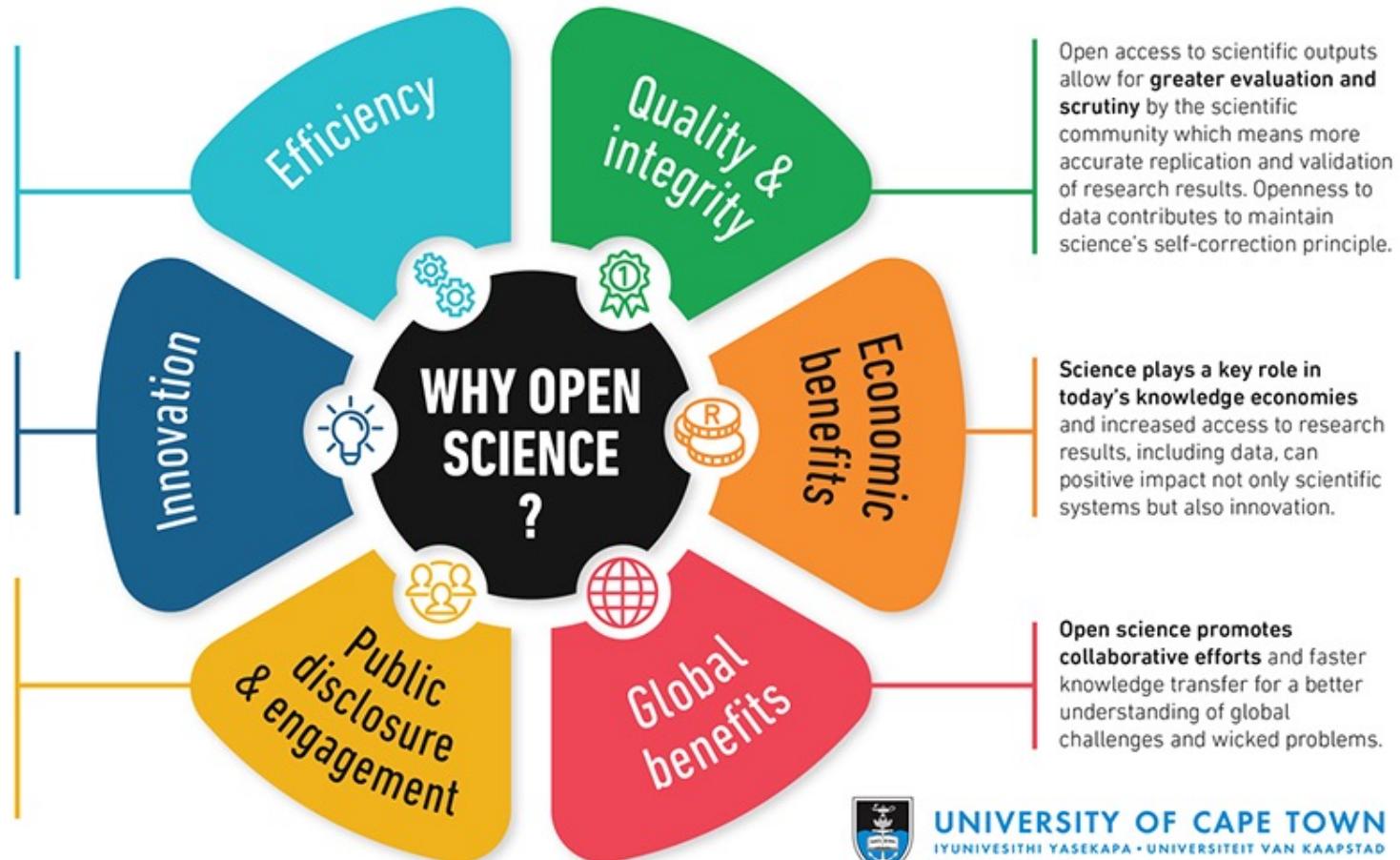
F indable A ccessible I nteroperable R eusable



Greater access to scientific inputs and outputs can increase scientific productivity through reducing duplication, allowing **more research from the same data** and multiplying opportunities for domestic and global participation in the research process.

Open science can **reduce delays in the re-use of scientific research** including articles and data, and promote a swifter path from research to innovation to produce new products and services.

Science, often publicly funded, should be publicly accessible to **promote a greater awareness** among citizens and to build public trust and support for public policies and investments in research. Open science also promotes citizen science in experiments and data collection.



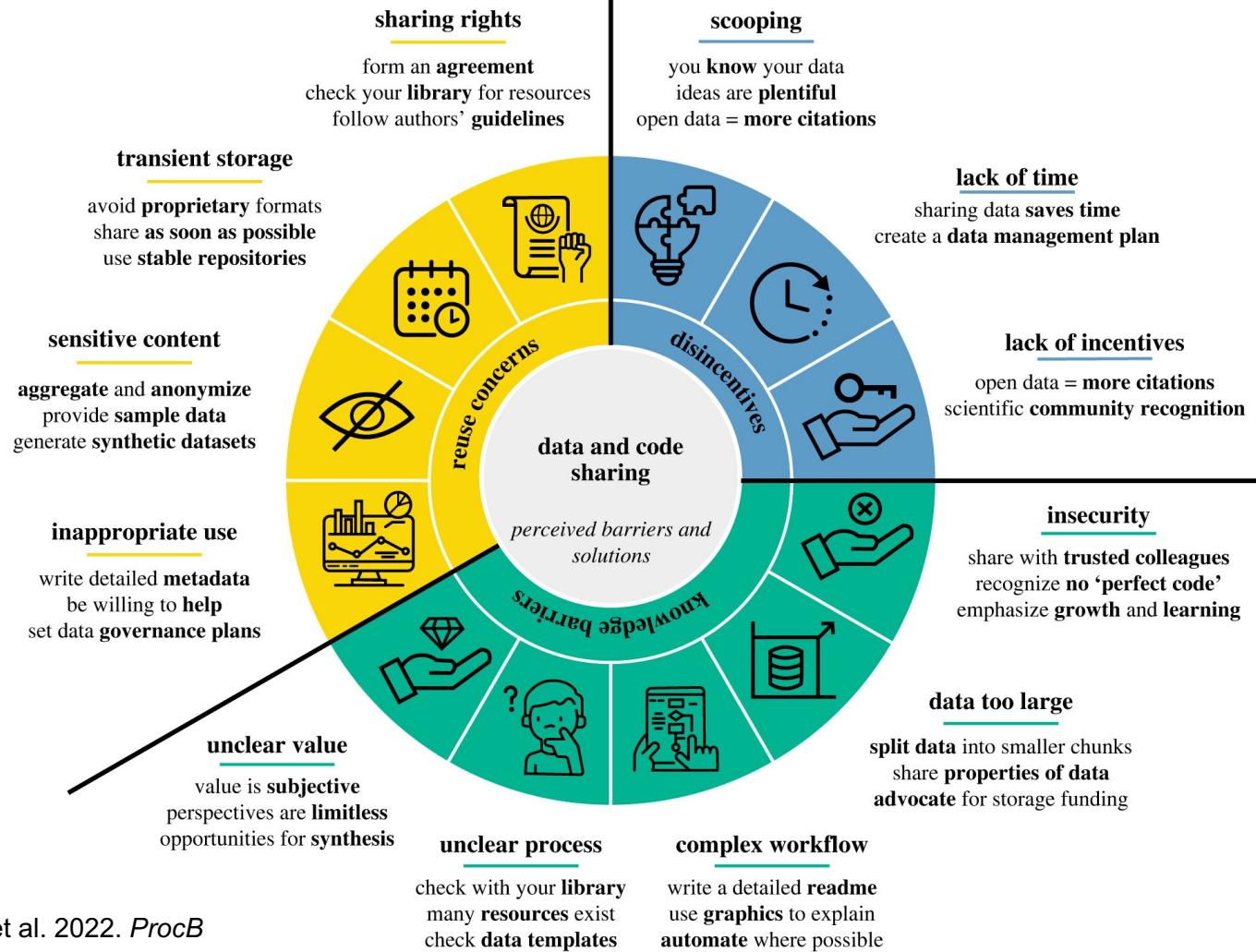
Open access to scientific outputs allow for **greater evaluation and scrutiny** by the scientific community which means more accurate replication and validation of research results. Openness to data contributes to maintain science's self-correction principle.

Science plays a key role in today's knowledge economies and increased access to research results, including data, can have a positive impact not only on scientific systems but also on innovation.

Open science promotes **collaborative efforts** and faster knowledge transfer for a better understanding of global challenges and wicked problems.



UNIVERSITY OF CAPE TOWN
IYUNIVESITI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD
GRAPHICS BY GAELEN PINNOCK

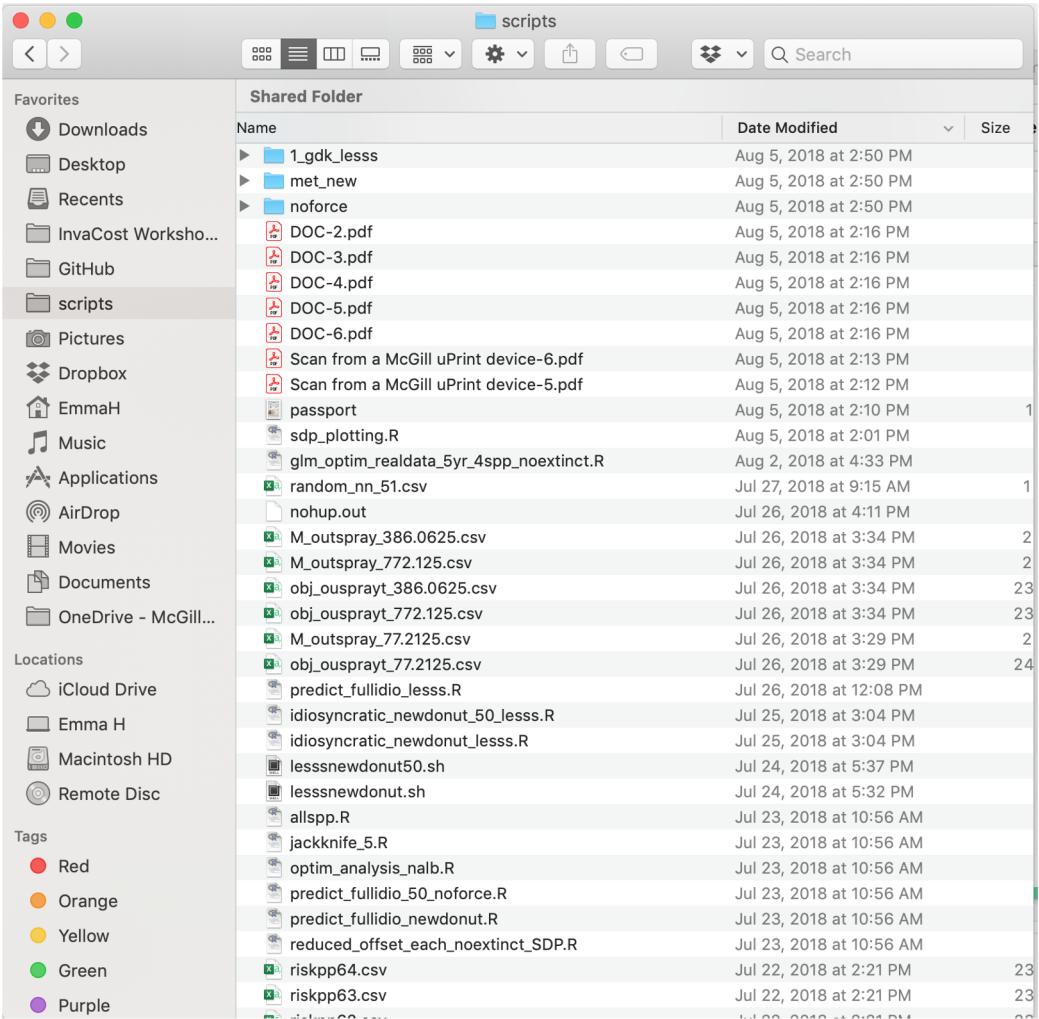




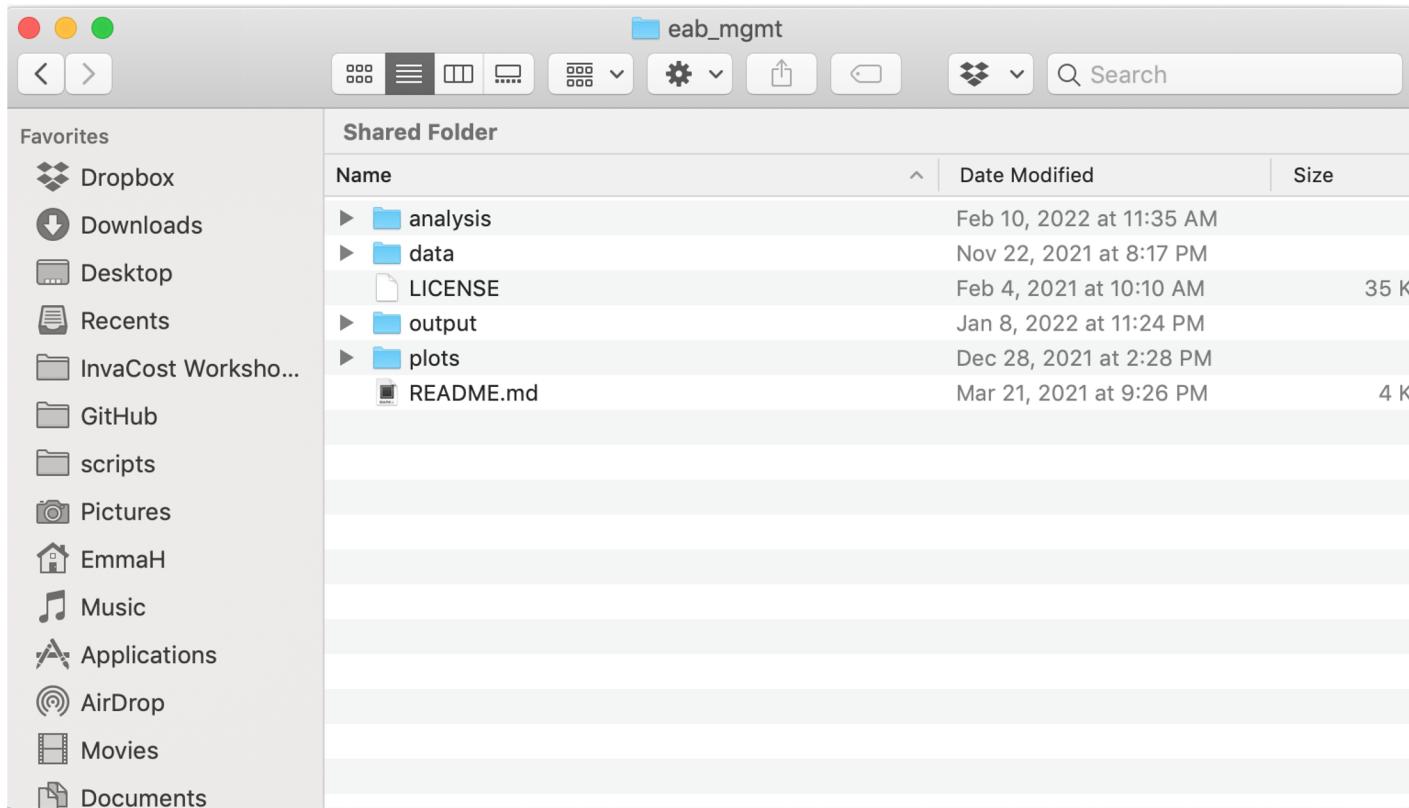
The power of projects, Git and GitHub

Your current organization

Could look something like this

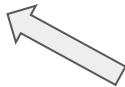


Your ideal organization





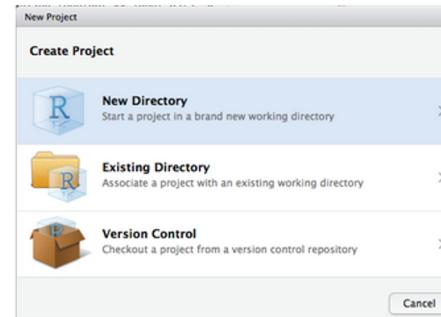
Push to GitHub



Commit often



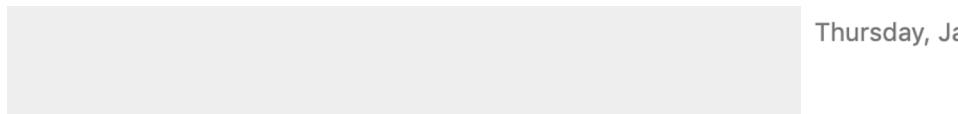
R project



R Projects

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

code and data



(i) 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 use them for that plot.

Thanks!

Name	Date Modified
SWPcomm	Today at 12:13 PM
RichnessAIC.R	Feb 14, 2022 at 1:21 PM
Mastersbiological.R	Feb 12, 2022 at 4:31 PM
TRCAFrog.csv	Feb 10, 2022 at 4:39 PM
TRCAFrog	Feb 8, 2022 at 6:00 PM
SWPFrog	Jan 28, 2022 at 3:00 PM
LRTRCAA3.png	Jan 17, 2022 at 4:00 PM
TRCALRA1.png	Jan 17, 2022 at 3:53 PM
LRSWPandscape.png	Jan 17, 2022 at 3:40 PM
LRSWPlocal.png	Jan 17, 2022 at 3:40 PM
LRSWPspecies.png	Jan 17, 2022 at 3:39 PM
TRCAxis3.png	Jan 13, 2022 at 6:19 PM
TRCAxis1.png	Jan 13, 2022 at 5:30 PM
Danny - PCA/NicheRover.R	Jan 7, 2022 at 4:31 PM
NicheRoverBoth.csv	Jan 7, 2022 at 4:12 PM
NicheRoverFrog.csv	Jan 7, 2022 at 12:41 PM
NicheRoverFrogT1.csv	Jan 6, 2022 at 10:12 AM
PCA_adj_score.csv	Jan 4, 2022 at 3:18 PM
PCA_adjs.csv	Jan 4, 2022 at 2:34 PM
SWPFish	Jan 4, 2022 at 12:10 PM
JulyBikingA	Dec 8, 2021 at 2:14 PM
Thermalprofiles.xlsx	Dec 1, 2021 at 10:49 AM
Thermalprofiles	Dec 1, 2021 at 10:21 AM
Thermalprofilesplot	Dec 1, 2021 at 9:57 AM
thermalprofiles.R	Nov 29, 2021 at 8:07 PM
OrdAIC.R	Nov 25, 2021 at 2:55 PM
Linear models.R	Nov 25, 2021 at 2:55 PM
Frogbar.csv	Nov 19, 2021 at 10:20 AM
PCA/NicheRover.R	Nov 17, 2021 at 3:02 PM
treebased models.R	Nov 4, 2021 at 10:47 AM
RCommander.R	Nov 4, 2021 at 10:35 AM
Cluster analysis.R	Nov 2, 2021 at 4:24 PM
Niche.rover.R	Oct 17, 2021 at 9:28 PM
FrogNiche2	Oct 15, 2021 at 6:03 PM
FrogNiche	Oct 15, 2021 at 5:22 PM
NMDStraheperformedvectors.jpg	Oct 13, 2021 at 10:11 AM

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

Git

VS.

GitHub



First developed
in 2005

Git is installed
and maintained
on your local
system (rather
than in the
cloud)



One thing that
really sets Git
apart is its
branching
model



Git is a high quality version control system



GitHub is
designed as a
Git repository
hosting service



You can share
your code with
others, giving
them the power
to make
revisions or edits



GitHub is a cloud-based hosting service

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)

GitHub basics

Your moves:

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

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

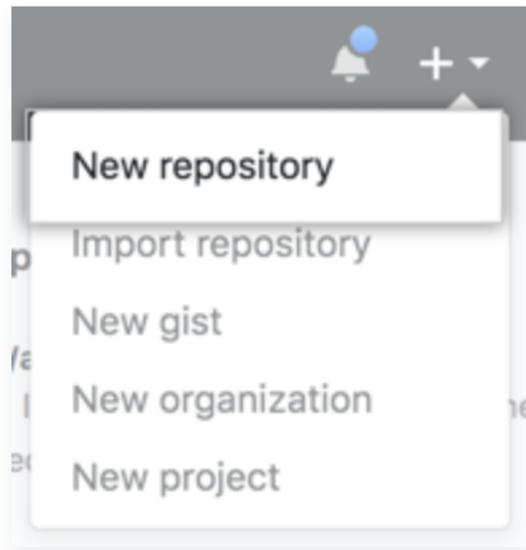
Pull - get changes from the cloud

Push - send changes to the cloud

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



- 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

 ✓

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.

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

[emmajhudgins / 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](#)

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:

- 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

 ⚡ 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](#)

LINK RSTUDIO AND GITHUB

Follow the steps here:

<https://gist.github.com/Z3tt/3dab3535007acf108391649766409421#file-github-r-L44>



RStudio

File

Edit

Code

View

Plots

Sess



New File



New Project...

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

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

 Back

Clone Git Repository



Repository URL:

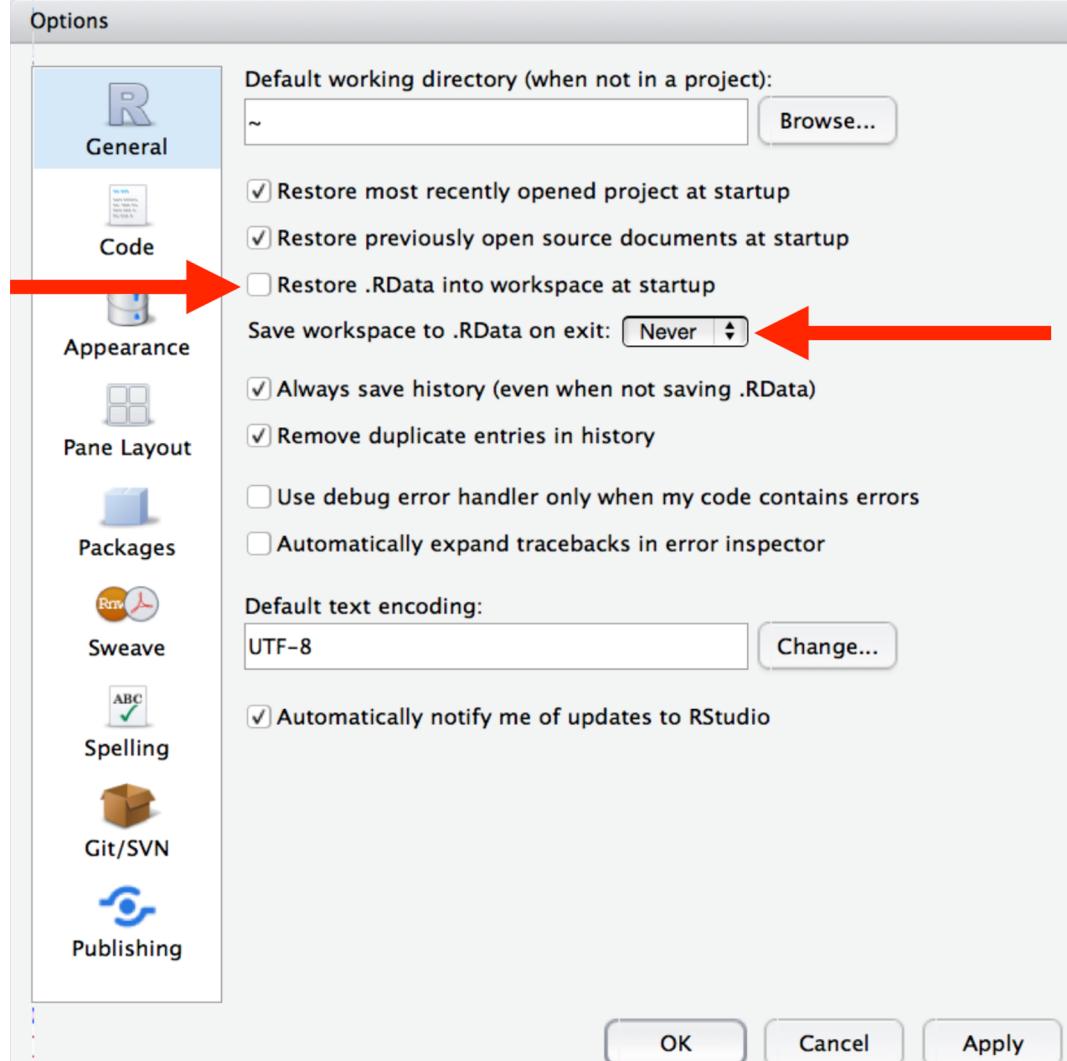
Project directory name:

Create project as subdirectory of:

  Open in new session Create Project Ca

Files		
Plots		
Packages		
Help		
Viewer		
 New Folder	 Delete	 Rename
 More		
OneDrive – McGill University > GitHub > WEN_github	R	...
Name		Size
 ..		
 .gitignore		570 B
 Data		
 LICENSE		1 KB
 Output		
 Raw data		
 README.md		2.7 KB
 Scripts		
 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

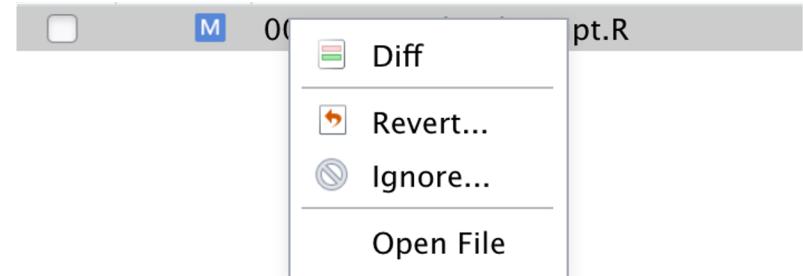
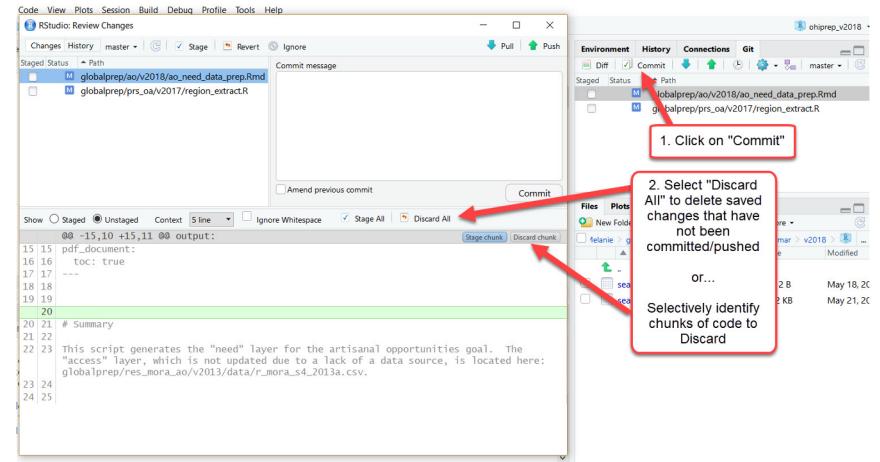
Pull often, commit after each change

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 shows the Zenodo website interface. At the top, there is a blue header bar with the Zenodo logo, a search bar, an 'Upload' button, and a 'Communities' link. A user profile is visible on the right. Below the header, a large green button says 'New Upload'. The main content area has a light gray background. It features a search bar at the top, followed by filters for 'Drafts 0', 'Published 1', and 'All versions'. A sorting section allows users to 'Sort' by 'Most recent' or 'asc.'. Below these, a single release is listed: 'July 19, 2019 (v1.0) Software Open Access emmajhudgins/GDK_vs_customized: Publication release - Ecological Applications'. The release was created on Jul 19, 2019, at 3:32:07 PM and modified on Jul 19, 2019, at 3:36:14 PM. At the bottom, there is a navigation bar with '<', '1', and '>' buttons.

Releases, Zenodo & DOI creation

The screenshot shows the Zenodo account settings interface. At the top, there's a navigation bar with the Zenodo logo, a search bar, and links for 'Upload' and 'Communities'. A user profile dropdown shows 'emma.hudgins@mail.mcgill.ca'. Below this, a breadcrumb navigation shows 'Home / Account / Linked accounts'. On the left, a sidebar menu includes 'Settings' (Profile, Change password, Security), 'Linked accounts' (selected), 'Applications', 'Shared links', and 'GitHub'. The main content area has two sections: 'Linked accounts' (describing single sign-on) and 'Repositories' (describing third-party access). Under 'Repositories', it lists a GitHub account ('emmaghudgins/Activity_sectors') and an ORCID account ('Connecting Research and Researchers').

Search

Upload Communities

emma.hudgins@mail.mcgill.ca

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/>