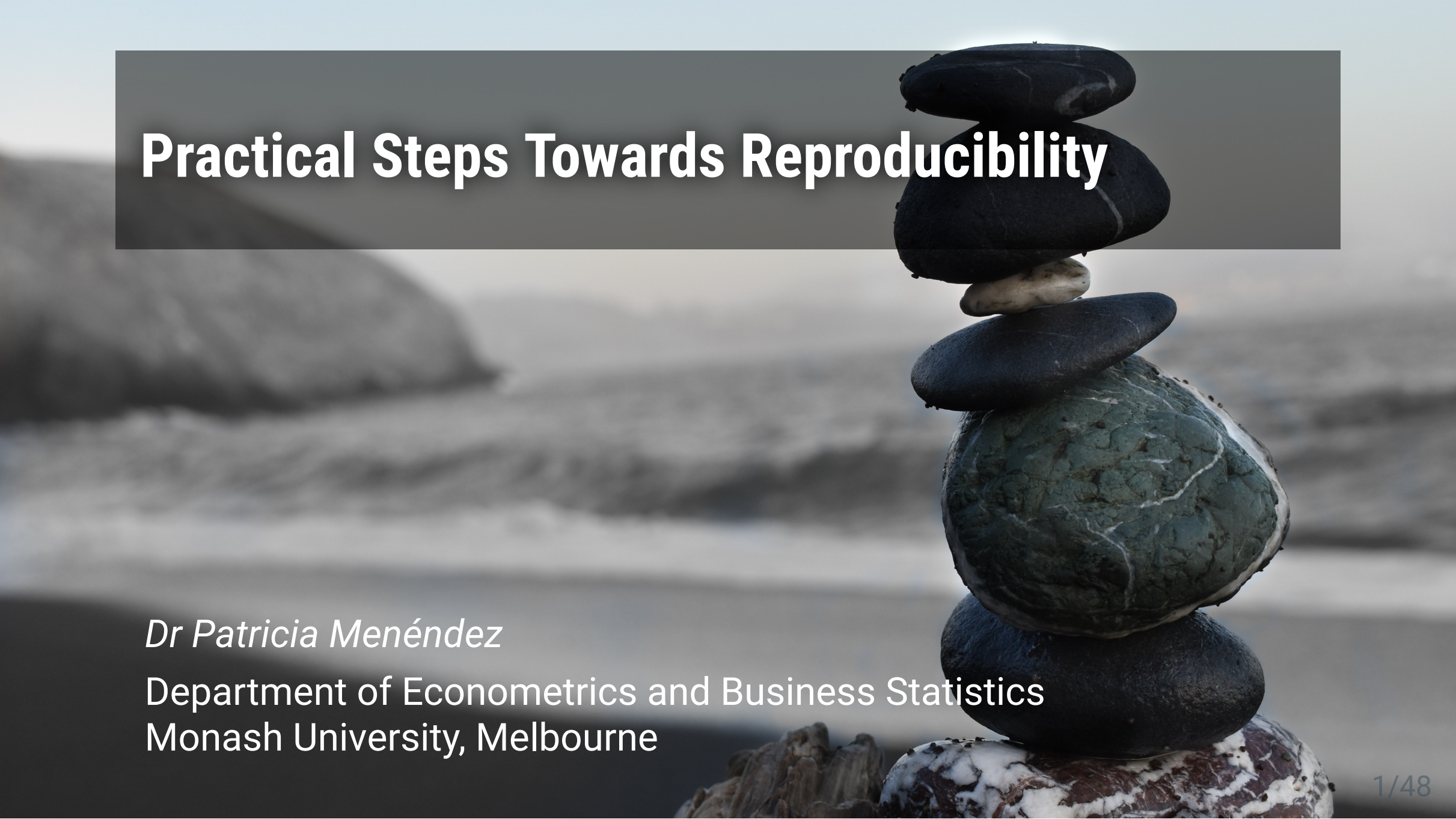
Practical Steps Towards Reproducibility



Dr Patricia Menéndez

Department of Econometrics and Business Statistics
Monash University, Melbourne

You are working on a project and it is going great!

You return to the same project a few months later

Photo by Matthew Guay on Unsplash

Critical issue

https://media.giphy.com/media/11fDMHAzihB8D6/giphy.gif

What could possible go wrong??!!

Which file was the last one?

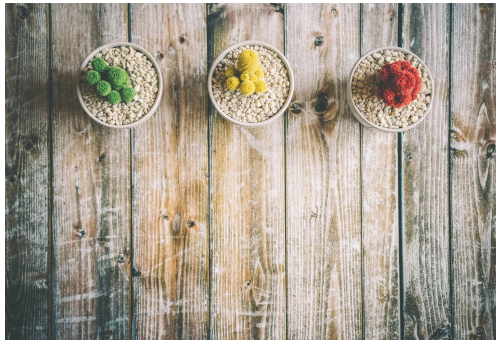
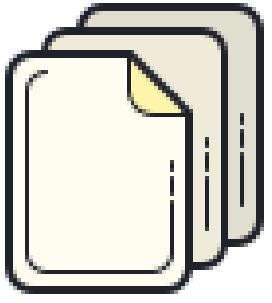


Photo by Scott Webb on Unsplash

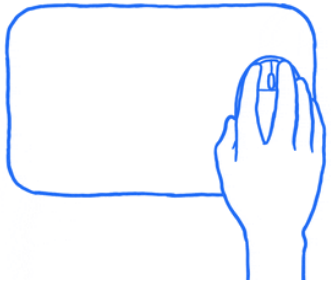
Did I modify my raw data?



Got a new computer?

Photo by Scott Graham on Unsplash

Copy & paste in the wrong place?



https://media.giphy.com/media/l0HlQXlQ3nHyLMvte/giphy.gif

Live our lives peacefully

Photo by Colton Sturgeon on Unsplash

Reproducible research and replicability

Definitions by the USA National Academies of Science, Engineering and Medicine:

- **Reproducibility** ("computational reproducibility") means obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis.
- **Replicability** means obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data.

Report on reproducibility and replicability

~~Figure it out~~

- Pieces
- Instructions/manual
- Tools

Photo by Xavi Cabrera on Unsplash

Photo by Karla Hernandez on Unsplash

Philosophy

Reproducibility is a way of thinking and approaching projects:

- Requires planning.
- Needs extra upfront effort.
- Demands us to be organized.
- Challenges us to think more broadly.

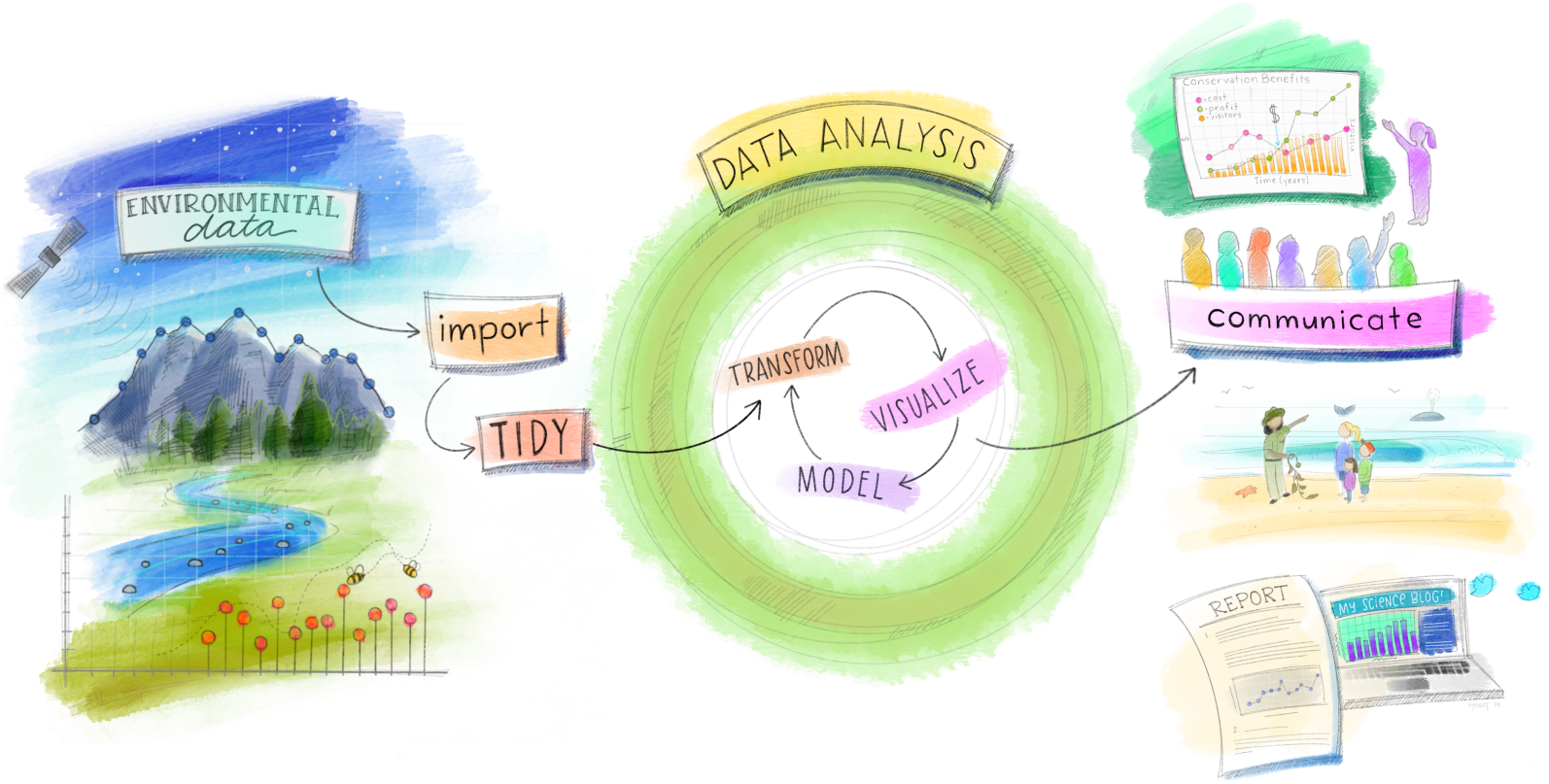
Photo by Diego PH on Unsplash

Reproducibility complexity

Complexity varies:

- Some projects require a single tool (R, Python, Matlab or Genstat for example) and involve only one person.
- Other projects might involve different teams and require many different tools.

Project example



Artwork by @allison_horst

Example: R, Rstudio and Rmarkdown files

first-rmarkdown.Rmd

Knit Insert

```
1  ---
2  title: "My first R Markdown"
3  output: html_document
4  ---
5
6  ## This is a second-level header
7
8  This is some text written in R Markdown.
9
10 I can use **markdown** syntax _here_.
11
12 ```{r}
13 # some R code
14 1 + 2
15 ```
16
```

My first R Markdown

This is a second-level header

This is some text written in R Markdown.

I can use **markdown** syntax *here*.

```
# some R code
1 + 2
```

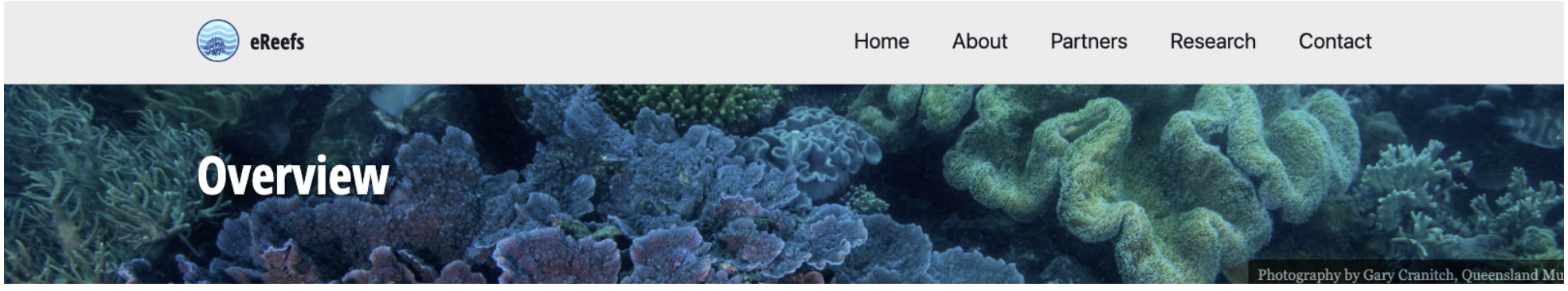
```
## [1] 3
```

Literate programming

Literate programming is an approach to writing reports using software that weaves together the source code and text at the time of creation.

Donald Knuth coined the term literate programming in the 1970s to refer to a source file that could be both run by a computer and “woven” with a formatted presentation document [Knuth, 1992].

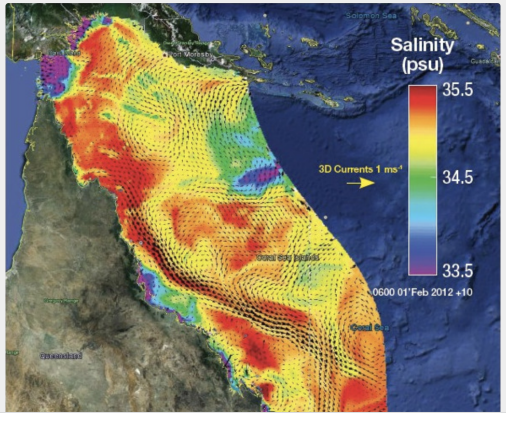
Complex workflow example



https://www.ereefs.org.au/about/

eReefs, which commenced in January 2012, is a six year $30 million collaborative project that combines government commitment to Reef protection, world-class science innovation and contributions from leading Australian businesses. Focused on the protection and preservation of the iconic Great Barrier Reef, it forms the first step in building comprehensive coastal information systems for Australia.

Using the latest technologies to collate data, and new and integrated modelling, eReefs will produce powerful visualisation, communication and reporting tools. It will provide for the Reef information akin to that provided by the Bureau of Meteorology for weather. This information will benefit government agencies, Reef managers, policy makers, researchers, industry and local communities.



Complex projects need more than literate programming

Photo by Umberto on Unsplash

Reproducibility set up will depend on the project

General practical tips for reproducible workflows

There is no one-size-fits-all approach!

1. Plan in adavance

- Plan the type and scale of the project for reproducibility (as much as possible!).
- Brainstorm about how different components are to be connected in a reproducible way. Update when necessary.
- Think about your "future self" and others.
- Set up a version control system.
- Plan for literate programming and scripting as much as possible.
- Keep revising and updating the plans as you go.

Workflow organization

Decide how you are going to organize your project workflow and prepare the scaffolding for that.

- For example, think about the project files structure
- In particular, where is the data going to be stored and how?
- Where will you place the scripts and codes?
- How are other things going to be organized in the project?
- How are the different elements going to be linked together?
- Where will you place the project documentation?

Think about someone else who have no clue about this project and will need to run it in 3 years time!

What is a version control system?

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

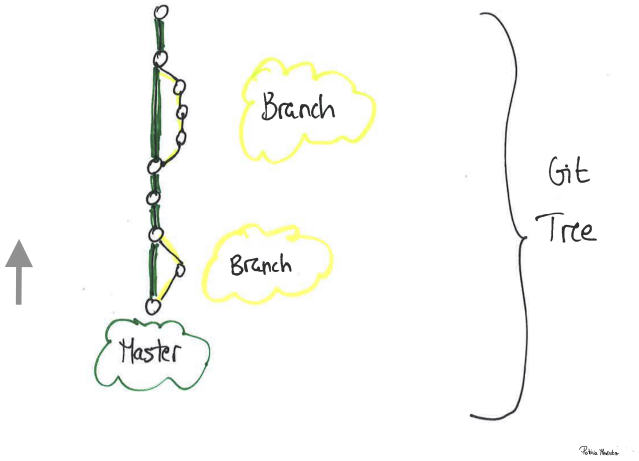
It also allows us to collaborate and share our projects with others!

Photo by Marco Lermer on Unsplash

Version control

- Version control systems are a category of software tools that help store and manage changes to source code (projects) over time.
- Version control software keeps track of every modification to the source code in a special database.
- If a mistake is made, you can turn back to previous versions and compare the code to fix the problem while minimizing disruption.
- It is easy to manage multiple versions of a project
- It is a very useful (actually essential!) tool for collaborating and for sharing open source resources.

Git



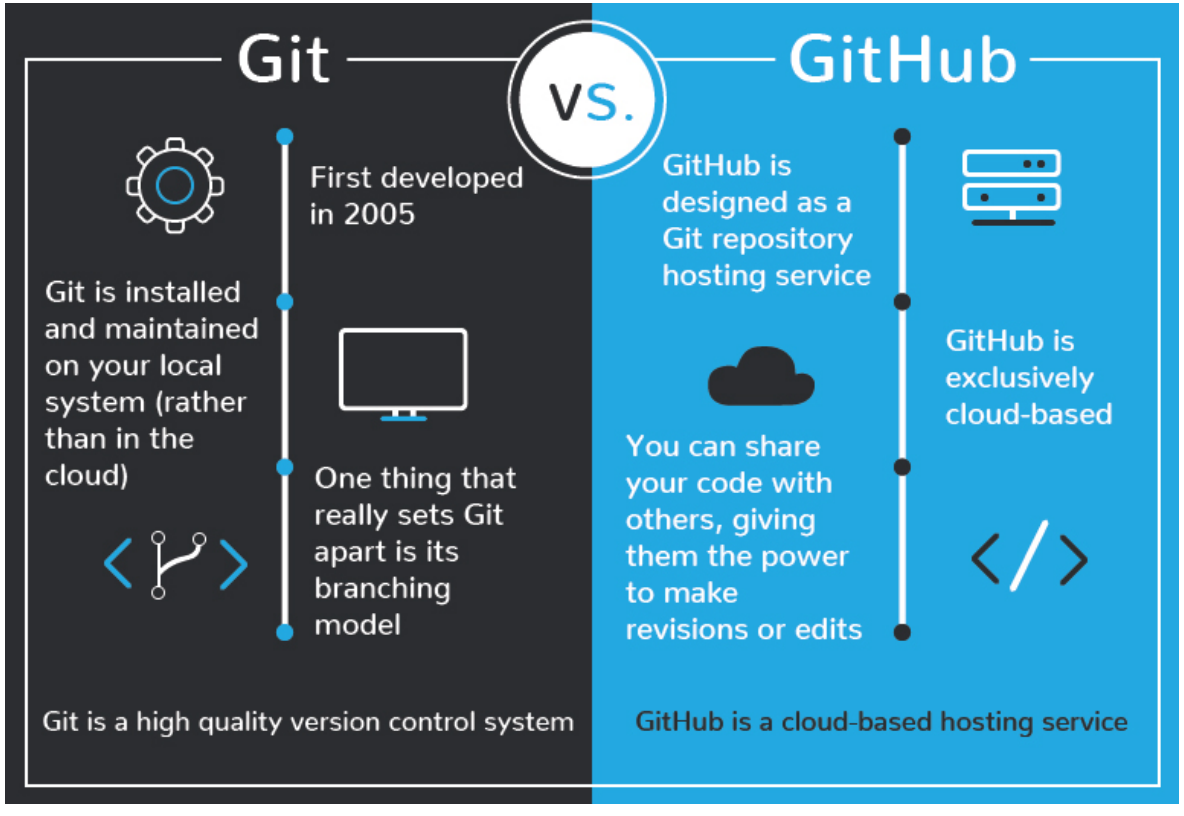
"Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows"

https://en.wikipedia.org/wiki/Git

GitHub, Bitbucket and others

- GitHub/Bitbucket are code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.
- Both are cloud-based hosting service that lets you manage Git repositories.

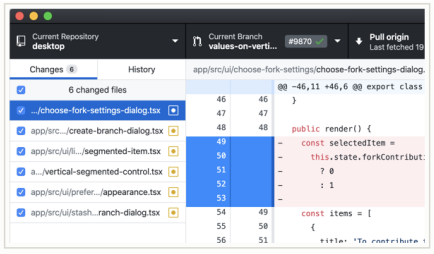
Git and GitHub



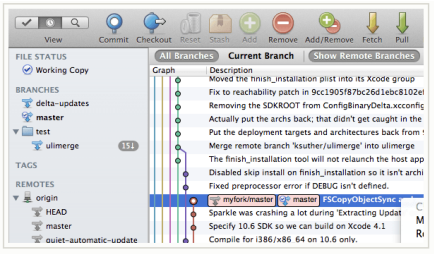
Source: https://blog.devmountain.com/git-vs-github-whats-the-difference/

Sharing: team work / open source

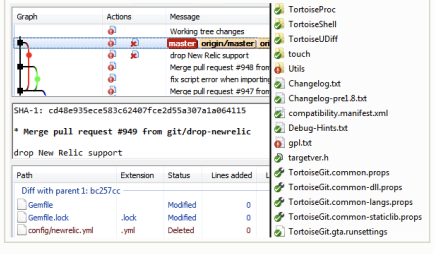
GUI clients



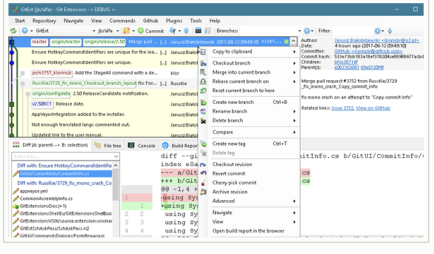
GitHub Desktop
Platforms: Mac, Windows
Price: Free
License: MIT



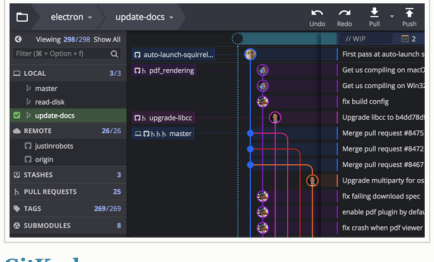
SourceTree
Platforms: Mac, Windows
Price: Free
License: Proprietary



TortoiseGit
Platforms: Windows
Price: Free
License: GNU GPL

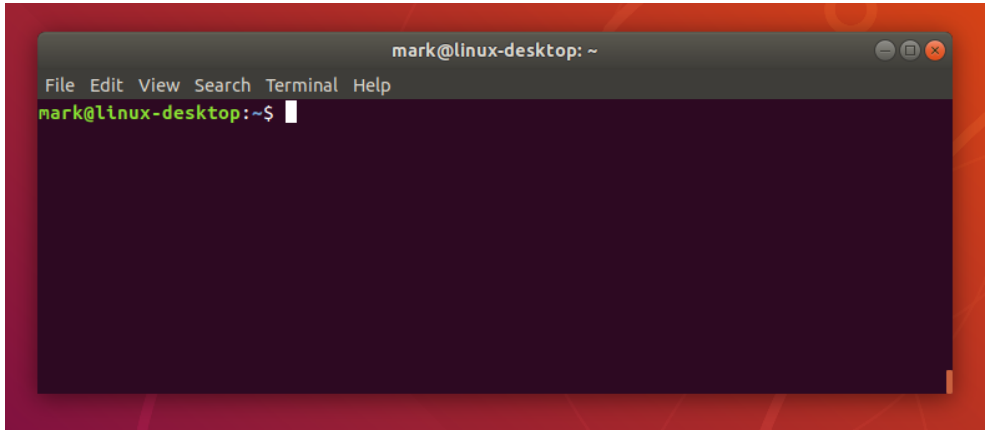


Git Extensions
Platforms: Linux, Mac, Windows
Price: Free
License: GNU GPL



GitKraken
Platforms: Linux, Mac, Windows
Price: Free / $59+/user annually
License: Proprietary

Magit
Platforms: Linux, Mac, Windows
Price: Free
License: GNU GPL



- Have a look at more GUI clients here.

- A great Git open source book here.

2. File system for the project

- Organize project files for reproducibility.
- Think carefully about files accessibility.
- Consider the different types of files in the project.
- Prepare for storage requirements.
- Think about the connectivity between the different components.

Computer paths

Where are files and folders stored in our computer?

Photo by Nathan Anderson on Unsplash

Computer paths

- A *path* is the complete location or name of where a computer file, directory, device, or web page is located.

- Windows \(\rightarrow\) C:\documents\example
- Mac/linux \(\rightarrow\) /Users/documents/example

Absolute and relative paths

- **An absolute or full path** contains the computer root's directory and all other sub directories where a file is located.

  "C:\documents\example\file.txt"

- **Relative path** refers to a location that is relative to a current directory (or folder).

  "example\file.txt"

A reproducible project should not have absolute paths!

Neat file system

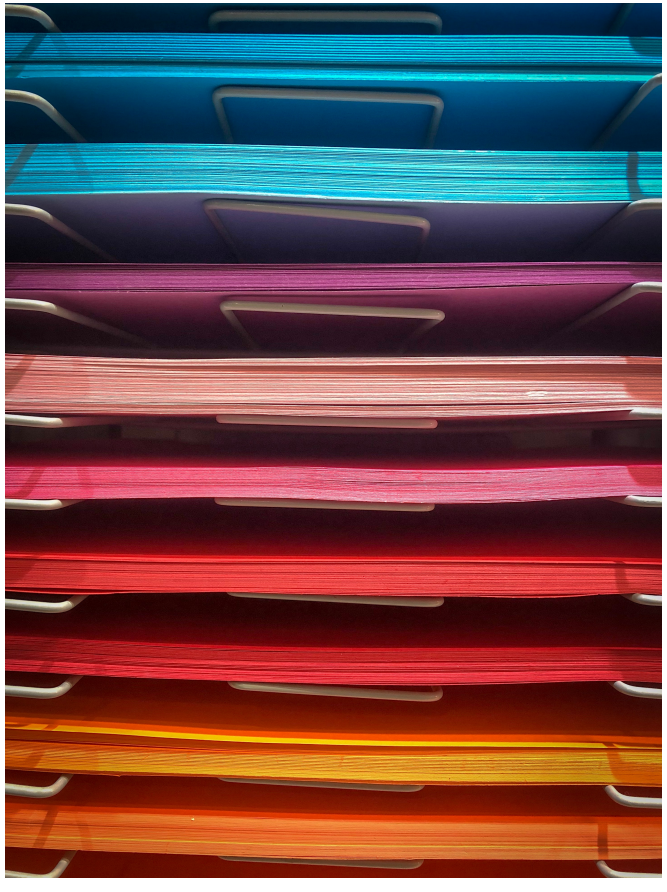
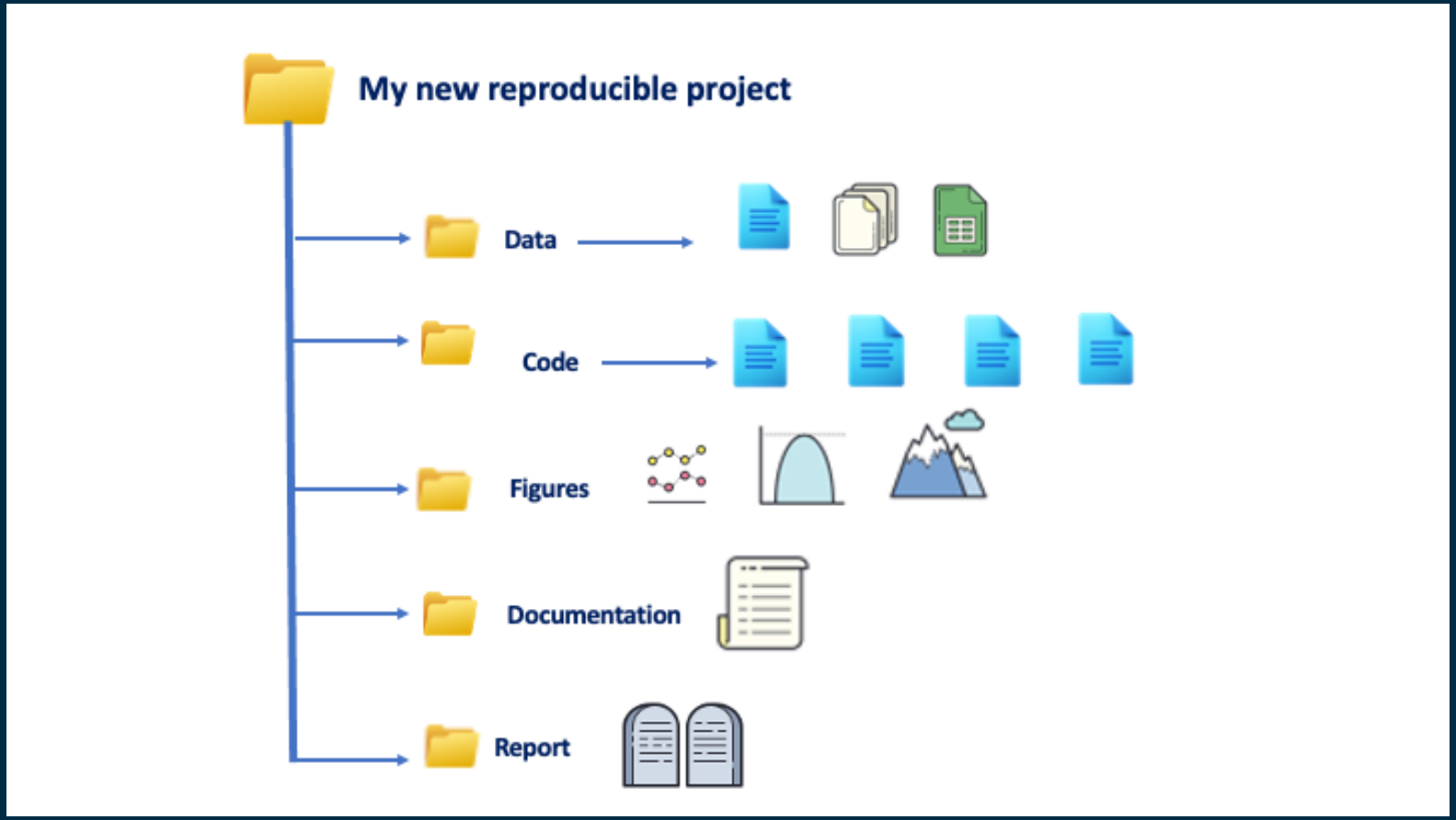


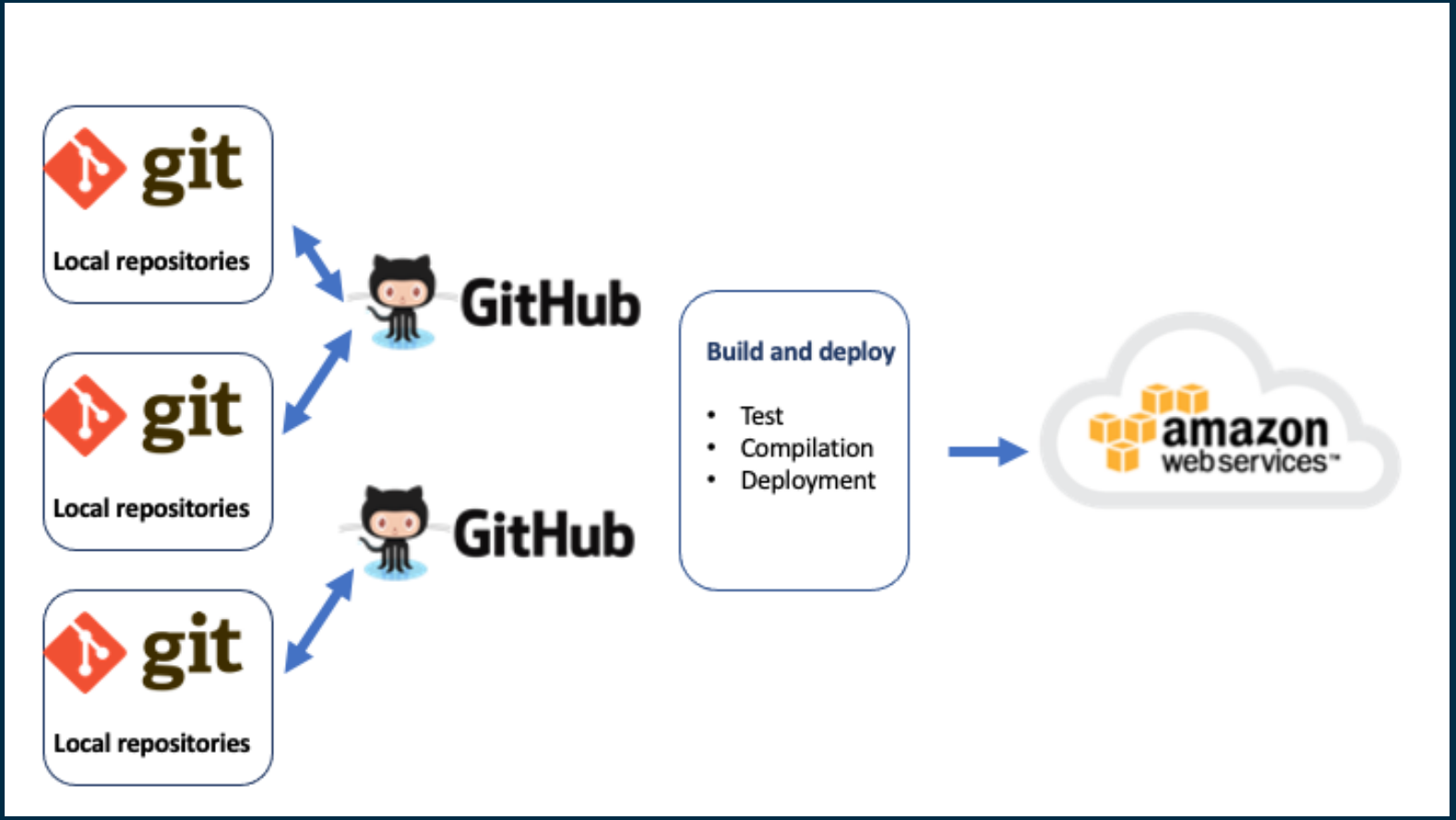
Photo by Omid Kashmari on Unsplash

Photo by Alexander Grey on Unsplash

Project organization: Examples



Project organization: Examples



3. Accessible connected workflow

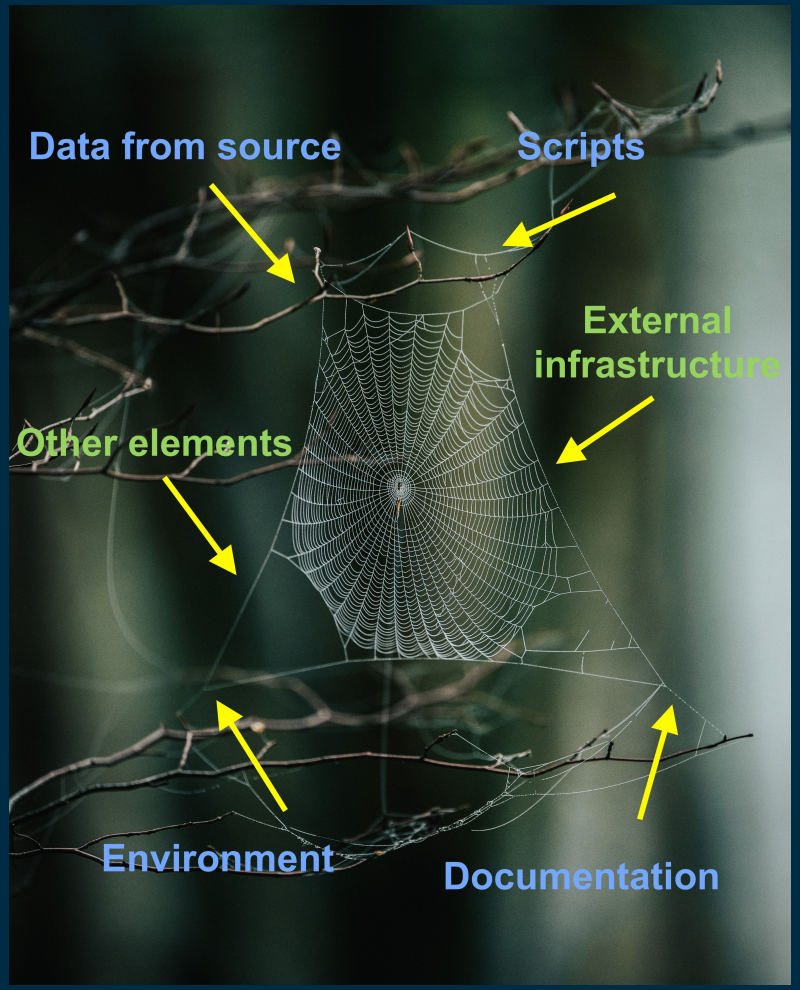


Photo by Tamas Tuzes-Katai on Unsplash

Data handling

- Record the **provenance** and licensing of the data.
- Use raw data and keep data files as **read-only** whenever possible.
- Integrate all the data preparation and data wrangling in the project to ensure reproducibility:

- In some projects data wrangling might be done using different software.
  - When different software is used (SQL/SAS/Excel/Python) all the steps and how they connect must be documented.

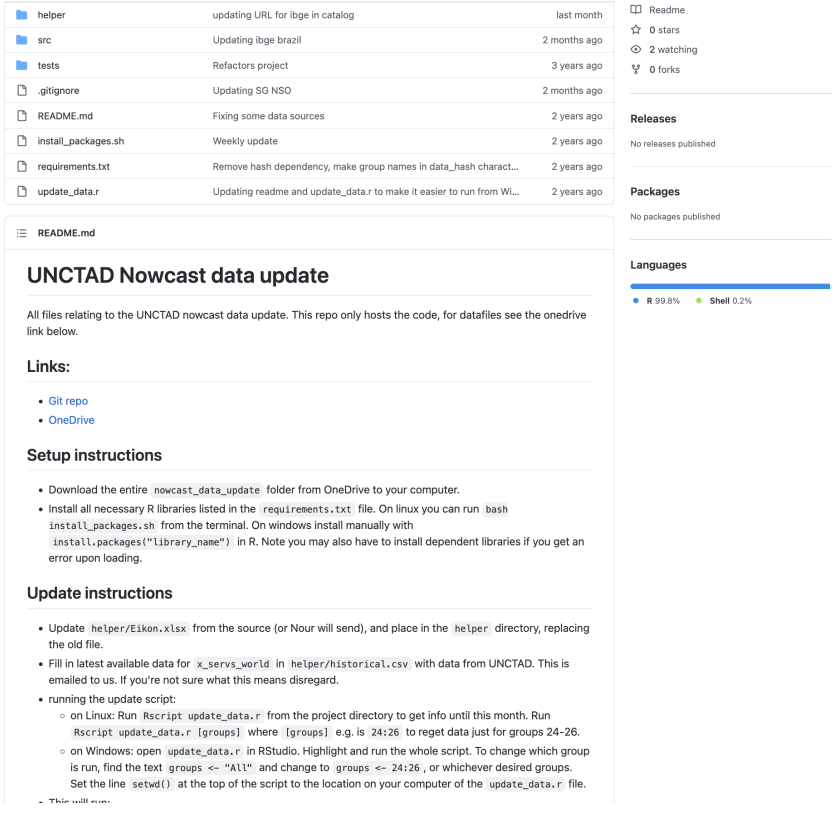
4. Documentation

- Document for reproducibility from day 0.
- Describe all the elements of the project.
- Keep a journal for manual steps.
- Be specific about documenting how different components are linked together to ensure reproducibility.
- Data provenance, licensing, scripts, any tools used in the project.
- Clean up when necessary!

Think about your future self and others that might need to use the project.

README example

README example from UNCTAD: **United Nations Conference on Trade and Development**.



helper	updating URL for ibge in catalog	last month
src	Updating ibge brazil	2 months ago
tests	Refactors project	3 years ago
.gitignore	Updating SG NSO	2 months ago
README.md	Fixing some data sources	2 years ago
install_packages.sh	Weekly update	2 years ago
requirements.txt	Remove hash dependency, make group names in data_hash charact...	2 years ago
update_data.r	Updating readme and update_data.r to make it easier to run from Wi...	2 years ago

README.md

UNCTAD Nowcast data update

All files relating to the UNCTAD nowcast data update. This repo only hosts the code, for datafiles see the onedrive link below.

Links:

- Git repo
- OneDrive

Setup instructions

- Download the entire `nowcast_data_update` folder from OneDrive to your computer.
- Install all necessary R libraries listed in the `requirements.txt` file. On linux you can run `bash install_packages.sh` from the terminal. On windows install manually with `install.packages("library_name")` in R. Note you may also have to install dependent libraries if you get an error upon loading.

Update instructions

- Update `helper/Eikon.xlsx` from the source (or Nour will send), and place in the `helper` directory, replacing the old file.
- Fill in latest available data for `x_servs_world` in `helper/historical.csv` with data from UNCTAD. This is emailed to us. If you're not sure what this means disregard.
- running the update script:
  - on Linux: Run `Rscript update_data.r` from the project directory to get info until this month. Run `Rscript update_data.r [groups]` where `[groups]` e.g. is `24:26` to reget data just for groups 24-26.
  - on Windows: open `update_data.r` in RStudio. Highlight and run the whole script. To change which group is run, find the text `groups <- "All"` and change to `groups <- 24:26`, or whichever desired groups. Set the line `setwd()` at the top of the script to the location on your computer of the `update_data.r` file.

Readme
0 stars
2 watching
0 forks

Releases

No releases published

Packages

No packages published

Languages

R 99.8% Shell 0.2%

https://github.com/dhopp1-UNCTAD/nowcast_data_update

README complex example

AIMS eReefs Visualisations and aggregations platform documentation repository.

README.md

Background

eReefs has many components developed and maintained by each of the organisations in the collaboration, including catchment modelling (Queensland Government), remote sensing (BOM and CSIRO), hydrodynamic modelling (BOM and CSIRO) and biogeochemical modelling (CSIRO). AIMS's contribution is to provide data aggregation and visualisation services. Figure 1 shows an overview of the major components of eReefs. The AIMS eReefs platform generates its products based on eReefs model data provided by CSIRO, which are driven from boundary data from BOM and DES (Queensland Government). Both CSIRO and BOM provide some additional visualisation websites. Each of the eReefs website are accessible from the eReefs website. This repository focuses on the AIMS eReefs Platform which provides visualisations of the eReefs models as animated multi-panel map videos that show the relationships between variables in the models.

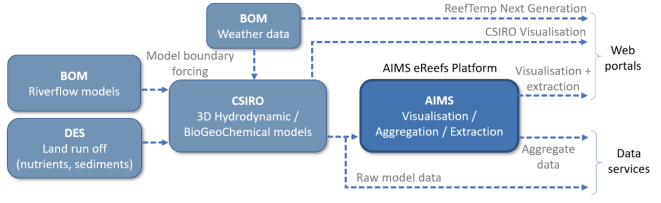


Figure 1. Overview of the data workflow in eReefs.

https://github.com/open-AIMS/ereefs-documentation

5. Code Environment Container

Have you ever encountered the following issue?

- My code run 6 months ago (I am sure!) and now it is not working.
- My figures look funny ...
- Mostly it is all about your program versions!!

Photo by Markus Spiske on Unsplash

Light weight dependency management

There are a couple of R dependency management solutions available: packrat and renv packages offer capabilities for dependency management.

The idea is to create a project-local-library to ensure that projects get their own unique library of R packages!

renv might not be enough

renv 0.16.0 Get started Reference Articles Changelog Search for

Using renv with Docker

Source: `vignettes/docker.Rmd`

On this page

- Creating Docker Images with renv
- Dynamically Provisioning R Libraries with renv
- Handling the renv Autoloader

While `renv` can help capture the state of your R library at some point in time, there are still other aspects of the system that can influence the run-time behavior of your R application. In particular, the same R code can produce different results depending on:

- The operating system in use,
- The compiler flags used when R and packages are built,
- The LAPACK / BLAS system(s) in use,
- The versions of system libraries installed and in use,

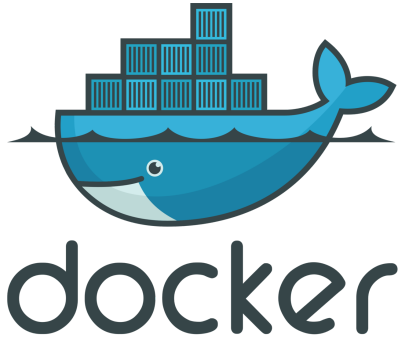
And so on. Docker is a tool that helps solve this problem through the use of **containers**. Very

https://rstudio.github.io/renv/articles/docker.html

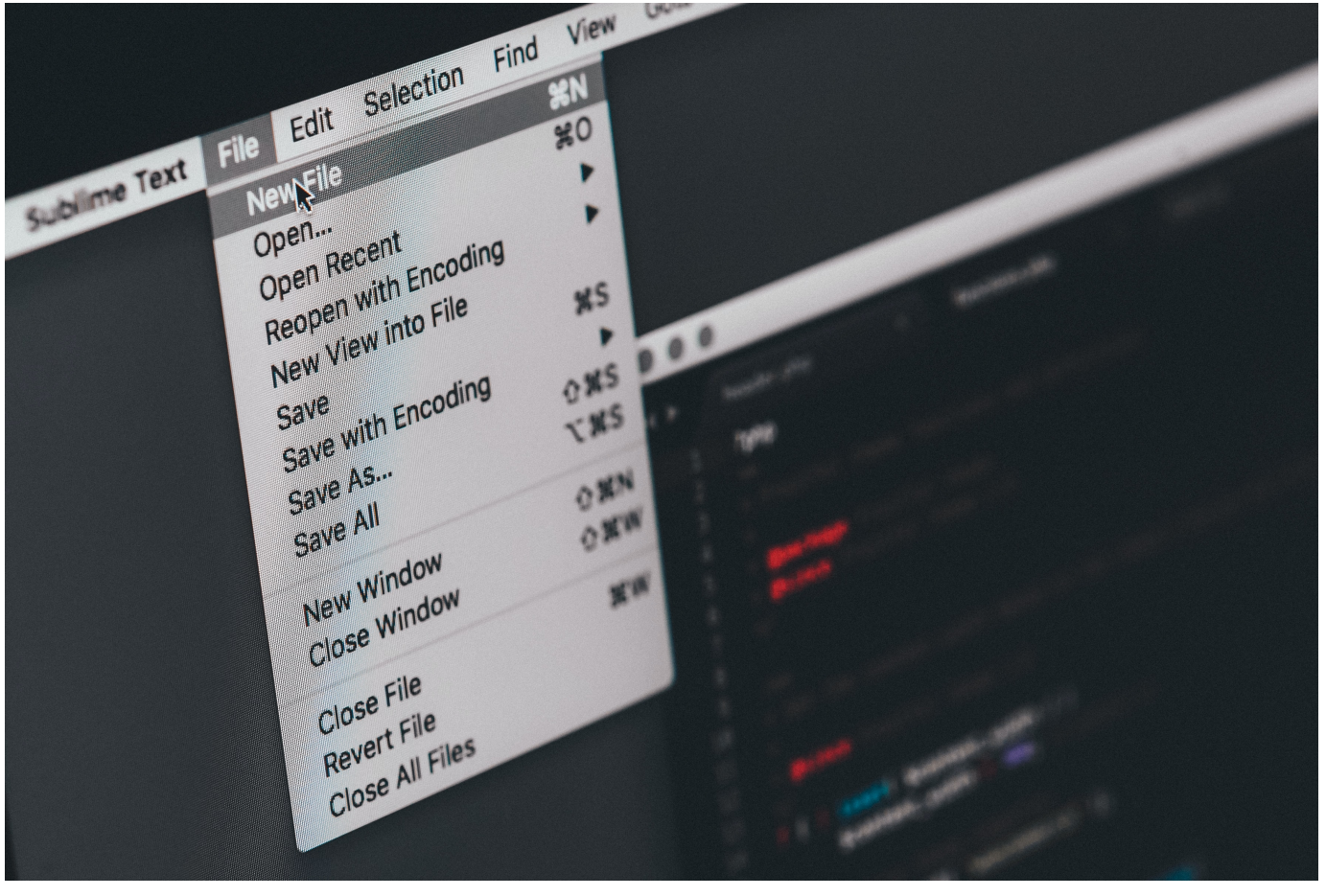
Docker

"A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers – images become containers when they run on Docker Engine." Instructions for container environment.

https://www.docker.com/resources/what-container/

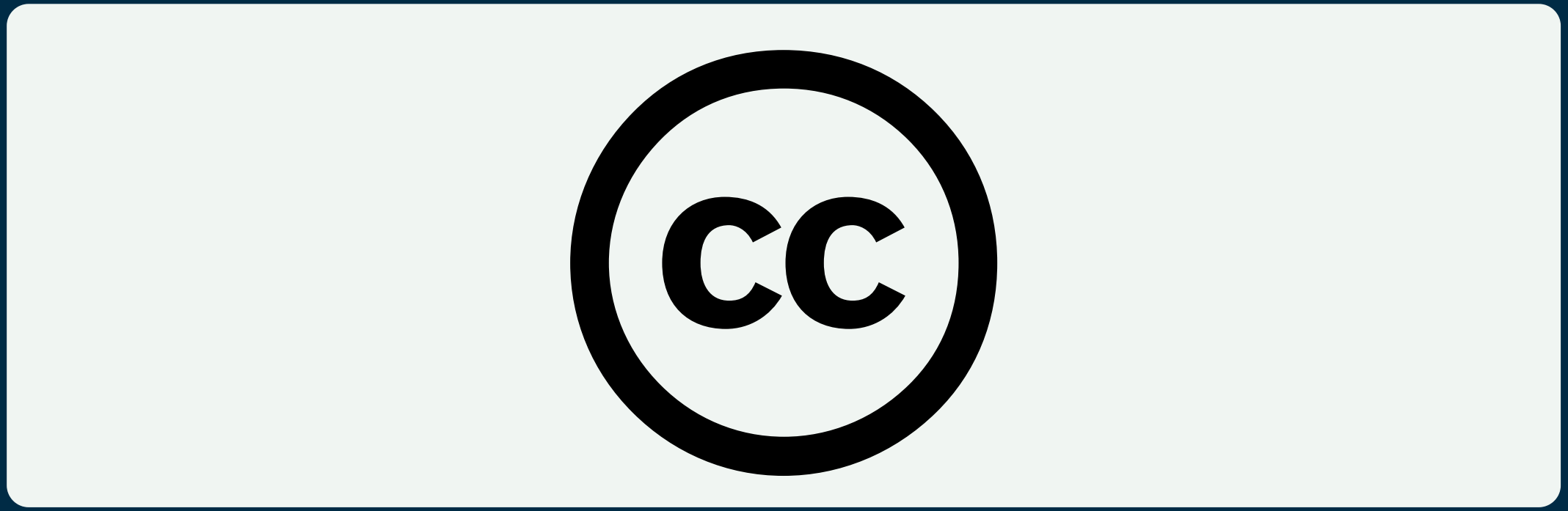


Record manually



Written instructions

6. Add a license



By Creative Commons, fixed by Quibik - Official Creative Commons' base, Public Domain, https://commons.wikimedia.org/w/index.php?curid=1484324

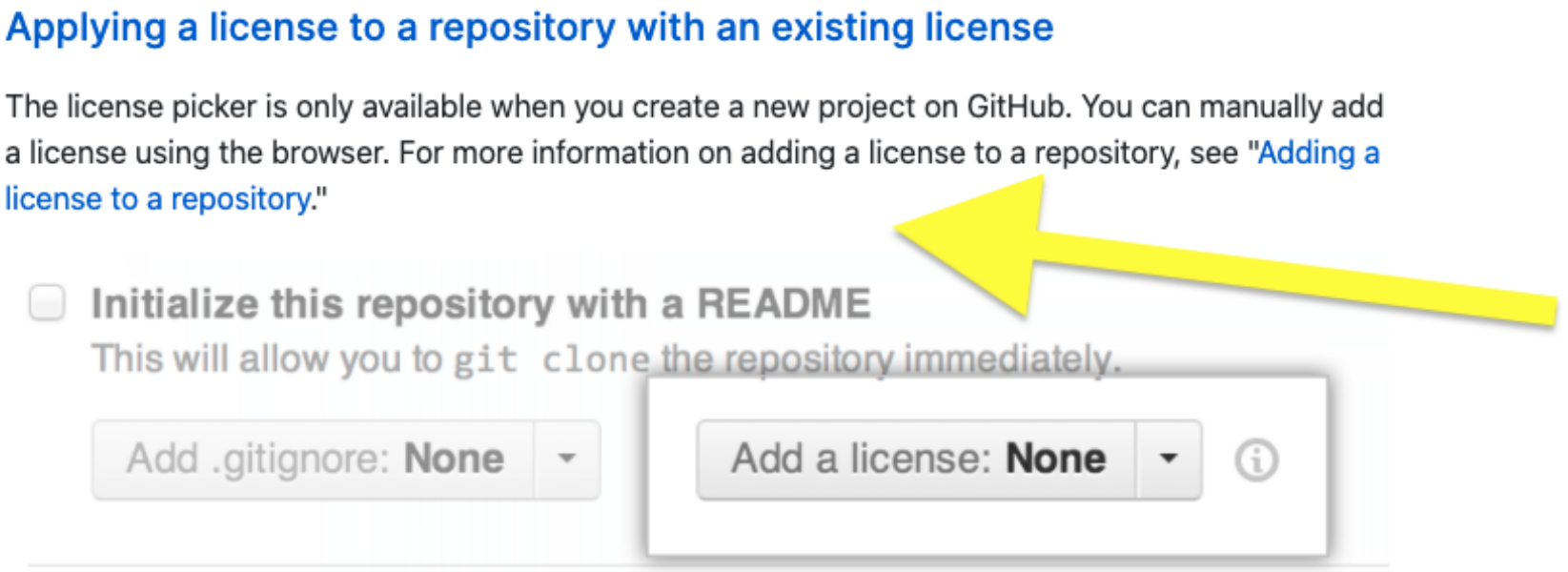
Licensing an open source repository

Public repos in GitHub make your work publicly available and therefore it is important to establish how your work should be acknowledged if someone else wants to use it.

> "Public repositories on GitHub are often used to share open source software. For your repository to truly be open source, you'll need to license it so that others are free to use, change, and distribute the software."

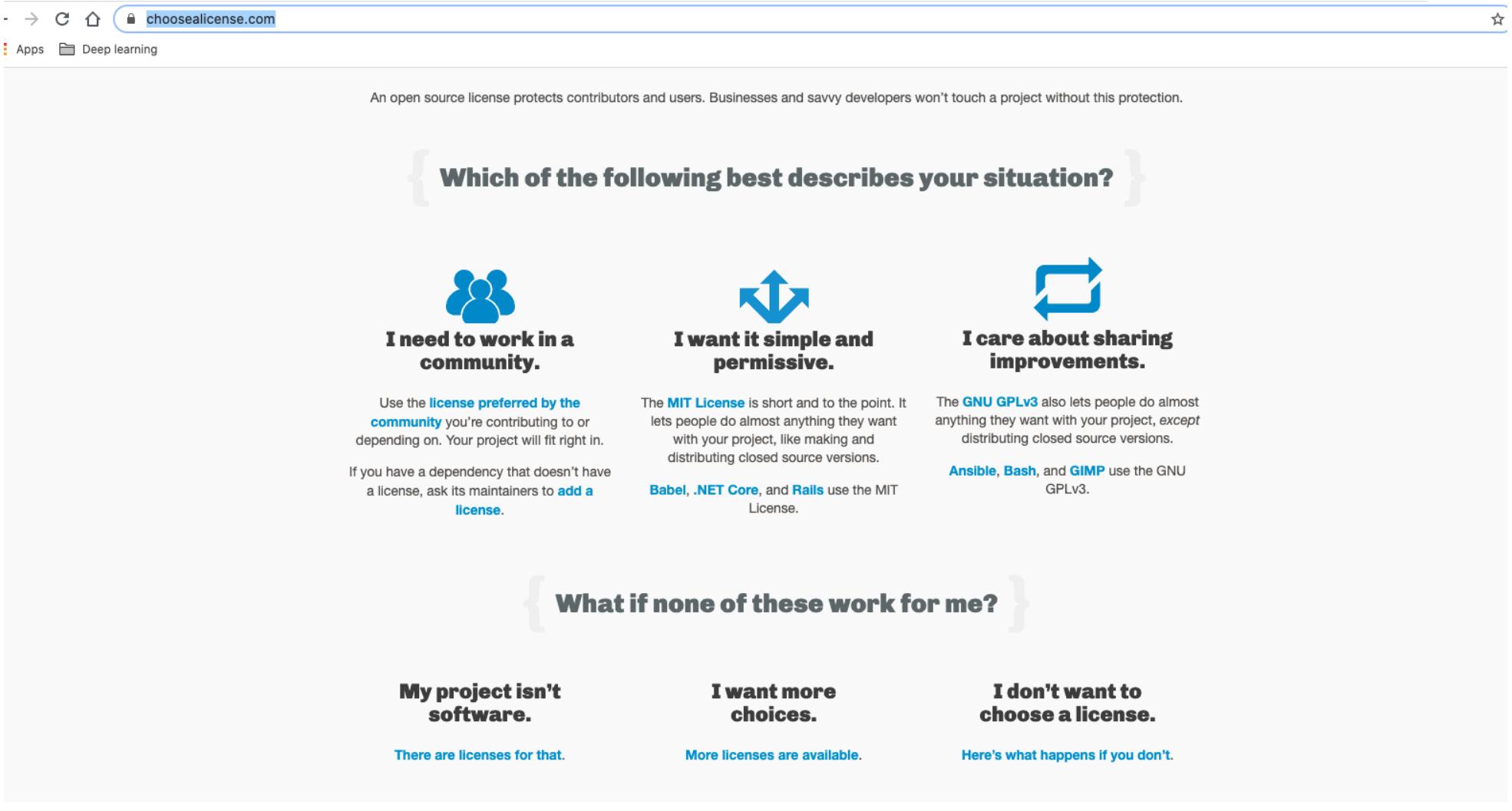
More info here

Available licenses in GitHub



More info here

Choose an open source license



[Source here]

No license

Home

No License

When you make a creative work (which includes code), the work is under exclusive copyright by default. Unless you include a license that specifies otherwise, nobody else can copy, distribute, or modify your work without being at risk of take-downs, shake-downs, or litigation. Once the work has other contributors (each a copyright holder), "nobody" starts including you.

Even in the absence of a license file, you may grant some rights in cases where you publish your source code to a site that requires accepting terms of service. For example, if you publish your source code in a public repository on GitHub, you have accepted the **Terms of Service**, by which you allow others to view and fork your repository. Others may not need your permission if **limitations and exceptions to copyright** apply to their particular situation. Neither site terms nor jurisdiction-specific copyright limitations are sufficient for the kinds of collaboration that people usually seek on a public code host, such as experimentation, modification, and sharing as fostered by an open source license.

You don't have to do anything to *not* offer a license. You may, however, wish to add a copyright notice and statement that you are not offering any license in a prominent place (e.g., your project's README) so that **users** don't assume you made an oversight. If you're going to accept others' contributions to your non-licensed project, you may wish to explore adding a contributor agreement to your project with your lawyer so that you maintain copyright permission from contributors, even though you're not granting the same.

Disallowing use of your code might not be what you intend by "no license." An **open source license** allows reuse of your code while retaining copyright. If your goal is to completely opt-out of copyright restrictions, try a **public domain dedication** instead.

For users

If you find software that doesn't have a license, that generally means you have no permission from the creators of the software to use, modify, or share the software. Although a code host such as GitHub may allow you to view and fork the code, this does not imply that you are permitted to use, modify, or share the software for any purpose.

Your options:

- **Ask the maintainers nicely to add a license.** Unless the software includes strong indications to the contrary, lack of a license is probably an oversight. If the software is hosted on a site like GitHub, open an issue requesting a license and include a link to this site. If you're bold and it's fairly obvious what license is most appropriate, open a pull request to add a license – see "suggest this license" in the sidebar of the page for each license on this site (e.g., **MIT**).
- **Don't use the software.** Find or create an alternative that is under an open source license.
- **Negotiate a private license.** Bring your lawyer.

Source here

Recommendations summary

- 1: Plan in advance
- 2: Consider adequate file system for the project
- 3: Create accessible connected workflows
- 4: Document, document, document
- 5: Consider using a code environment container
- 6: Add a license

Think about all this in advance. It is worth your while!

Many thanks!

patricia.menendez@monash.edu
@PM_maths
https://okayama1.github.io/Australassian-applied-statistics-conference-talk/