



# Introduction to Git and GitHub with RStudio

Julien Martin  
17 January, 2024

# Learning expectations

This won't even touch the surface of git... But! Hopefully, you:

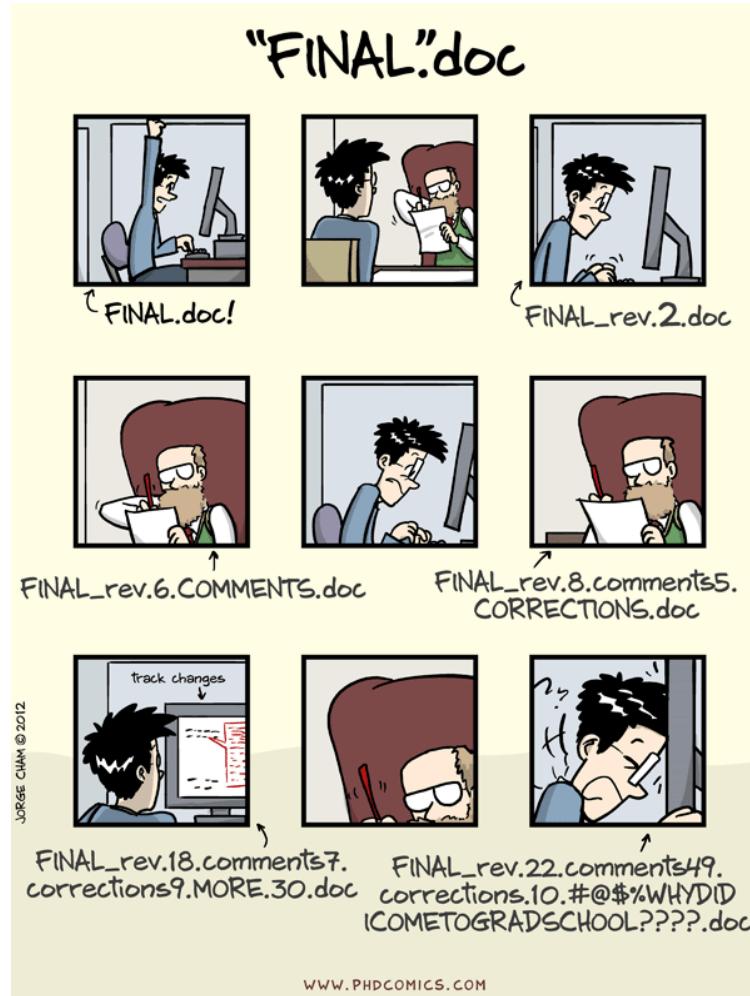
- Recognize the power of using version control
- Know the basic tools to get started using git
- Where to go for help

# Getting started

You need:

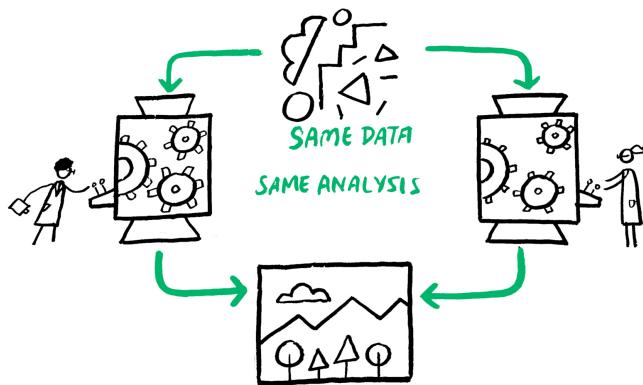
- RStudio or VS Code
- R
- a github account
- git software
- R packages:
  - usethis
  - gitcreds

# Rings a bell?

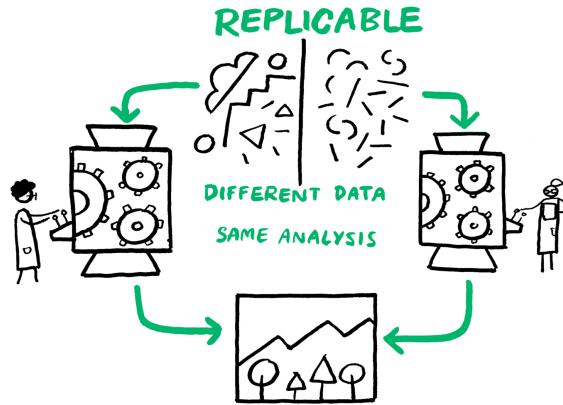


# Is your workflow...

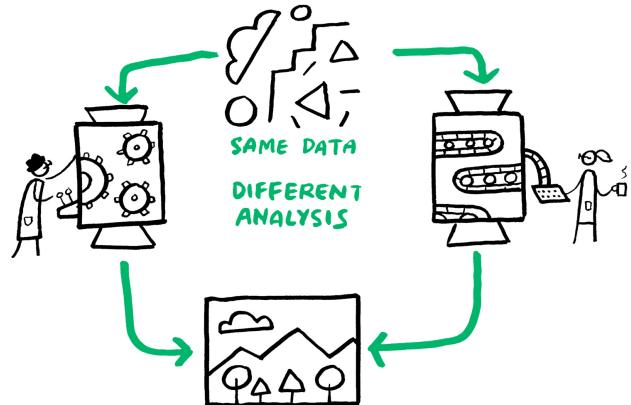
**REPRODUCIBLE**



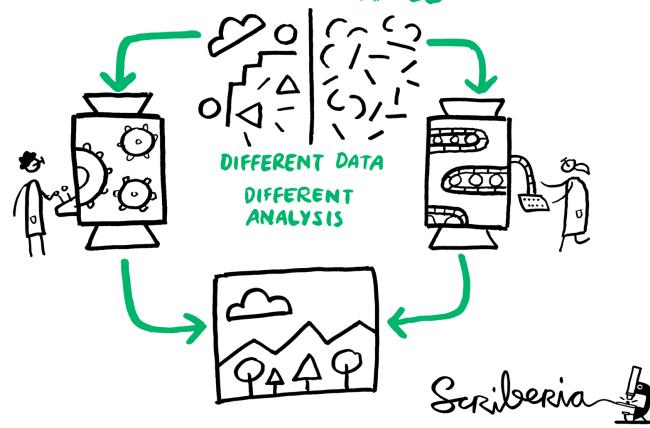
**REPLICABLE**



**ROBUST**



**GENERALISABLE**



# Why reproducible science?

The screenshot shows the header of the Nature journal website. The word "nature" is written in a large serif font, with "International weekly journal of science" in a smaller sans-serif font below it. A navigation bar below the header includes links for Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, Audio & Video, and For Authors. Below the navigation bar is a breadcrumb trail: Archive > Specials and supplements archive > Challenges in irreproducible research.

## SPECIAL

See all specials

### CHALLENGES IN IRREPRODUCIBLE RESEARCH

Science moves forward by corroboration – when researchers verify others' results. Science advances faster when people waste less time pursuing false leads. No research paper can ever be considered to be the final word, but there are too many that do not stand up to further study.

There is growing alarm about results that cannot be reproduced. Explanations include increased levels of scrutiny, complexity of experiments and statistics, and pressures on researchers. Journals, scientists, institutions and funders all have a part in tackling reproducibility. *Nature* has taken substantive steps to improve the transparency and robustness in what we publish, and to promote

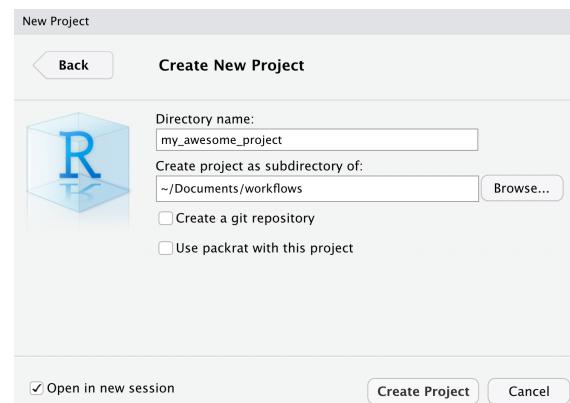
# RStudio Projects

Use RStudio projects to keep materials associated with a particular analysis together

- Self contained and portable
- Working directory set to root of project on launch
- Fresh session everytime the project is launched

See Jenny Bryan's post on [project oriented workflows](#) for more details

File > New Project > New Directory



# Alternatives to RStudio Projects

So You don't like RStudio 

Essentially a **Rstudio Project**:

- organise all the files within a given project folder
- set project folder as *working directory*
- ensure a fresh, clean session

simply, do the same without using RStudio 😍

# Version Control

What is it? 🤔

The **management of changes** to documents, computer programs, large web sites and other collections of information.

Git  git

Open source (free to use) **Version control software**.

GitHub 

A **website** (<https://github.com/>) that allows you to **store your Git repositories online** and makes it easy to collaborate with others.

# Getting started with Git and GitHub

## 1. Create an account on GitHub

- You set your username, email and password here

## 2. install git on your computer

- Windows: <https://gitforwindows.org/>
- MacOS: need Xcode tools.
  - In a shell/terminal: `xcode-select --install`
- Linux:
  - debian-based: `sudo apt install git`
  - Fedora, RedHat: `dnf install git`

## 1. You might need to restart your computer

For more details go to <https://git-scm.com/downloads>

# Git, Github & IDE

Before: git only through the terminal 😢

---

Now: git can be used via multiple user-friendly interfaces

RStudio + usethis 📦 + gitcreds 📦 == ❤️ Git & GitHub 😁

When using VS Code



# Configure git & GitHub

to be done only once

## Configure git

To check your configuration

```
usethis::git_sitrep()
```

Set your configuration

Use your github username and and the email you used to sign-up on GitHub

```
usethis::use_git_config(  
  user.name = "JulienGAMartin",  
  user.email = "julien.martin@uottawa.ca")
```

# Configure GitHub authentication

## Get GITHUB Personal Authorisation Token

```
usethis::create_github_token()
```

Need scopes *repo, users, workflows* at least

Settings / Developer settings

OAuth Apps

GitHub Apps

Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Token description

R:GITHUB\_PAT

What's this token for?

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

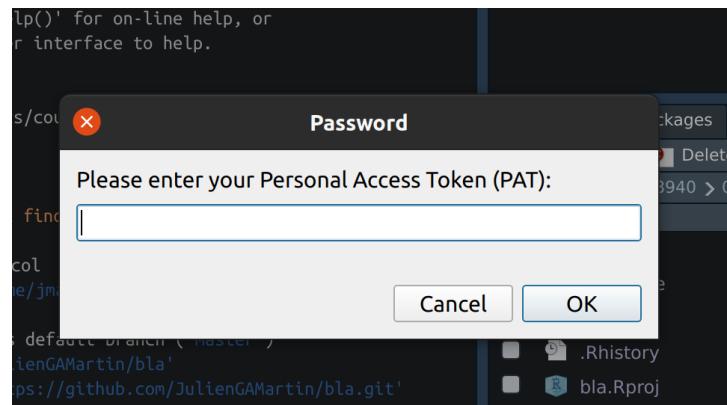
|   |                                      |
|---|--------------------------------------|
| <input checked="" type="checkbox"/> repo            | Full control of private repositories |
| <input checked="" type="checkbox"/> repo:status     | Access commit status                 |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status             |
| <input checked="" type="checkbox"/> public_repo     | Access public repositories           |
| <input checked="" type="checkbox"/> repo:invite     | Access repository invitations        |

# Configure GitHub authentication

Store your git token with credentials

```
gitcreds::gitcreds_set()
```

Provide your github PAT token in the window



Restart RStudio and R

**Git and github access should be configured now**

# New project using GitHub and Rstudio

Might be problematic the first time you do it

1. create R project
2. initialise it for git

```
usethis::use_git()
```

3. Accept the commit and restart R
4. In R, to create github repo and link it

```
usethis::use_github()  
usethis::git_vaccinate()
```

5. Check on github that the repos is there

**et voila**

# New project using GitHub and NOT Rstudio

## Option 1

1. create new empty folder for your project(no space, no accent)
2. add at least one file in it (suggest README.md)
3. using terminal/shell

```
git init  
git add .  
git commit -a -m "first commit"
```

4. In R, to create github repo and link it

```
usethis::use_github()  
usethis::git_vaccinate()
```

5. Check on github that the repos is there

**et voila**

# New project using GitHub and NOT Rstudio

## Option 2

1. Create new repository on github
2. Set R working directory where you want to save your project
3. To clone (and link) github repo to your computer

```
usethis::create_from_github(  
  "https://github.com/your_github/your_repo.git",  
  destdir=".")
```

4. Check that your folder has been created where you want it

**et voila**

# Git panel

Integrated graphical user interface

**Version Control** with Git or SVN

Turn on at **Tools > Project Options > Git/SVN**

The screenshot shows a software interface titled "Version Control with Git or SVN". A toolbar at the top includes icons for Stage files, Show file diff, Commit, Push/Pull staged files to remote, and View History. Below the toolbar is a status bar with tabs for Environment, History, and Git, and buttons for Diff, Commit, Staged, Status, Path, and a dropdown for master. A context menu is open over a file named "file-with-changes.R", listing options: Revert..., Ignore..., and Shell... A tooltip says "Open shell to type commands". The current branch is indicated as "current branch".

Stage files:

- A Added
- D Deleted
- M Modified
- R Renamed
- ? Untracked

Show file diff

Commit

Push/Pull staged files to remote

View History

Environment History Git

Diff Commit

Staged Status Path

file-with-changes.R

Revert... Ignore... Shell...

master

current branch

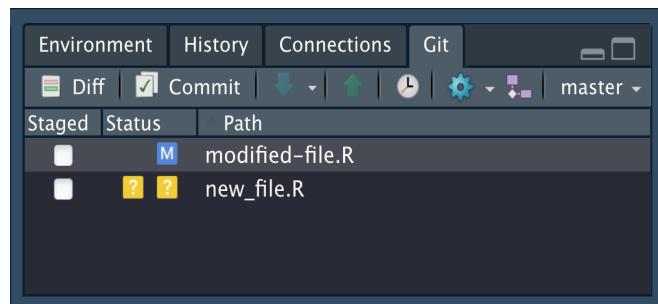
Open shell to type commands

# Git terms

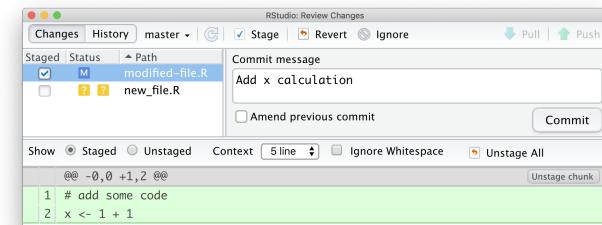
- **repository** your project folder
- **commit** a snapshot of your repo
- **push** send commits to a remote
- **pull** get commits from a remote
- **clone** get the repository from the remote for the first time
- **branch** a movable label that points to a commit
- **merge** combining two branches
- **remote** a computer or server with the repository on it

# Git RStudio workflow

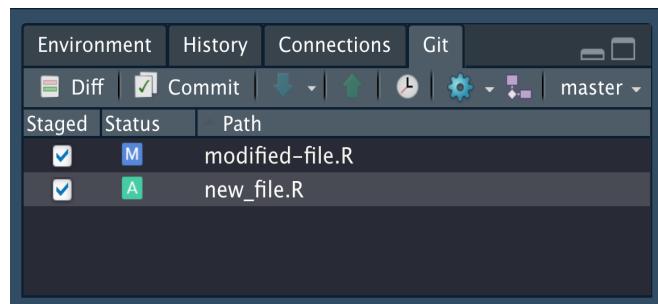
## 1.view file status



## 3.commit changes



## 2.stage files

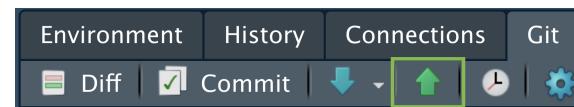


# Share on GitHub

## Create repo

The screenshot shows a GitHub repository page. At the top, there's a header with navigation links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, the repository name is 'JulienGAMartin / Bio8940'. On the left, there's a sidebar with options for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows a commit history with one commit from 'JulienGAMartin' titled 'Initial commit'. It also lists files: LICENSE and README.md, both with 'Initial commit' status. Below the files, there's a 'README.md' editor with the text 'Bio8940'. To the right, there are sections for 'About', 'Releases', and 'Packages', each with a 'Create a new release' or 'Publish your first package' button. At the bottom, there's a footer with links to GitHub's terms, privacy, security, status, docs, contact, pricing, API, training, blog, and about pages.

## Push further changes



# Anatomy of a GitHub repo

- **README**. Explain what your project is, and how to use it.
  - `usethis::use_readme_md()`
  - `usethis::use_readme_rmd()`
- **LICENSE**. Without a licence, the contents of the repository are technically closed.
  - Examples licence **CC-by**: `usethis::use_ccby_license(name = "Julien Martin")`
  - `?licenses`: details of functions available to generate licenses
  - <https://choosealicense.com/> help on choosing a licence.
- **CONTRIBUTING.md** - guidelines for contributors.
  - `usethis::use_tidy_contributing()` provides a relatively strict but instructive template
- **CODE\_OF\_CONDUCT.md** set the tone for discourse between contributors.
  - `use_code_of_conduct()`

# GitHub issues

use GitHub issues to plan, record and discuss tasks.

The screenshot shows the GitHub Issues page for the rstudio/rstudio repository. The top navigation bar includes links for Code, Issues (1.9k), Pull requests, Actions, Projects (5), Wiki, Security, and Insights. On the right, there are buttons for Watch (252), Star, and a search icon. Below the navigation is a callout box with a hand icon and the text "Want to contribute to rstudio/rstudio? If you have a bug or an idea, read the [contributing guidelines](#) before opening an issue." The main area features a search bar with the query "is:issue is:open". Below the search bar are filters for Labels (126), Milestones (4), and a "New issue" button. A summary shows 1,863 Open issues and 3,147 Closed issues. The issues are listed in a grid, each with a title, status, labels, and a small profile picture. The first few issues are:

- show Stop button when running long-running Python computation (enhancement, python)
- support Python installations installed via pyenv (enhancement, python)
- Undo destroy .rmd document
- Low resolution plots with UI zoom
- "Hand pointer" mouse cursor in RStudio does not follow system theme and it is much smaller than normal size (backlog, bug)
- SQL chunks do not respect option 'max.print'.
- Step in/out buttons disappeared and keyboard shortcuts don't work

# Why using a remote like GitHub ?

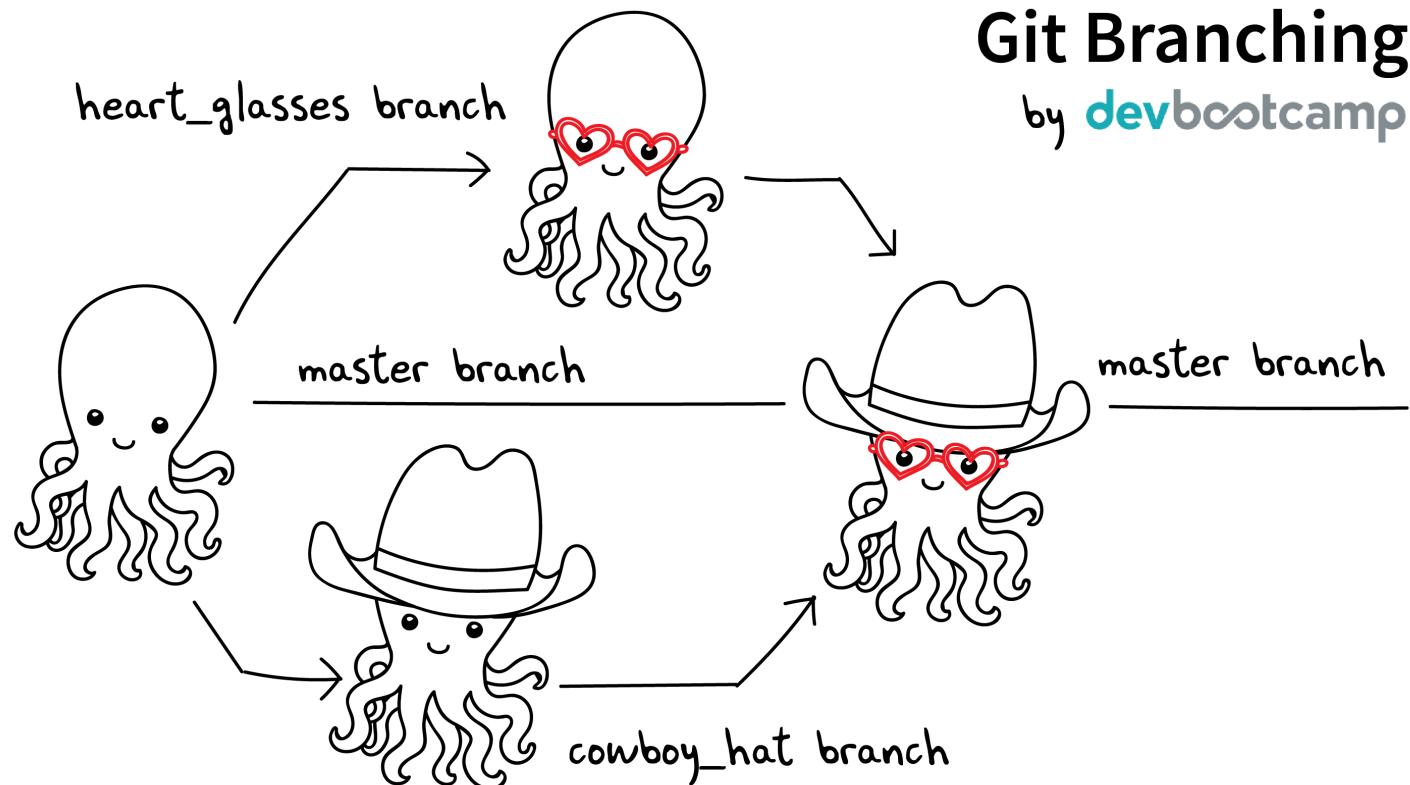
- A backup of your repository (**Dropbox** is **NOT** a backup)
  - Work with others (not covered)
  - Increase your visibility
  - Increase interactions with users
  - Easy distribution of R packages (without CRAN submissions)
- 

one more time just in case

**Dropbox** is **NOT** a backup

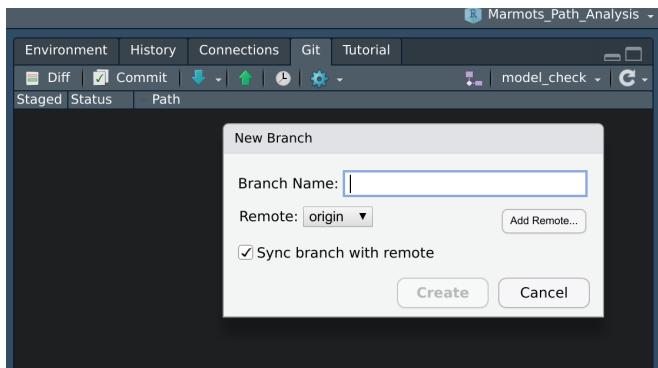
# Branching and merging

This the true Power of git



# Branching and merging

## Branching in Rstudio



## Pull request in Github

When you want to merge branches:

- create a pull request on Github
- check for incompatibilities
- then merge

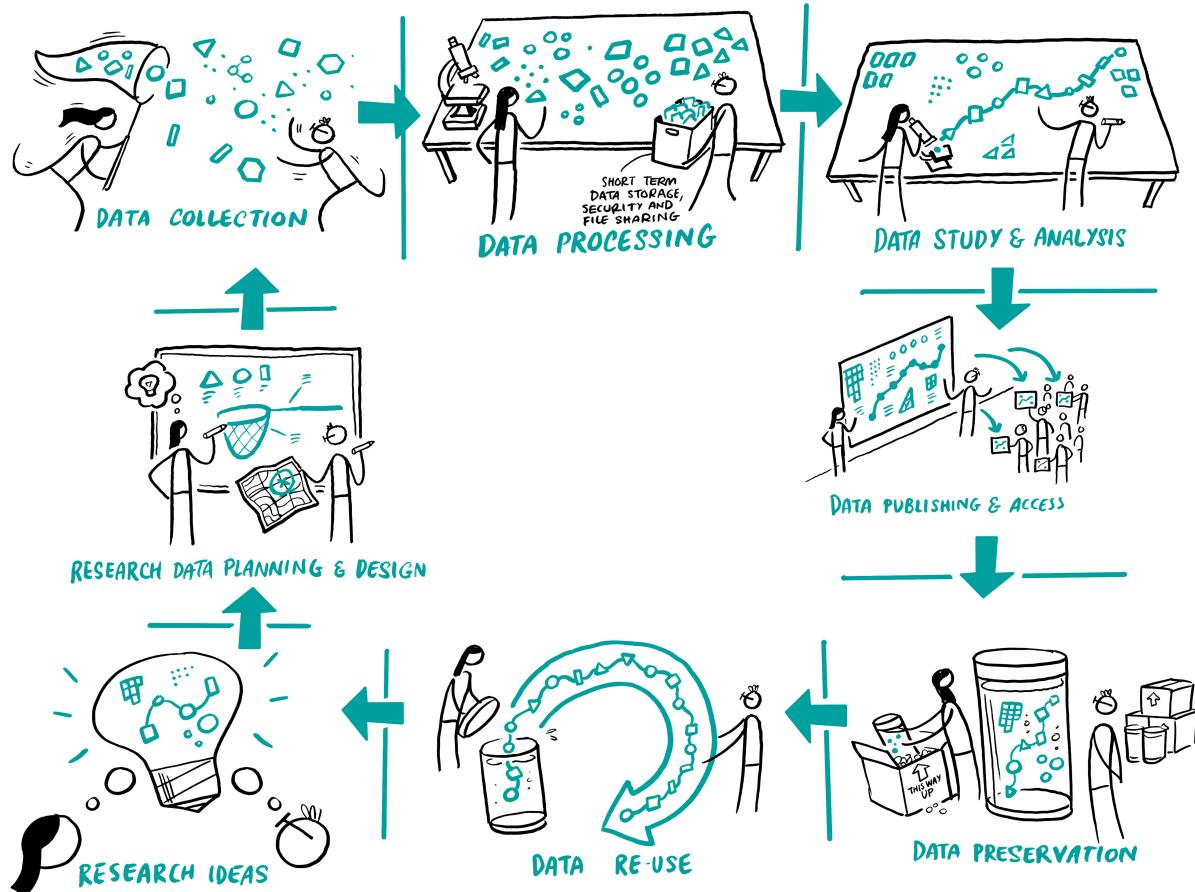
# Resources

I used many slides from [Olivier Gimenez]  
(<https://oliviergimenez.github.io>)

Check out Happy Git and GitHub for the useR for more joy with git

The British Ecological Society has A Guide to Reproducible Code in Ecology and Evolution

# Virtuous research cycle



Scriberia