Supported by:

British Columbia
Ministry of Forests

MACROECOLOGY
GROUP

Reproducible workflows with R and GitHub

Ceres Barros

July 12th, 2023
2023 MacroBrum
Birmingham UK

Outline

1. The importance of repeatability, reproducibility, reusability and transparency – R^{3}T

2. General guidelines

3. A working example in R and GitHub

Outline

1. The importance of repeatability, reproducibility, reusability and transparency – R^{3}T

2. General guidelines

3. A working example in R and GitHub

Repeatability, reproducibility, reusability and transparency R³T

What?

Repeatability ≠ Reproducibility ≠ Reusability

Repeatability, reproducibility, reusability and transparency R^{3}T

What?

Repeatability ≠ Reproducibility ≠ Reusability

agreement of results obtained by the same individual using same methods

Repeatability, reproducibility, reusability and transparency R^3T

What?

Repeatability ≠ Reproducibility ≠ Reusability

agreement of results obtained by the same individual using same methods

agreement of results obtained by two individuals/groups using same methods

Repeatability, reproducibility, reusability and transparency
R^{3}T
What?

Repeatability ≠ Reproducibility ≠ Reusability

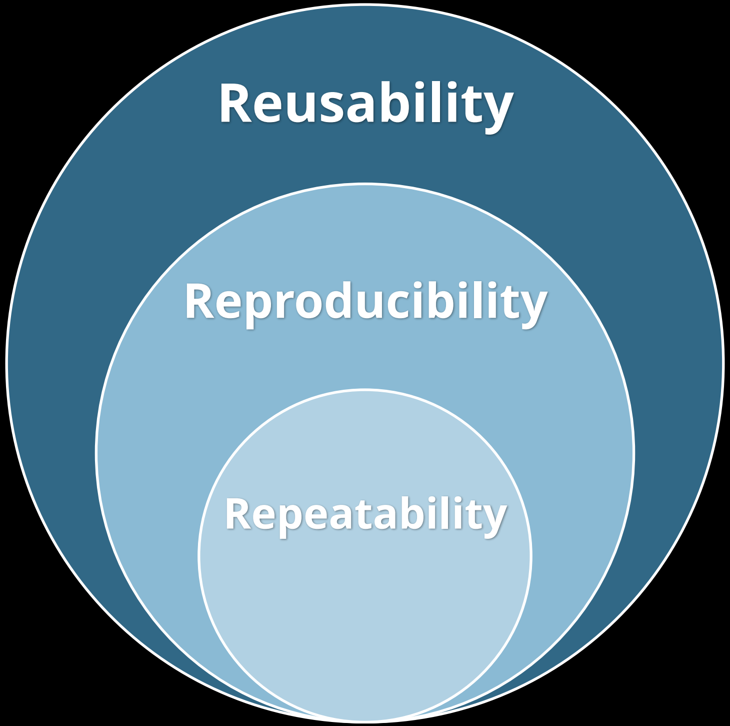
agreement of results obtained by the same individual using same methods

ability to re-use the same methods in a different context
(e.g. new study area)

agreement of results obtained by two individuals/groups using same methods

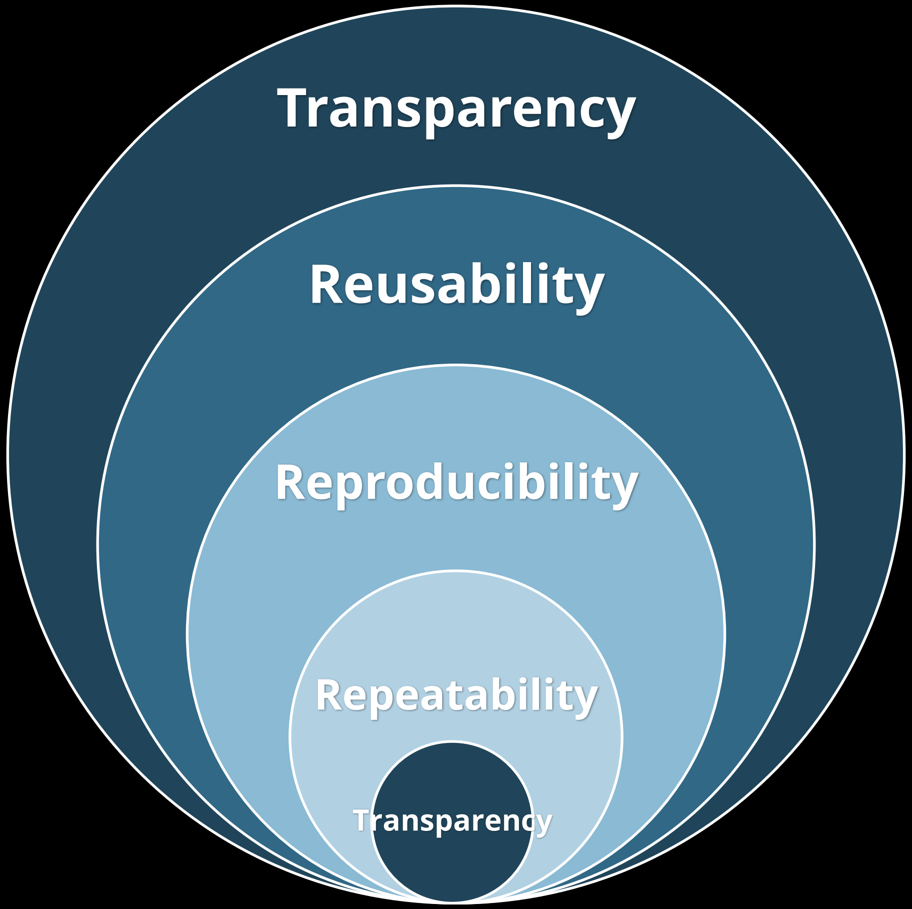
Repeatability, reproducibility, reusability and transparency R^{3}T

What?



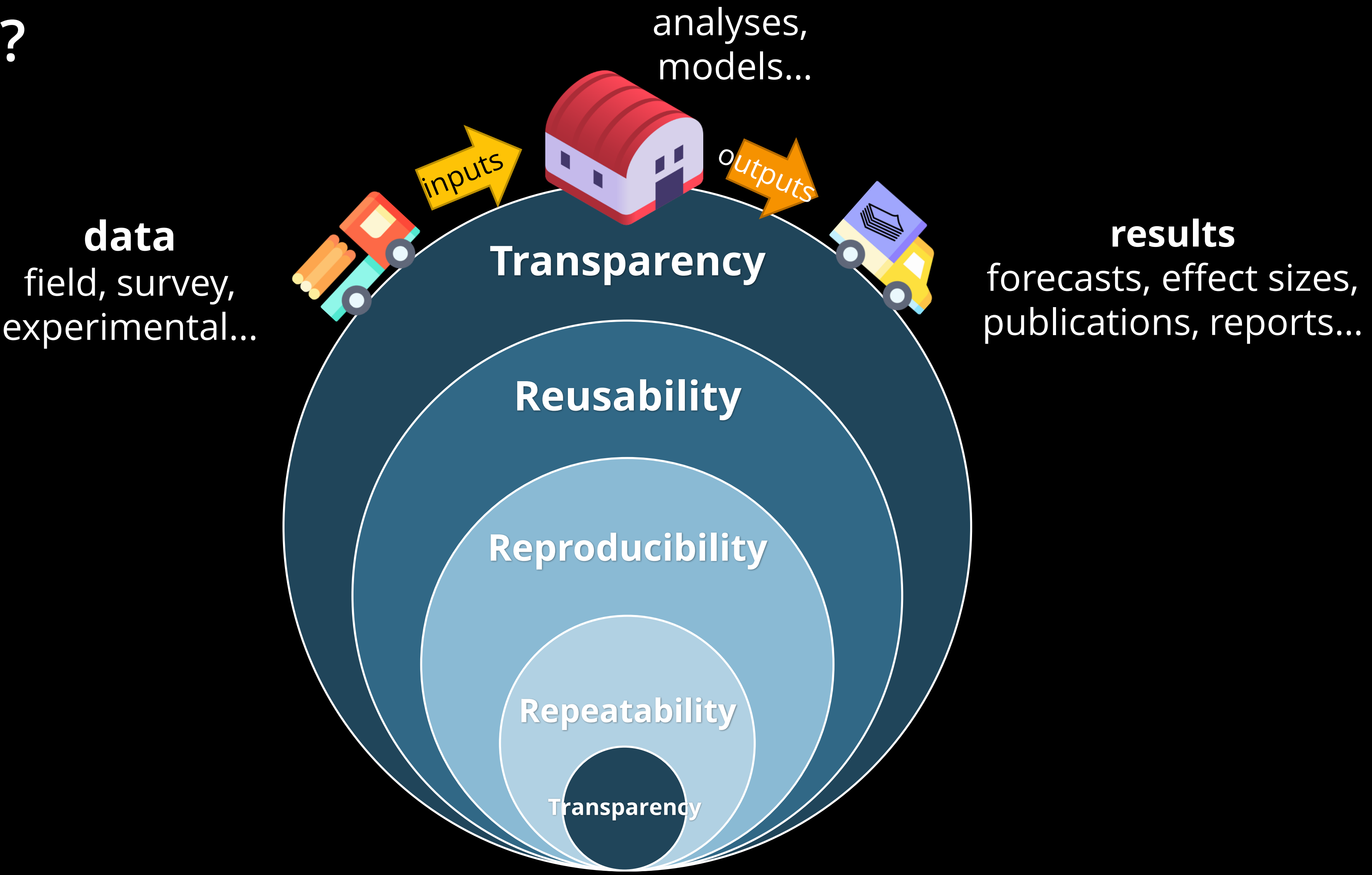
Repeatability, reproducibility, reusability and transparency R^{3}T

What?



Repeatability, reproducibility, reusability and transparency R³T

What?



Repeatability, reproducibility, reusability and transparency R^{3}T

Why?

Trust

Repeatability, reproducibility, reusability and transparency R³T

Why?

Trust

Benchmarking & meta-analyses

Repeatability, reproducibility, reusability and transparency R^{3}T

Why?

Trust

Benchmarking & meta-analyses

Building-on & improving analyses/models/workflows

Repeatability, reproducibility, reusability and transparency R³T

How?

Depends on **context**

- Project type and size
- Purpose
- Audience

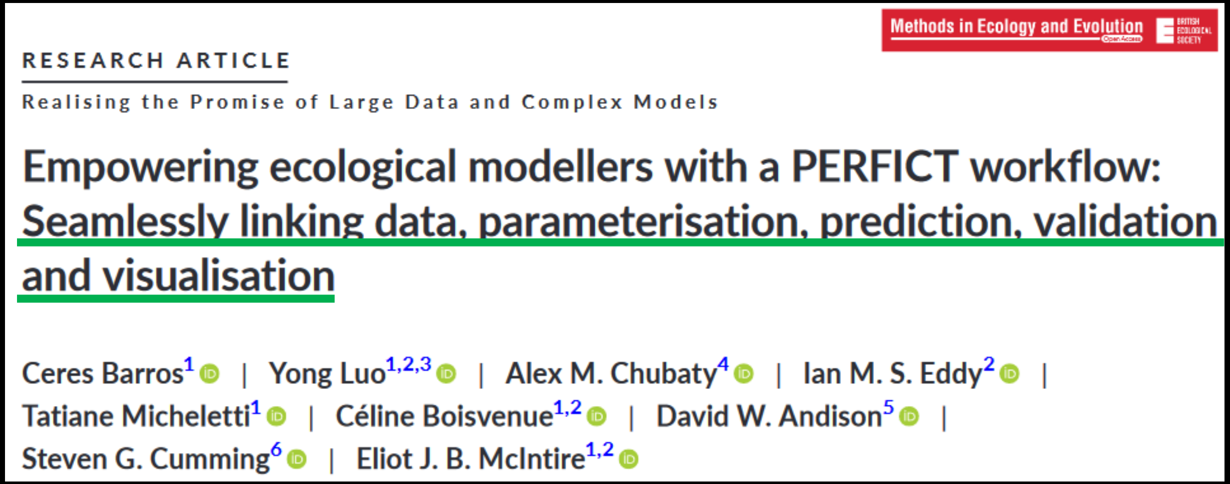
Repeatability, reproducibility, reusability and transparency R^{3}T
How?

Depends on **context**

- Project type and size
- Purpose
- Audience

Data (both input and output) types
Input and output management
Suitable workflow

Repeatability, reproducibility, reusability and transparency
R³T
How?



Methods in Ecology and Evolution

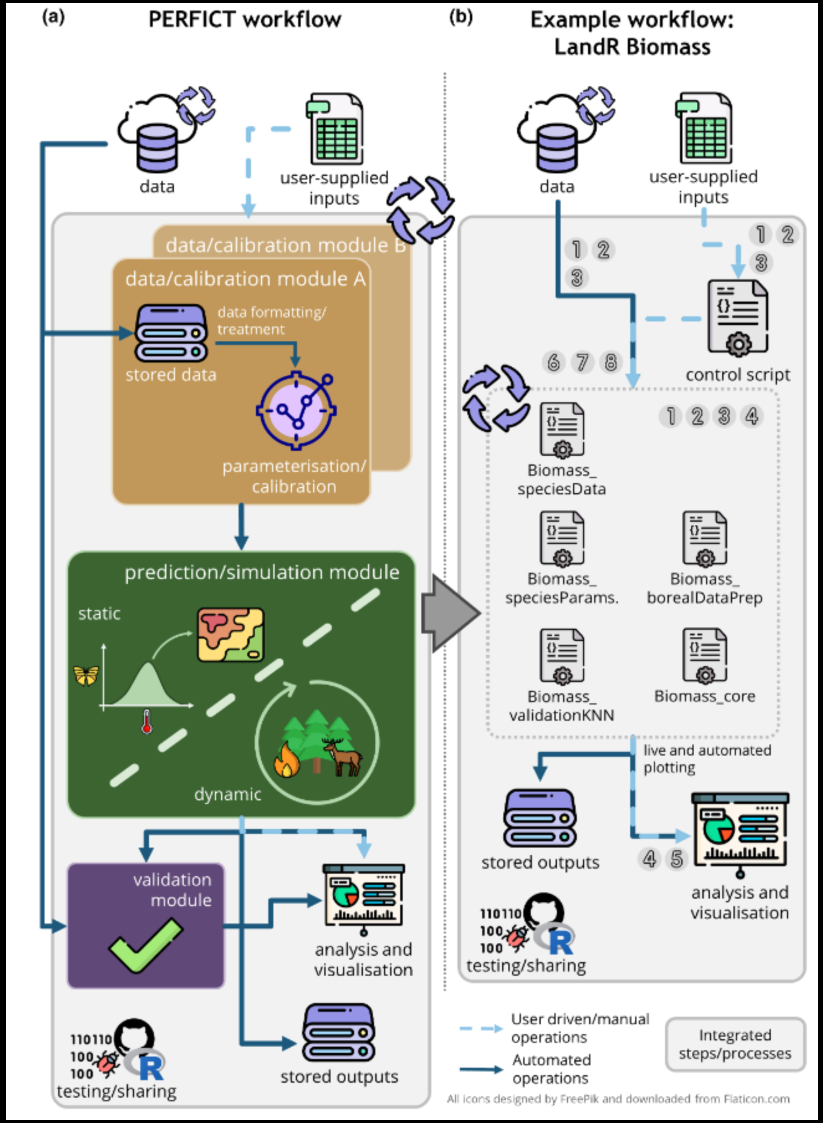
BRITISH ECOLOGICAL SOCIETY

RESEARCH ARTICLE

Realising the Promise of Large Data and Complex Models

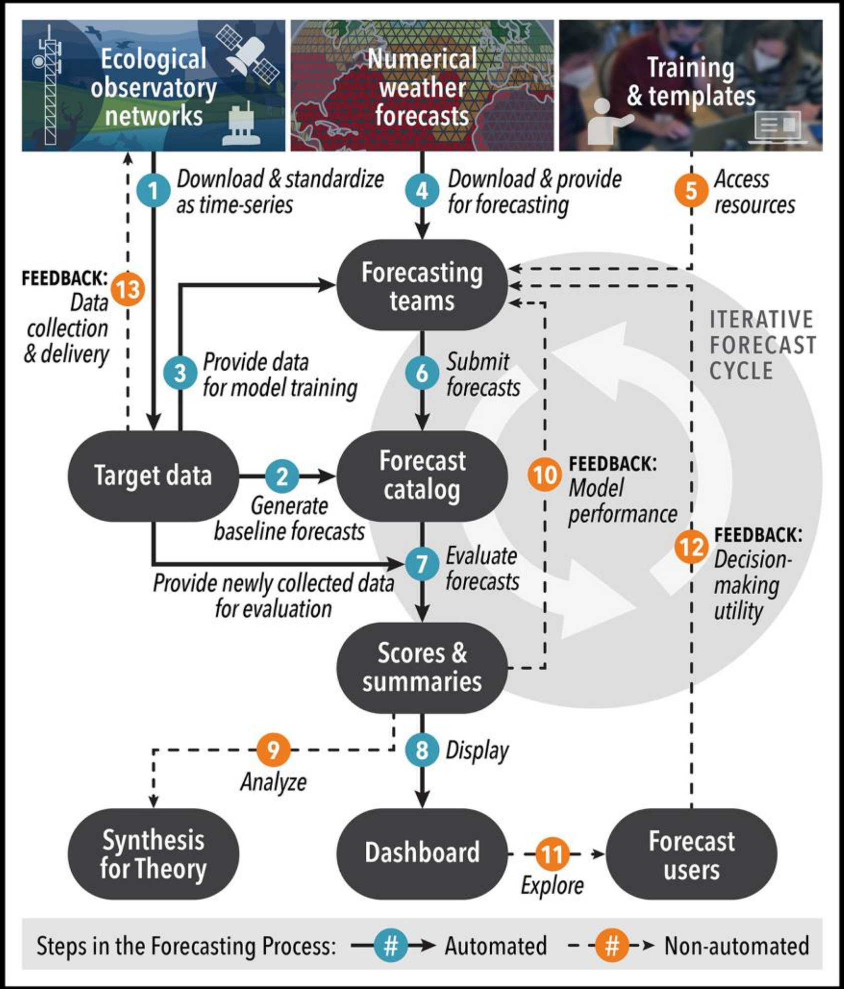
Empowering ecological modellers with a PERFICT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation

Ceres Barros[1] | Yong Luo[1,2,3] | Alex M. Chubaty[4] | Ian M. S. Eddy[2] | Tatiane Micheletti[1] | Céline Boisvenue[1,2] | David W. Andison[5] | Steven G. Cumming[6] | Eliot J. B. McIntire[1,2]


(a)
PERFICT workflow
(b)
Example workflow: LandR Biomass
data
user-supplied inputs
data/calibration module B
data/calibration module A
stored data
data formatting/ treatment
parameterisation/ calibration
prediction/simulation module
static
dynamic
validation module
analysis and visualisation
stored outputs
testing/sharing
data
user-supplied inputs
1 2 3
control script
6 7 8
1 2 3 4
Biomass_ speciesData
Biomass_ speciesParams.
Biomass_ borealDataPrep
Biomass_ validationKNN
Biomass_core
live and automated plotting
stored outputs
4 5
analysis and visualisation
testing/sharing
User driven/manual operations
Automated operations
Integrated steps/processes
All icons designed by FreePik and downloaded from Flaticon.com

Repeatability, reproducibility, reusability and transparency R^{3}T

How?


Ecological observatory networks
Numerical weather forecasts
Training & templates
1 Download & standardize as time-series
4 Download & provide for forecasting
5 Access resources
FEEDBACK: 13 Data collection & delivery
Forecasting teams
ITERATIVE FORECAST CYCLE
3 Provide data for model training
6 Submit forecasts
Target data
2 Generate baseline forecasts
Forecast catalog
10 FEEDBACK: Model performance
12 FEEDBACK: Decision-making utility
Provide newly collected data for evaluation
7 Evaluate forecasts
Scores & summaries
9 Analyze
8 Display
Synthesis for Theory
Dashboard
11 Explore
Forecast users
Steps in the Forecasting Process: # Automated # Non-automated

NEON Forecasting Challenge workflow
Thomas et al. (2023)

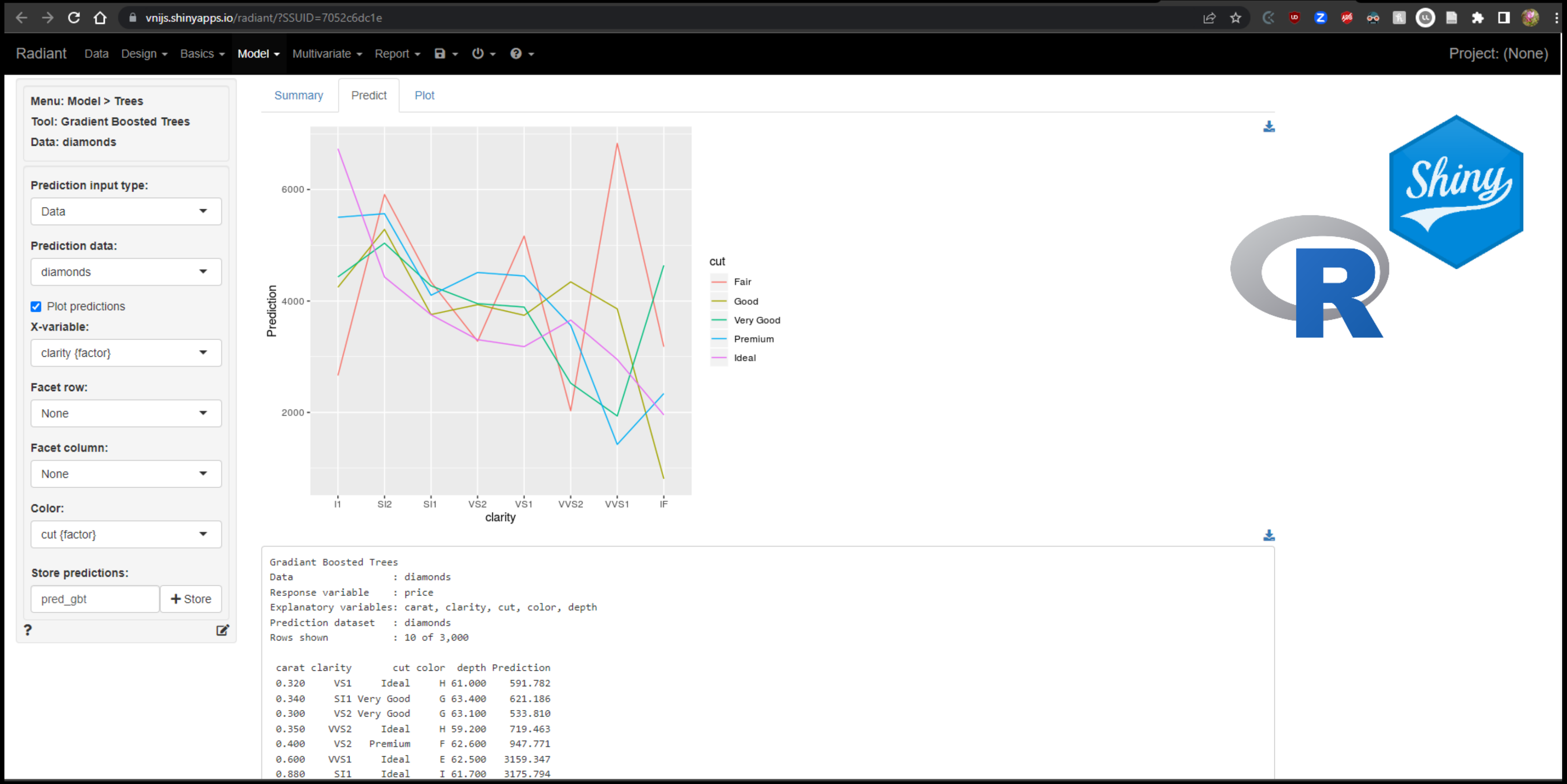
Ecological (iterative) forecasting (continuous and integrated) workflow based on monitoring data

Repeatability, reproducibility, reusability and transparency R^{3}T

How?

engagement/education
point of view

R-shiny apps can be useful for education, engaging stakeholders/public and delivering an interactive product to end-users



Menu: Model > Trees
Tool: Gradient Boosted Trees
Data: diamonds

Prediction input type: Data

Prediction data: diamonds

☑ Plot predictions

X-variable: clarity {factor}

Facet row: None

Facet column: None

Color: cut {factor}

Store predictions: pred_gbt

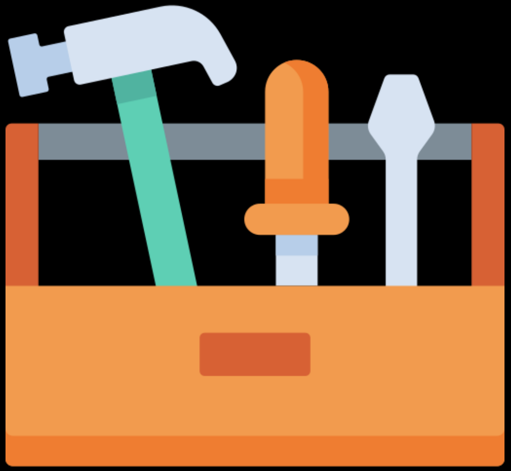
```
Gradiant Boosted Trees
Data                   : diamonds
Response variable      : price
Explanatory variables: carat, clarity, cut, color, depth
Prediction dataset     : diamonds
Rows shown             : 10 of 3,000

 carat clarity       cut color  depth Prediction
 0.320     VS1     Ideal     H 61.000    591.782
 0.340     SI1 Very Good     G 63.400    621.186
 0.300     VS2 Very Good     G 63.100    533.810
 0.350    VVS2     Ideal     H 59.200    719.463
 0.400     VS2   Premium     F 62.600    947.771
 0.600    VVS1     Ideal     E 62.500   3159.347
 0.880     SI1     Ideal     I 61.700   3175.794
```

https://vnijs.shinyapps.io/radiant/?SSUID=03eddd27f4

Repeatability, reproducibility, reusability and transparency R^{3}T

How?

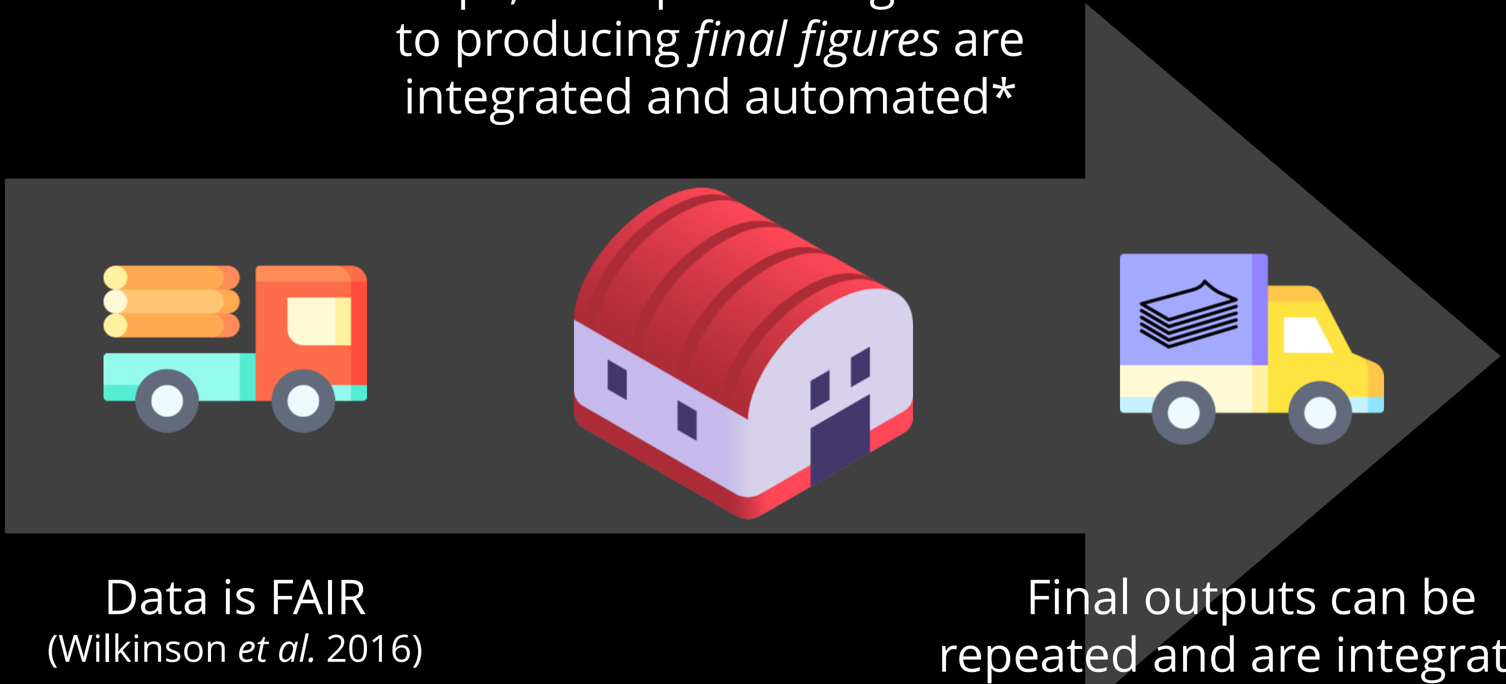
Most ecological research likely benefits from using a R^{3}T approach, but the tools used to accomplish it can be varied



Repeatability, reproducibility, reusability and transparency R^{3}T

How?

All steps, from processing *raw data* to producing *final figures* are integrated and automated*



Data is FAIR
(Wilkinson *et al.* 2016)

Final outputs can be repeated and are integrated in the reporting**

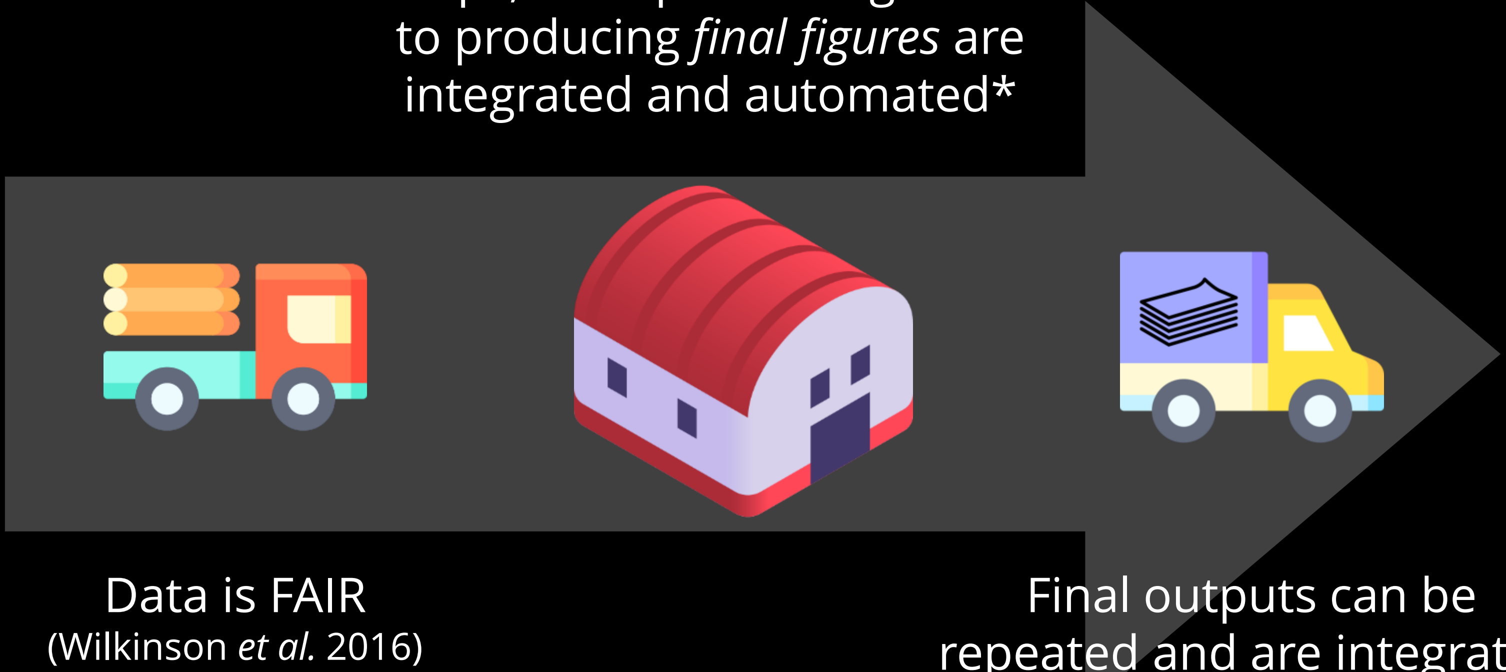
*as much as possible
**directly, or indirectly via links

Repeatability, reproducibility, reusability and transparency R^{3}T

How?

Self-contained

All steps, from processing *raw data* to producing *final figures* are integrated and automated*



Data is FAIR
(Wilkinson *et al.* 2016)

Final outputs can be repeated and are integrated in the reporting**

*as much as possible
**directly, or indirectly via links

Outline

1. The importance of repeatability, reproducibility, reusability and transparency – R3T

2. General guidelines

3. A working example in R and GitHub

General guidelines

1. Scripting/executing the workflow

1.1. Script, script, script

- **Goal**: no “secret handshakes” + record all steps of an analysis
- ALL steps – this includes package/library installation/loading and sourcing data

DOComment your code

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines
1. Scripting/executing the workflow

1.1. Script, script, script

1.2. Minimise software/languages used

- **Goal**: increase workflow robustness - fewer “moving parts”, fewer “secret handshakes”, fewer manual operations
- Interpreted languages (real-time user interaction)

  R, Julia, Python…
- Compiled languages (pre-compiled programs)

  C, C++, C#, Fortran,… Do you really need this?

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines

1. Scripting/executing the workflow

1.1. Script, script, script

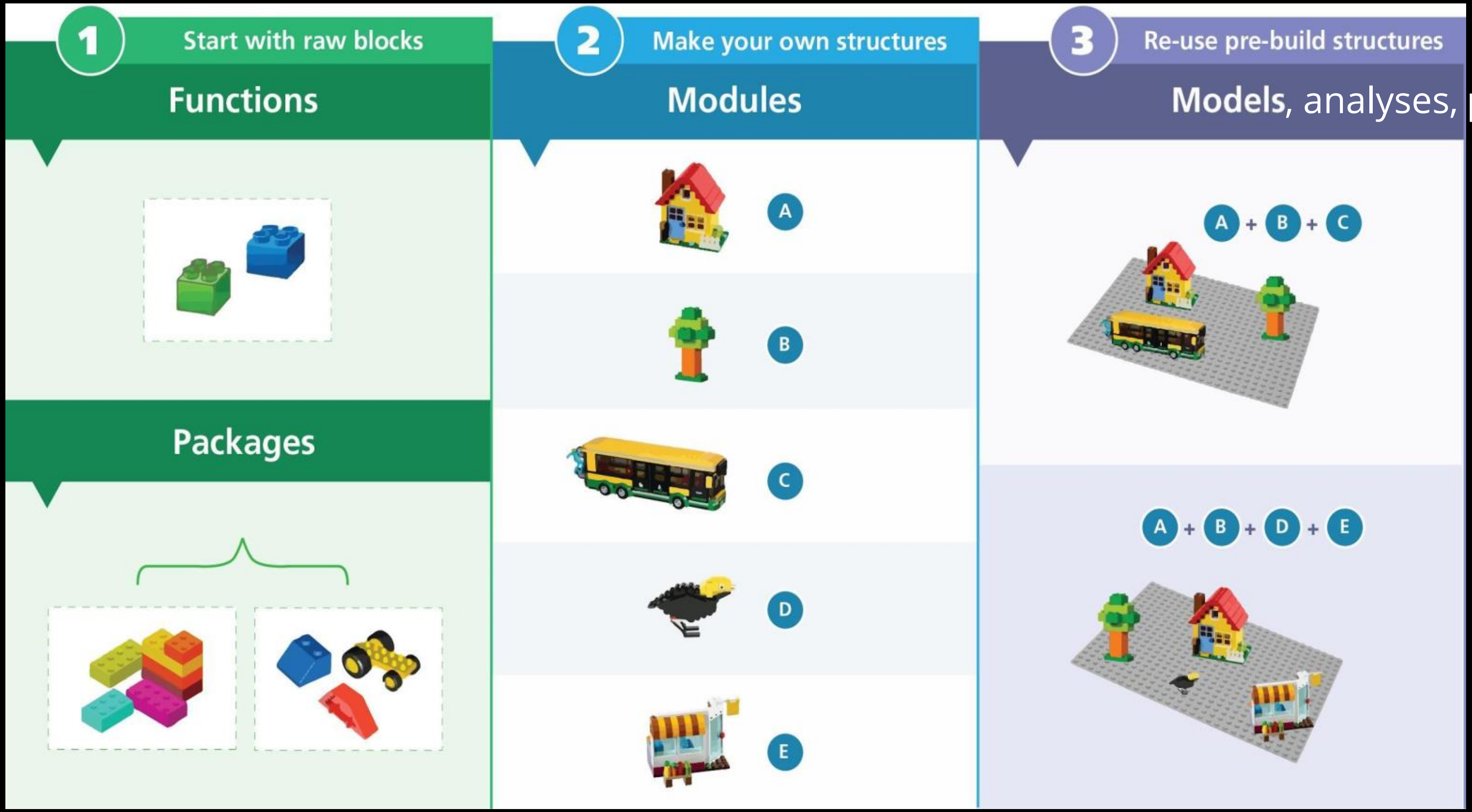
1.2. Minimise software/languages used

1.3. Modularise and "functionise" (!)

- **Goal:** code organisation/readability; easier propagation of code updates/changes
- Avoid looooooooong scripts
- Break scripts into logical pieces
- Encapsulate code into functions, *especially* when used multiple times/in multiple places
- Consider "packaging" your functions.

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines
1. Scripting/executing the workflow



, analyses, projects…

Functions and modules as key tools for R³T, but also for building integrated and continuous workflows
McIntire *et al.* (2022)

General guidelines

1. Scripting/executing the workflow

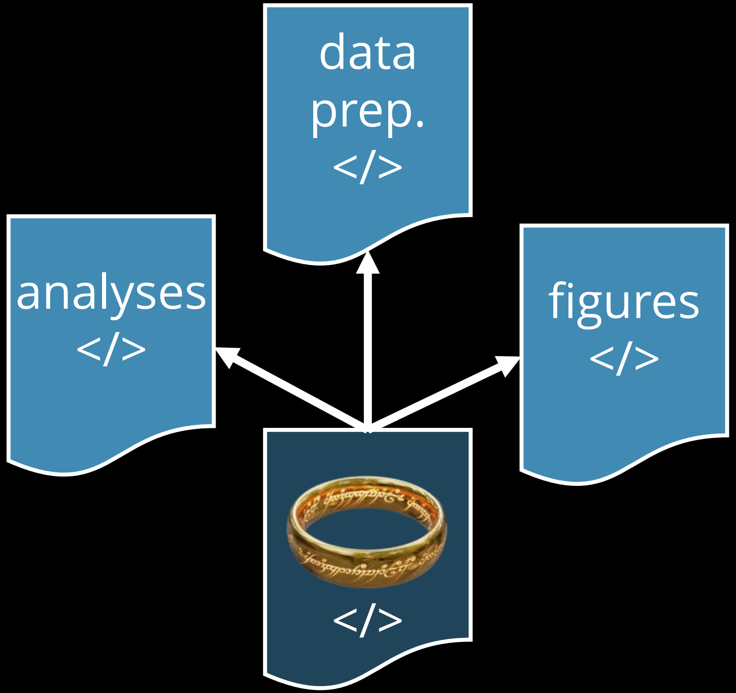
1.1. Script, script, script

1.2. Minimise software/languages used

1.3. Modularise and “functionise” (!)

1.4. Centralise workflow in a single script

- **Goal:** no “secret handshakes” - all scripts are utilised in correct way/sequence
- Call/execute scripts/steps from central (“control”) script



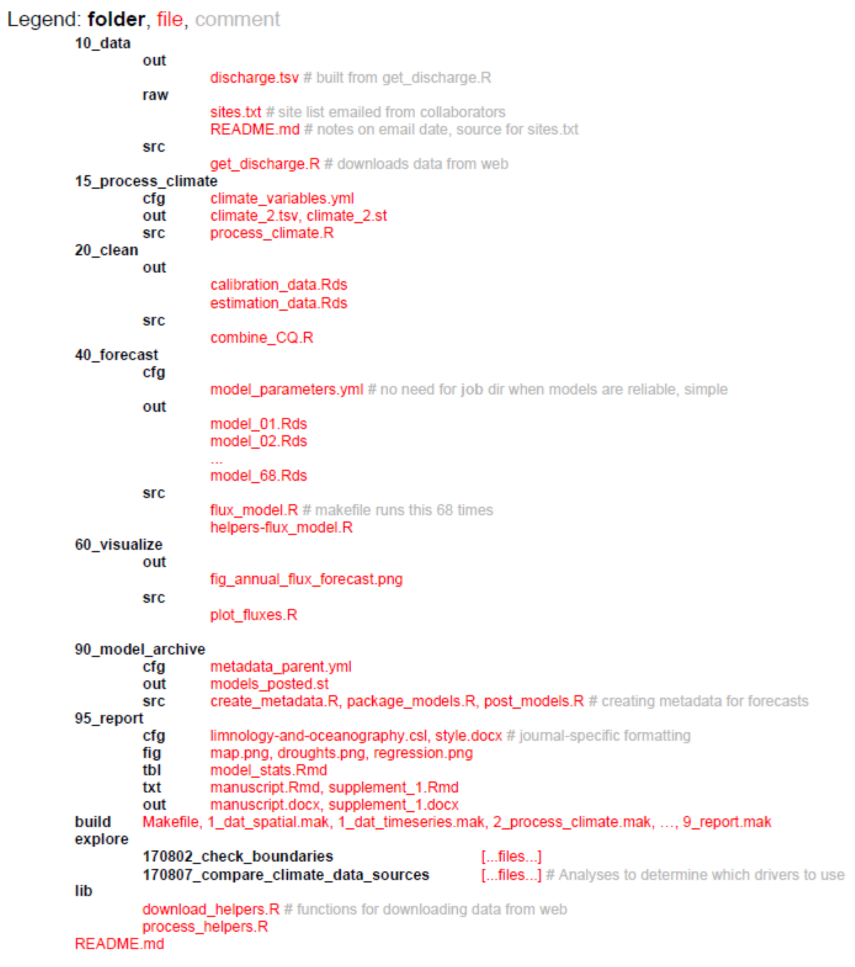
Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines

2. Project structure

2.1. Project-oriented workflows

- **Goal:** the entirely workflow can be re-run easily, and without changing code or files
- Choose a structure that is self-explanatory
- Relative paths *vs.* absolute paths
- Project-libraries



```
Legend: folder, file, comment
        10_data
                out
                        discharge.tsv # built from get_discharge.R
                raw
                        sites.txt # site list emailed from collaborators
                        README.md # notes on email date, source for sites.txt
                src
                        get_discharge.R # downloads data from web
        15_process_climate
                cfg     climate_variables.yml
                out     climate_2.tsv, climate_2.st
                src     process_climate.R
        20_clean
                out
                        calibration_data.Rds
                        estimation_data.Rds
                src
                        combine_CQ.R
        40_forecast
                cfg
                        model_parameters.yml # no need for job dir when models are reliable, simple
                out
                        model_01.Rds
                        model_02.Rds
                        ...
                        model_68.Rds
                src
                        flux_model.R # makefile runs this 68 times
                        helpers-flux_model.R
        60_visualize
                out
                        fig_annual_flux_forecast.png
                src
                        plot_fluxes.R

        90_model_archive
                cfg     metadata_parent.yml
                out     models_posted.st
                src     create_metadata.R, package_models.R, post_models.R # creating metadata for forecasts
        95_report
                cfg     limnology-and-oceanography.csl, style.docx # journal-specific formatting
                fig     map.png, droughts.png, regression.png
                tbl     model_stats.Rmd
                txt     manuscript.Rmd, supplement_1.Rmd
                out     manuscript.docx, supplement_1.docx
        build   Makefile, 1_dat_spatial.mak, 1_dat_timeseries.mak, 2_process_climate.mak, …, 9_report.mak
        explore
                170802_check_boundaries                 [...files...]
                170807_compare_climate_data_sources     [...files...] # Analyses to determine which drivers to use
        lib
                download_helpers.R # functions for downloading data from web
                process_helpers.R
        README.md
```

Source: https://ecoforecast.org/reproducible-forecasting-workflows/

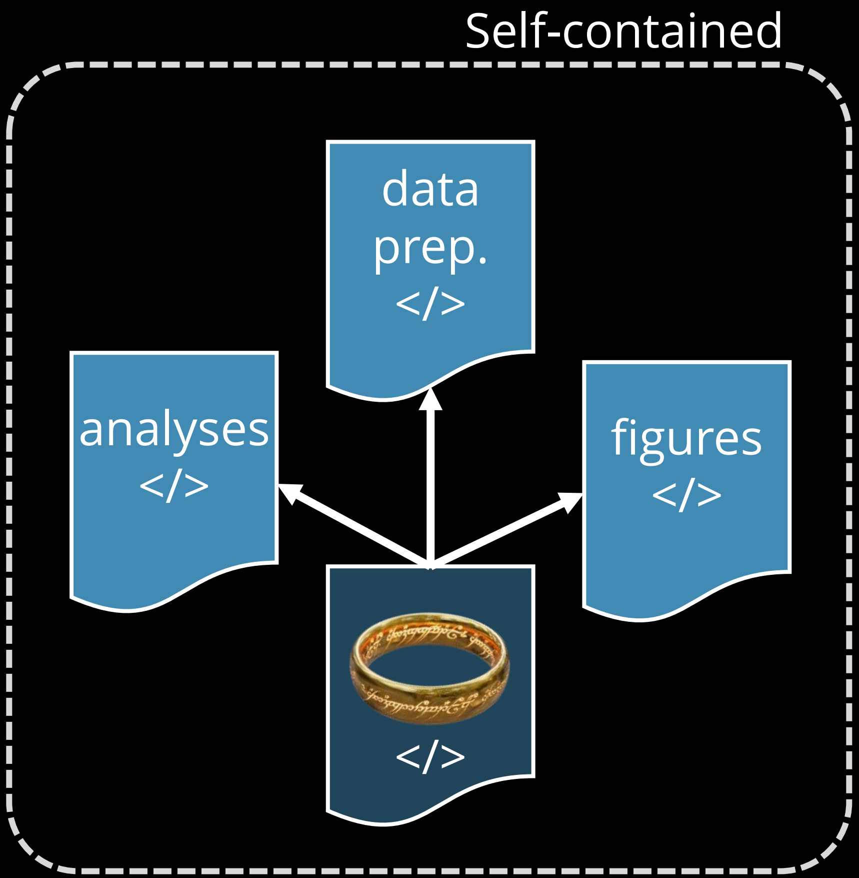
Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines
2. Project structure

2.1. Project-oriented workflows

2.2. Self-contained workflows

- **Goal:** ensure reproducibility
- E.g. RStudio-projects
- Containerisation – encapsulates the whole system (even OS) – e.g. Docker

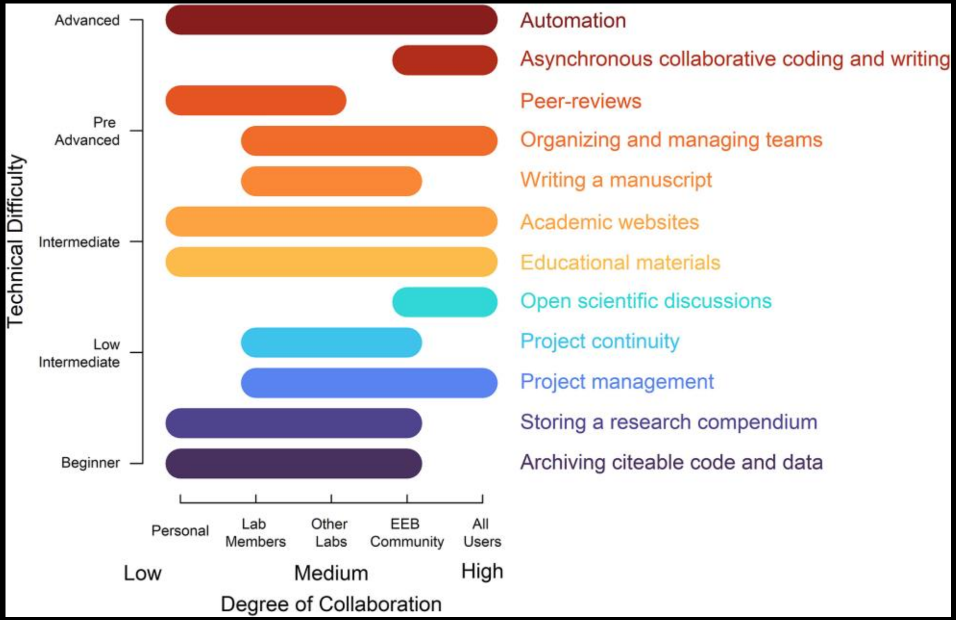


Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines
3. Project management

3.1. Version control

- **Goal:** change tracking in code/files + continuous and collaborative development
- Keeps a formal record of all changes
- Allows recovering old versions
- Allows keeping/working on multiple versions of the same code/project
- E.g. Git, CVS, SVN, …



GitHub is a multifaceted tool that can be appropriate to manage, track and collaborate on projects for various purposes and at various levels of complexity.
(Braga *et al*. 2023)

We'll come back to this!

GitHub

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines
3. Project management

3.1. Version control

3.2. Integrated testing

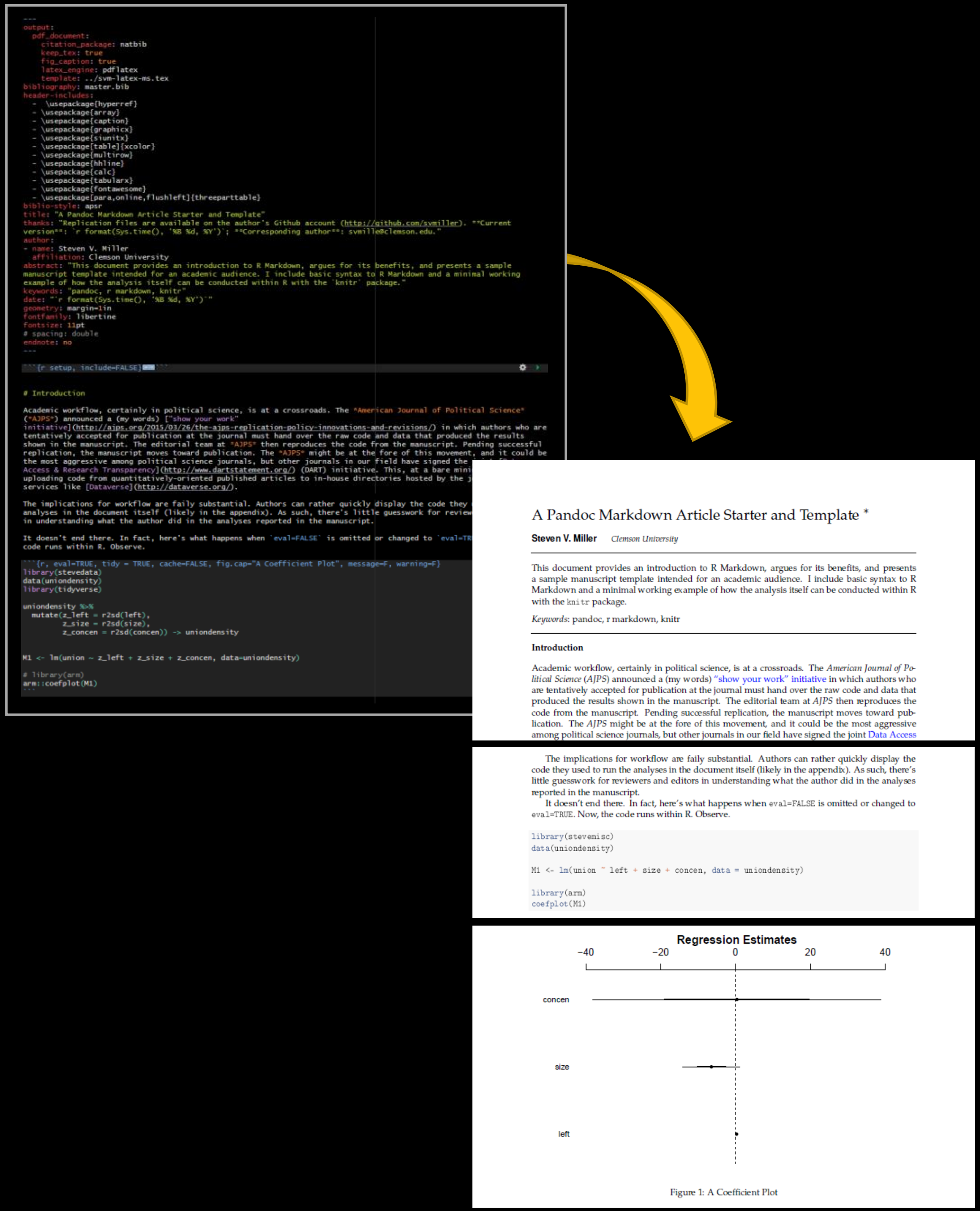
- **Goal:** enhance code robustness and longevity
- Not always necessary, but always a good idea ;)
- Integration tests *vs* unit tests *vs* assertions
- Manual *vs* automated execution
- E.g.
  - `testthat` in R (unit tests)
  - simple code/object checks (assertions)
  - GitHub Actions and Travis CI for automated testing – all types.

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

General guidelines
4. Literate programming

4.1. Integrate code and reporting/publication

- **Goal**: establishing explicit links between report/publication, data and analyses
- Integrates code and text in a single file
- Enhances transparency/reproducibility of reported outputs.
- E.g.
    - RMarkdown, Quarto – static or interactive; multiple languages in a single file
    - Jupyter – interactive; single language at a time (Julia, R or Python)



~~~
---
output:
  pdf_document:
    citation_package: natbib
    keep_tex: true
    fig_caption: true
    latex_engine: pdflatex
    template: ../svm-latex-ms.tex
bibliography: master.bib
header-includes:
  -  \usepackage{hyperref}
  - \usepackage{array}
  - \usepackage{caption}
  - \usepackage{graphicx}
  - \usepackage{siunitx}
  - \usepackage[table]{xcolor}
  - \usepackage{multirow}
  - \usepackage{hhline}
  - \usepackage{calc}
  - \usepackage{tabularx}
  - \usepackage{fontawesome}
  - \usepackage[para,online,flushleft]{threeparttable}
biblio-style: apsr
title: "A Pandoc Markdown Article Starter and Template"
thanks: "Replication files are available on the author's Github account (http://github.com/svmiller). **Current
version**: `r format(Sys.time(), '%B %d, %Y')`; **Corresponding author**: svmille@clemson.edu."
author:
- name: Steven V. Miller
  affiliation: Clemson University
abstract: "This document provides an introduction to R Markdown, argues for its benefits, and presents a sample
manuscript template intended for an academic audience. I include basic syntax to R Markdown and a minimal working
example of how the analysis itself can be conducted within R with the `knitr` package."
keywords: "pandoc, r markdown, knitr"
date: "`r format(Sys.time(), '%B %d, %Y')`"
geometry: margin=1in
fontfamily: libertine
fontsize: 11pt
# spacing: double
endnote: no
---

```{r setup, include=FALSE}```


# Introduction

Academic workflow, certainly in political science, is at a crossroads. The *American Journal of Political Science*
(*AJPS*) announced a (my words) ["show your work"
initiative](http://ajps.org/2015/03/26/the-ajps-replication-policy-innovations-and-revisions/) in which authors who are
tentatively accepted for publication at the journal must hand over the raw code and data that produced the results
shown in the manuscript. The editorial team at *AJPS* then reproduces the code from the manuscript. Pending successful
replication, the manuscript moves toward publication. The *AJPS* might be at the fore of this movement, and it could be
the most aggressive among political science journals, but other journals in our field have signed the
Access & Research Transparency](http://www.dartstatement.org/) (DART) initiative. This, at a bare mini
uploading code from quantitatively-oriented published articles to in-house directories hosted by the j
services like [Dataverse](http://dataverse.org/).

The implications for workflow are faily substantial. Authors can rather quickly display the code they
analyses in the document itself (likely in the appendix). As such, there's little guesswork for review
in understanding what the author did in the analyses reported in the manuscript.

It doesn't end there. In fact, here's what happens when `eval=FALSE` is omitted or changed to `eval=TR
code runs within R. Observe.

```{r, eval=TRUE, tidy = TRUE, cache=FALSE, fig.cap="A Coefficient Plot", message=F, warning=F}
library(stevedata)
data(uniondensity)
library(tidyverse)

uniondensity %>%
  mutate(z_left = r2sd(left),
         z_size = r2sd(size),
         z_concen = r2sd(concen)) -> uniondensity


M1 <- lm(union ~ z_left + z_size + z_concen, data=uniondensity)

# library(arm)
arm::coefplot(M1)
```
~~~

A Pandoc Markdown Article Starter and Template *

Steven V. Miller *Clemson University*

This document provides an introduction to R Markdown, argues for its benefits, and presents a sample manuscript template intended for an academic audience. I include basic syntax to R Markdown and a minimal working example of how the analysis itself can be conducted within R with the `knitr` package.

Keywords: pandoc, r markdown, knitr

Introduction

Academic workflow, certainly in political science, is at a crossroads. The *American Journal of Political Science* (*AJPS*) announced a (my words) "show your work" initiative in which authors who are tentatively accepted for publication at the journal must hand over the raw code and data that produced the results shown in the manuscript. The editorial team at *AJPS* then reproduces the code from the manuscript. Pending successful replication, the manuscript moves toward publication. The *AJPS* might be at the fore of this movement, and it could be the most aggressive among political science journals, but other journals in our field have signed the joint Data Access

The implications for workflow are faily substantial. Authors can rather quickly display the code they used to run the analyses in the document itself (likely in the appendix). As such, there's little guesswork for reviewers and editors in understanding what the author did in the analyses reported in the manuscript.

It doesn't end there. In fact, here's what happens when `eval=FALSE` is omitted or changed to `eval=TRUE`. Now, the code runs within R. Observe.

```
library(stevemisc)
data(uniondensity)

M1 <- lm(union ~ left + size + concen, data = uniondensity)

library(arm)
coefplot(M1)
```

Figure 1: A Coefficient Plot

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

Outline

1. The importance of repeatability, reproducibility, reusability and transparency – R3T

2. General guidelines

3. A working example in R + RStudio + GitHub

Shall we try this?

What we will cover:

Project structure and management

- Version control – using GitHub and GitKraken
- Self-contained workflows – using R and Rstudio

Scripting/executing the workflow

- Script, script, script
- Modularise and "functionise" (!)
- Centralise workflow in a single script

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

Shall we try this?

What we will cover:

Project structure and management

- Version control – using GitHub and GitKraken
- Self-contained workflows – using R and Rstudio

Scripting/executing the workflow

- Script, script, script
- Modularise and "functionise" (!)
- Centralise workflow in a single script

The order is variable;
it depends on the stage of the project and your own preference

Tools used in each step can also vary

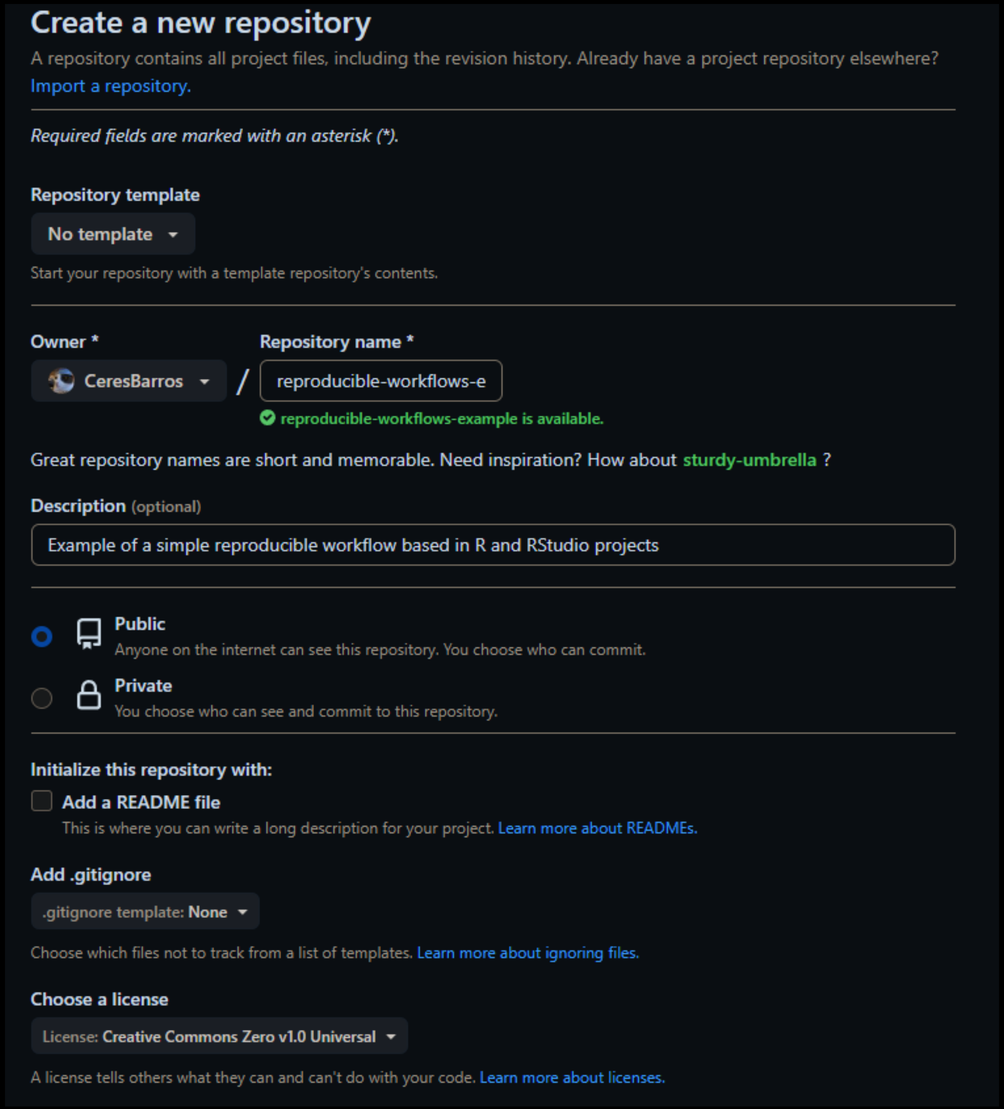
*you can do this in…

Adapted from EFI
https://ecoforecast.org/reproducible-forecasting-workflows/

1. Create a repository for your project

Assuming you already have an account on GitHub.com…

Create a repo



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner * CeresBarros ▾ / **Repository name *** reproducible-workflows-e

✓ reproducible-workflows-example is available.

Great repository names are short and memorable. Need inspiration? How about sturdy-umbrella ?

Description (optional)

Example of a simple reproducible workflow based in R and RStudio projects

● **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.

Initialize this repository with:

□ **Add a README file**
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

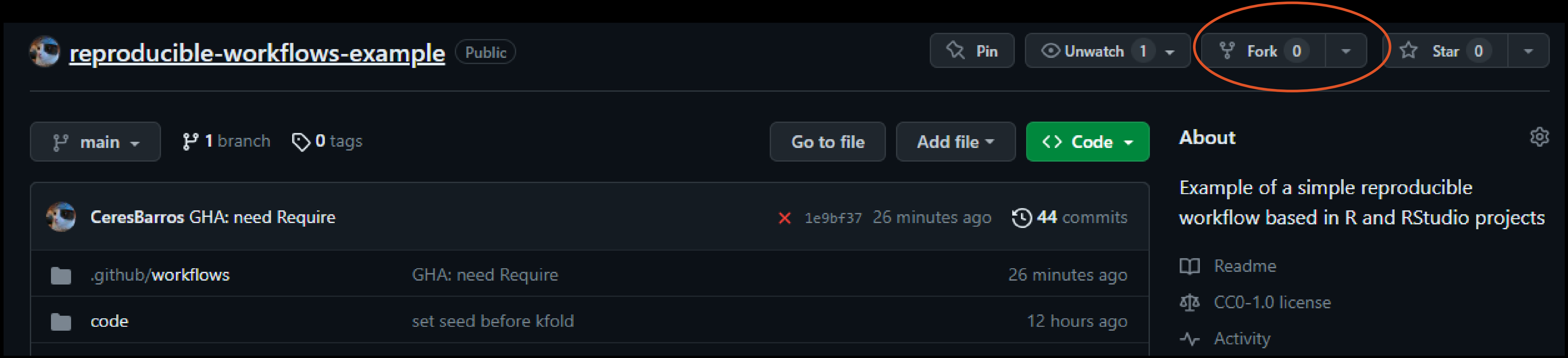
License: Creative Commons Zero v1.0 Universal ▾

A license tells others what they can and can't do with your code. Learn more about licenses.

1. Create a repository for your project

Assuming you already have an account on GitHub.com…

Or fork someone else’s



reproducible-workflows-example Public

CeresBarros GHA: need Require ✕ 1e9bf37 26 minutes ago 44 commits

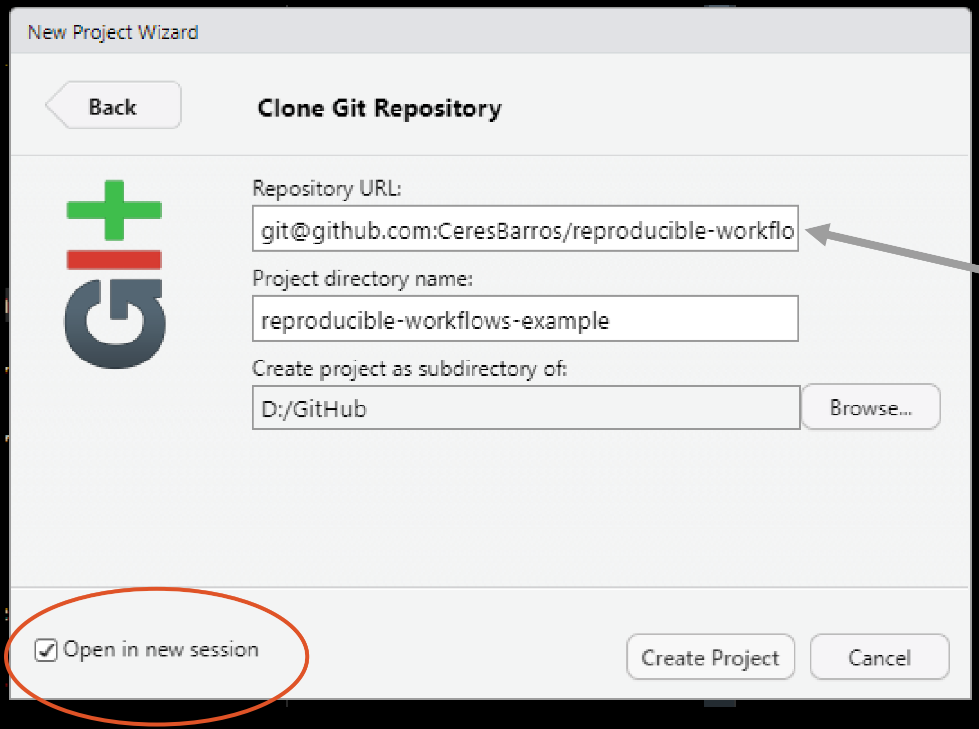
.github/workflows	GHA: need Require	26 minutes ago
code	set seed before kfold	12 hours ago

About

Example of a simple reproducible workflow based in R and RStudio projects

2. Create a self-contained project

In RStudio, go to
File > New Project… > Version Control > Git



New Project Wizard

Clone Git Repository

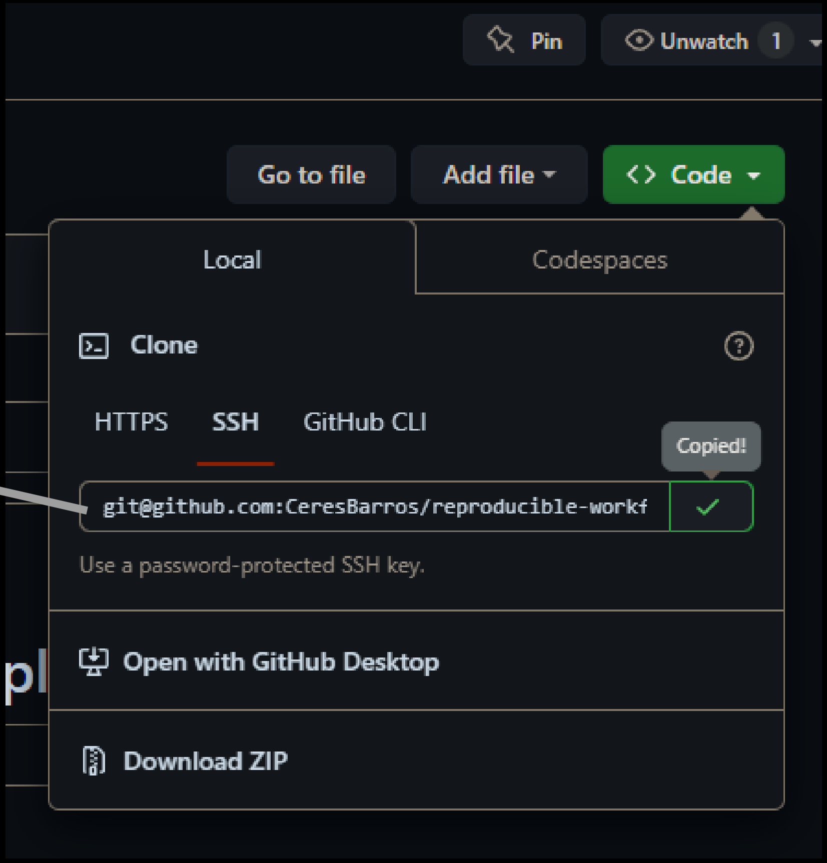
Repository URL: git@github.com:CeresBarros/reproducible-workflo

Project directory name: reproducible-workflows-example

Create project as subdirectory of: D:/GitHub

☑ Open in new session

Get repo URL from
GitHub.com/your_username/your_repo



Clone

HTTPS SSH GitHub CLI

git@github.com:CeresBarros/reproducible-workf

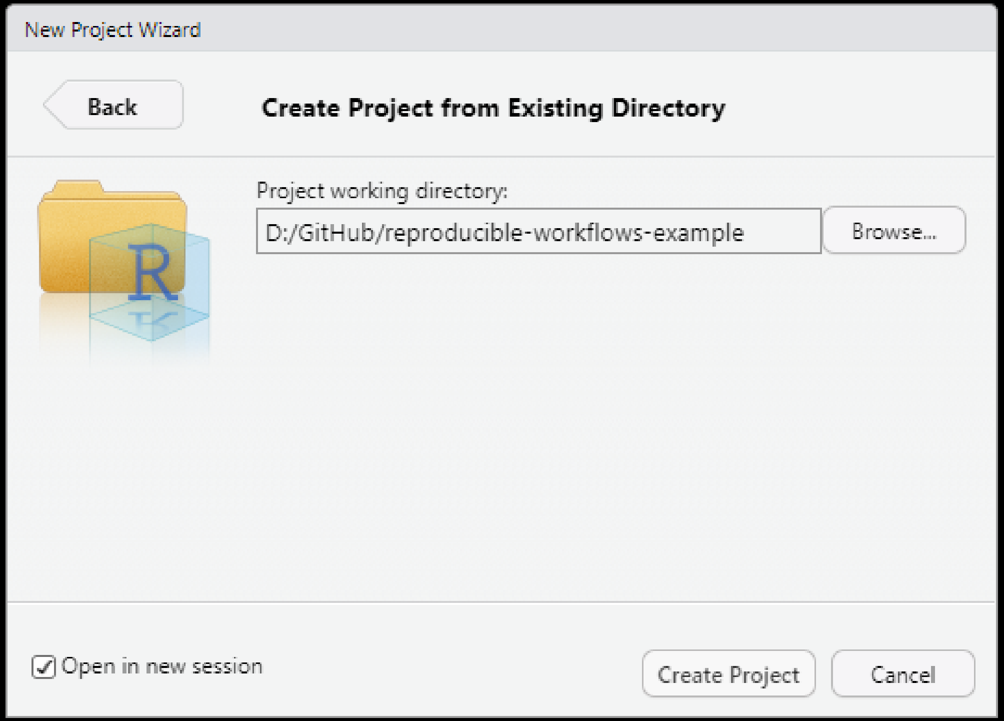
Copied!

Use a password-protected SSH key.

2. Create a self-contained project

If you already have a project folder (e.g. created by GitKraken, or from an existing project):

In RStudio, go to
File > Existing Directory



New Project Wizard

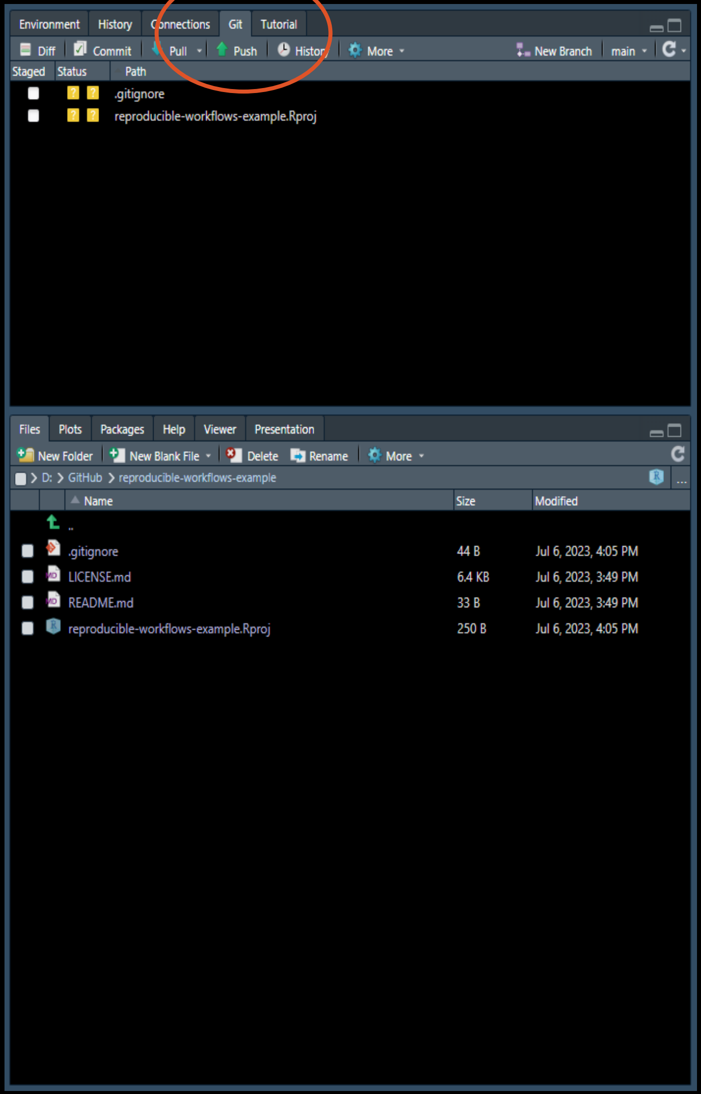
Create Project from Existing Directory

Project working directory:
D:/GitHub/reproducible-workflows-example

☑ Open in new session

2. Create a self-contained project

You can now manage your Git repo from RStudio



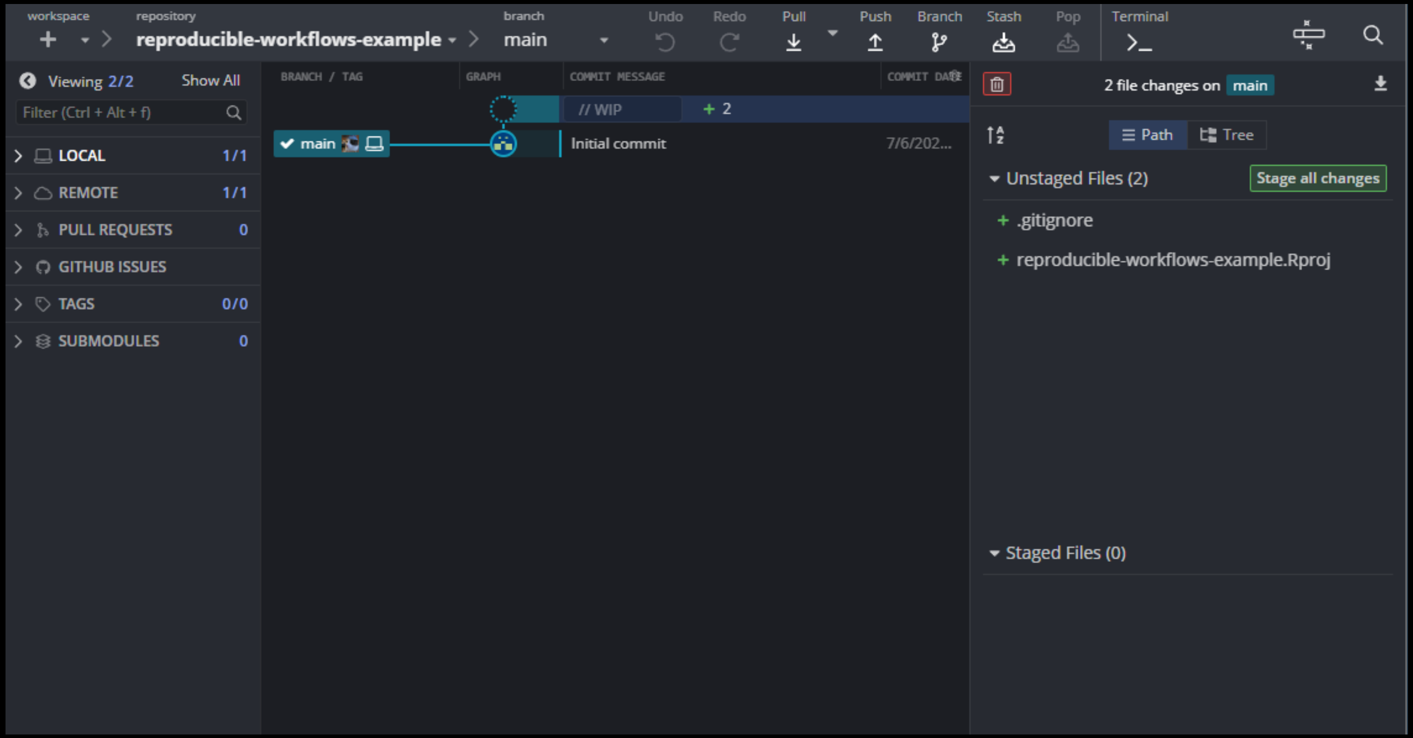
Staged	Status	Path
□	? ?	.gitignore
□	? ?	reproducible-workflows-example.Rproj

D: > GitHub > reproducible-workflows-example

Name	Size	Modified
..		
.gitignore	44 B	Jul 6, 2023, 4:05 PM
LICENSE.md	6.4 KB	Jul 6, 2023, 3:49 PM
README.md	33 B	Jul 6, 2023, 3:49 PM
reproducible-workflows-example.Rproj	250 B	Jul 6, 2023, 4:05 PM

2. Create a project

You can now manage your Git repo from RStudio,
GitKraken



BRANCH / TAG	GRAPH	COMMIT MESSAGE	COMMIT DATE
		// WIP + 2	
main		Initial commit	7/6/202...

2 file changes on main

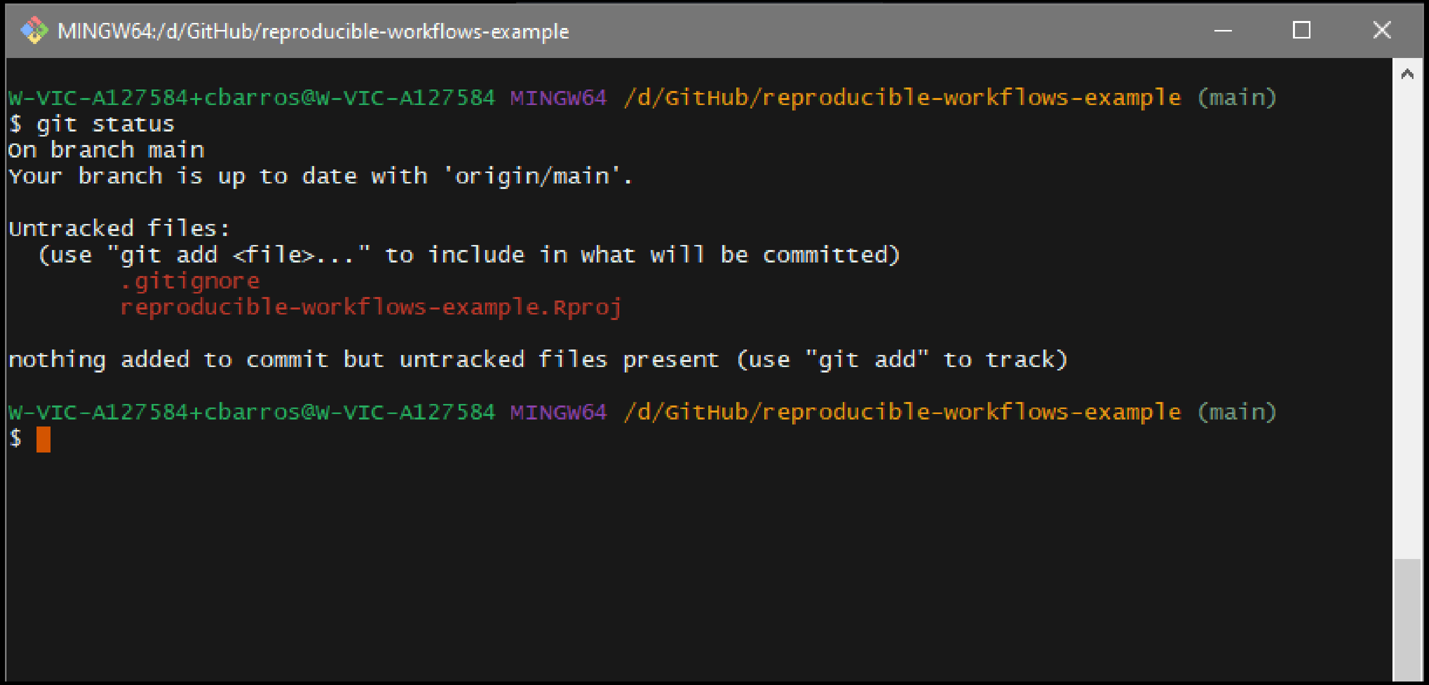
Unstaged Files (2)

+ .gitignore
+ reproducible-workflows-example.Rproj

Staged Files (0)

2. Create a project

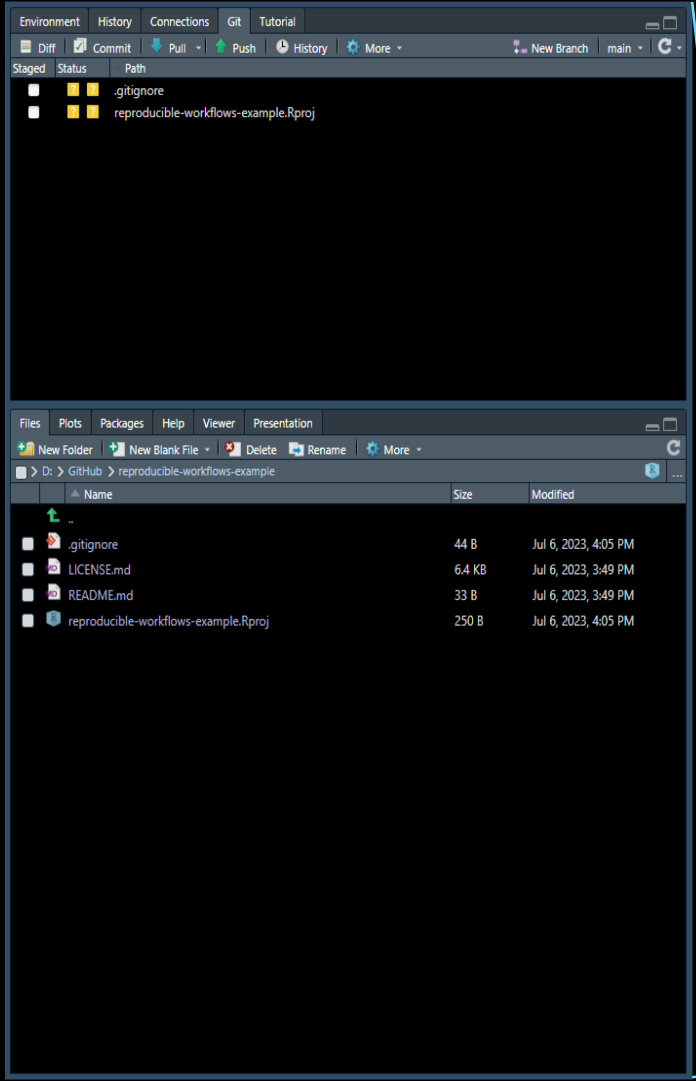
You can now manage your Git repo from RStudio,
GitKraken,
or even the command-line
(e.g., git bash for Windows)



```
W-VIC-A127584+cbarros@W-VIC-A127584 MINGW64 /d/GitHub/reproducible-workflows-example (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        reproducible-workflows-example.Rproj

nothing added to commit but untracked files present (use "git add" to track)

W-VIC-A127584+cbarros@W-VIC-A127584 MINGW64 /d/GitHub/reproducible-workflows-example (main)
$ 
```

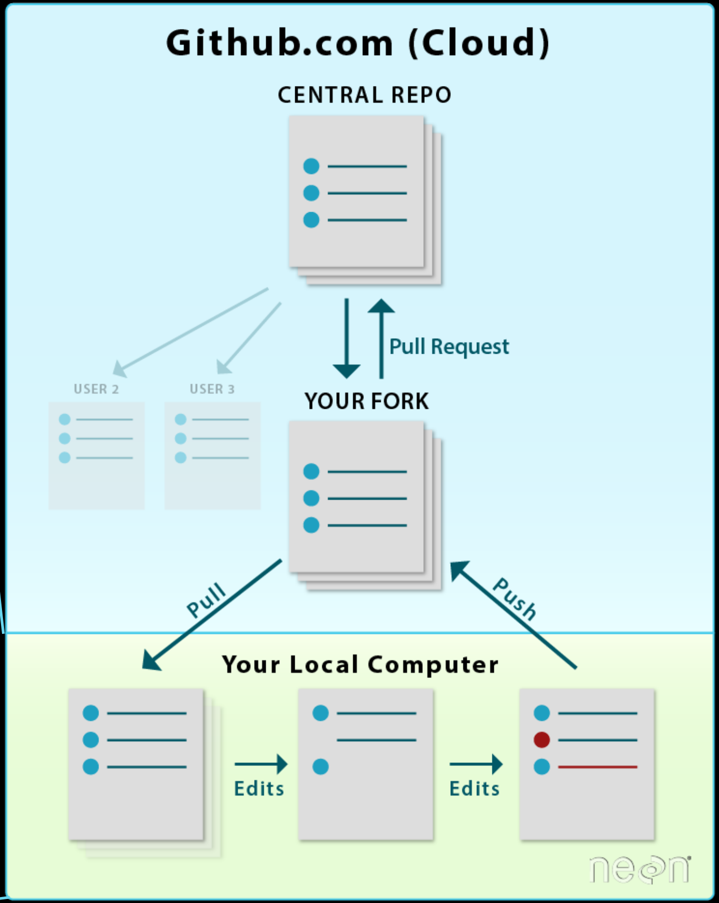
3. Version control

- ✓ keep master/main branch clean; develop in other branches
- ✓ small, incremental, commits
- ✓ ***.gitignore* sensitive and large files** – think about data storage
- ✓ pull first, push after



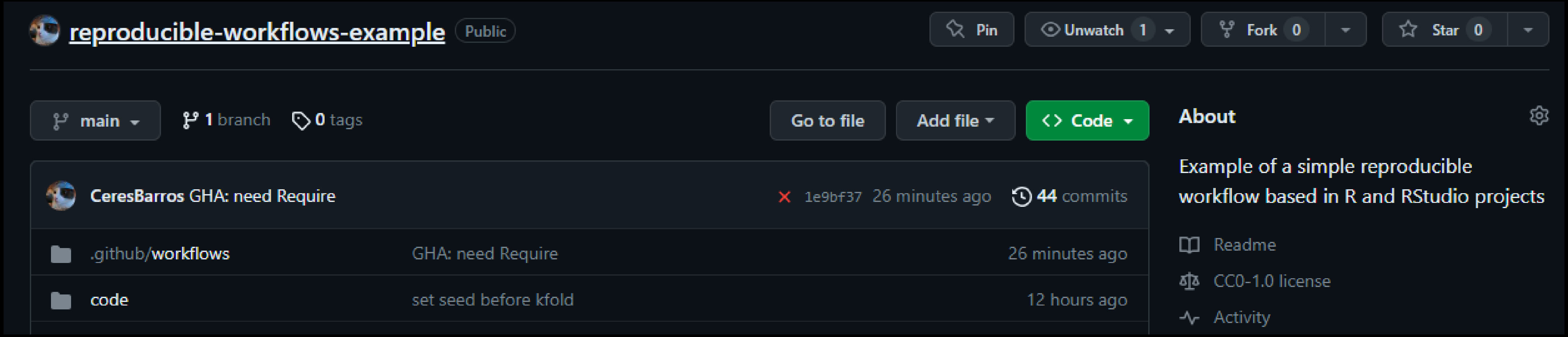
Staged	Status	Path
□	🟨 🟨	.gitignore
□	🟨 🟨	reproducible-workflows-example.Rproj

	Name	Size	Modified
	..		
□	.gitignore	44 B	Jul 6, 2023, 4:05 PM
□	LICENSE.md	6.4 KB	Jul 6, 2023, 3:49 PM
□	README.md	33 B	Jul 6, 2023, 3:49 PM
□	reproducible-workflows-example.Rproj	250 B	Jul 6, 2023, 4:05 PM



Source: https://www.earthdatascience.org/workshops/intro-version-control-git/pull-request/

3. Example of a reproducible workflow in R, RStudio and GitHub



reproducible-workflows-example Public

CeresBarros GHA: need Require ✕ 1e9bf37 26 minutes ago 44 commits

.github/workflows	GHA: need Require	26 minutes ago
code	set seed before kfold	12 hours ago

About

Example of a simple reproducible workflow based in R and RStudio projects

https://github.com/CeresBarros/reproducible-workflows-example

Useful resources

Peer-reviewed:

- Barros, C., Luo, Y., Chubaty, A.M., Eddy, I.M.S., Micheletti, T., Boisvenue, C., *et al.* (2023). Empowering ecological modellers with a PERFICT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation. *Methods Ecol Evol*, 14, 173–188.
- Braga, P.H.P., Hébert, K., Hudgins, E.J., Scott, E.R., Edwards, B.P.M., Sánchez Reyes, L.L., *et al.* (2023). Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution. *Methods in Ecology and Evolution*, 14, 1364–1380.
- Brousil, M.R., Filazzola, A., Meyer, M.F., Sharma, S. & Hampton, S.E. (2023). Improving ecological data science with workflow management software. *Methods in Ecology and Evolution*, 14, 1381–1388.
- Ellison, A.M. (2010). Repeatability and transparency in ecological research. *Ecology*, 91, 2536–2539.
- McIntire, E.J.B., Chubaty, A., Cumming, S., Andison, D., Barros, C., Boisvenue, C., *et al.* (2022). PERFICT: a Re-imagined Foundation for Predictive Ecology. *Ecology Letters*.
- Thomas, R.Q., Boettiger, C., Carey, C.C., Dietze, M.C., Johnson, L.R., Kenney, M.A., *et al.* (2023). The NEON Ecological Forecasting Challenge. *Frontiers in Ecology and the Environment*, 21, 112–113.
- Wilkinson, M.D., Dumontier, M., Aalbersberg, Ij.J., Appleton, G., Axton, M., Baak, A., *et al.* (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*, 3, 160018.

Reproducible workflows:

- Ecological Forecasting Initiative. (2020). *Reproducible Forecasting Workflows*. *Ecological Forecasting Initiative*. Available at: https://ecoforecast.org/reproducible-forecasting-workflows/. Last accessed 6 July 2023.
- **The Practice of Reproducible Research (**http://www.practicereproducibleresearch.org/)
- **R Markdown: The Definite Guide** (https://bookdown.org/yihui/rmarkdown/)
- **R Markdown cheat sheets** (https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf)
- **GitHub Quickstart** (https://docs.github.com/en/get-started/quickstart/hello-world)

Software:

RStudio

R

GitKraken

Git

References

GIFs/Images:

- https://www.reddit.com/r/gifs/comments/4a3exq/cat_typing_a_document_on_laptop/
- https://en.wikipedia.org/wiki/One_Ring#/media/File:One_Ring_Blender_Render.png
- All icons designed by Freepik and downloaded from Flaticon.com

Literature

- Barros, C., Luo, Y., Chubaty, A.M., Eddy, I.M.S., Micheletti, T., Boisvenue, C., *et al.* (2023). Empowering ecological modellers with a PERFICT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation. *Methods Ecol Evol*, 14, 173–188.
- Braga, P.H.P., Hébert, K., Hudgins, E.J., Scott, E.R., Edwards, B.P.M., Sánchez Reyes, L.L., *et al.* (2023). Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution. *Methods in Ecology and Evolution*, 14, 1364–1380.
- Ecological Forecasting Initiative. (2020). *Reproducible Forecasting Workflows*. *Ecological Forecasting Initiative*. Available at: https://ecoforecast.org/reproducible-forecasting-workflows/. Last accessed 6 July 2023.
- McIntire, E.J.B., Chubaty, A., Cumming, S., Andison, D., Barros, C., Boisvenue, C., *et al.* (2022). PERFICT: a Re-imagined Foundation for Predictive Ecology. *Ecology Letters*.
- Thomas, R.Q., Boettiger, C., Carey, C.C., Dietze, M.C., Johnson, L.R., Kenney, M.A., *et al.* (2023). The NEON Ecological Forecasting Challenge. *Frontiers in Ecology and the Environment*, 21, 112–113.
- Wilkinson, M.D., Dumontier, M., Aalbersberg, Ij.J., Appleton, G., Axton, M., Baak, A., *et al.* (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*, 3, 160018.