Open Science and Research Software Engineering

Workshop
Center for Advanced Internet Studies (CAIS)

Quirin Würschinger

September 21, 2023

Introduction

> `whoami`

Quirin Würschinger
q.wuerschinger@lmu.de
Wissenschaftlicher Mitarbeiter and PostDoc in (computational) linguistics
LMU Munich

Current work

- **research**
  - lexical innovation on the web and in social networks
  - variation and change in language use and social polarization in social networks

  - using Large Language Models (LLMs) like ChatGPT for research in linguistics and social science.

- **teaching**: corpus linguistics and research methodology

Promoting Open Science in (computational) linguistics at LMU

- teaching and applying reproducible corpuslinguistic methods
- creating and sharing corpora among researchers and students

---

Workshop materials

GitHub repository https://github.com/wuqui/opensciws
slides https://wuqui.github.io/opensciws/opensciws_slides.html
website version https://wuqui.github.io/opensciws/opensciws_website.html

---

Open Open Science workshop

Focus on …

- ask questions
- discuss
- apply and practice
- collaborate

Time table

Topic	Start	End
Intro	09:00	09:30
Open Science principles	09:30	10:30
—	10:30	10:50
version control	10:50	11:10
project structure	11:10	12:00
data	12:00	12:30
—	12:30	13:30
code	13:30	14:00

Topic	Start	End
methods	14:00	14:30
authoring	14:30	15:15
—	15:15	15:30
publishing	15:30	16:00
open issues and recap	16:00	16:30

Addressing different backgrounds and goals

Backgrounds and interests

CAIS: Forschung zu Digitalisierung und Digitale Gesellschaft

research fiels

- education and pedagogy
- political science
- sociology
- communications studies
- …

data and methods

- qualitative interviews
- text analysis
- quantitative surveys
- experimental designs
- social media studies
- …

Survey: main interests

- reproducible workflows
  - managing files and folders
  - plain text authoring
  - programming with Python and R
- methods
  - quantitative approaches
  - text analysis
  - questionnaires
- publishing
  - authoring papers
  - sharing data and code

Who are you?

Please briefly introduce yourself …

1. name
2. place and position
3. your research interest in about 3 sentences for someone outside your field

---

Open Science principles

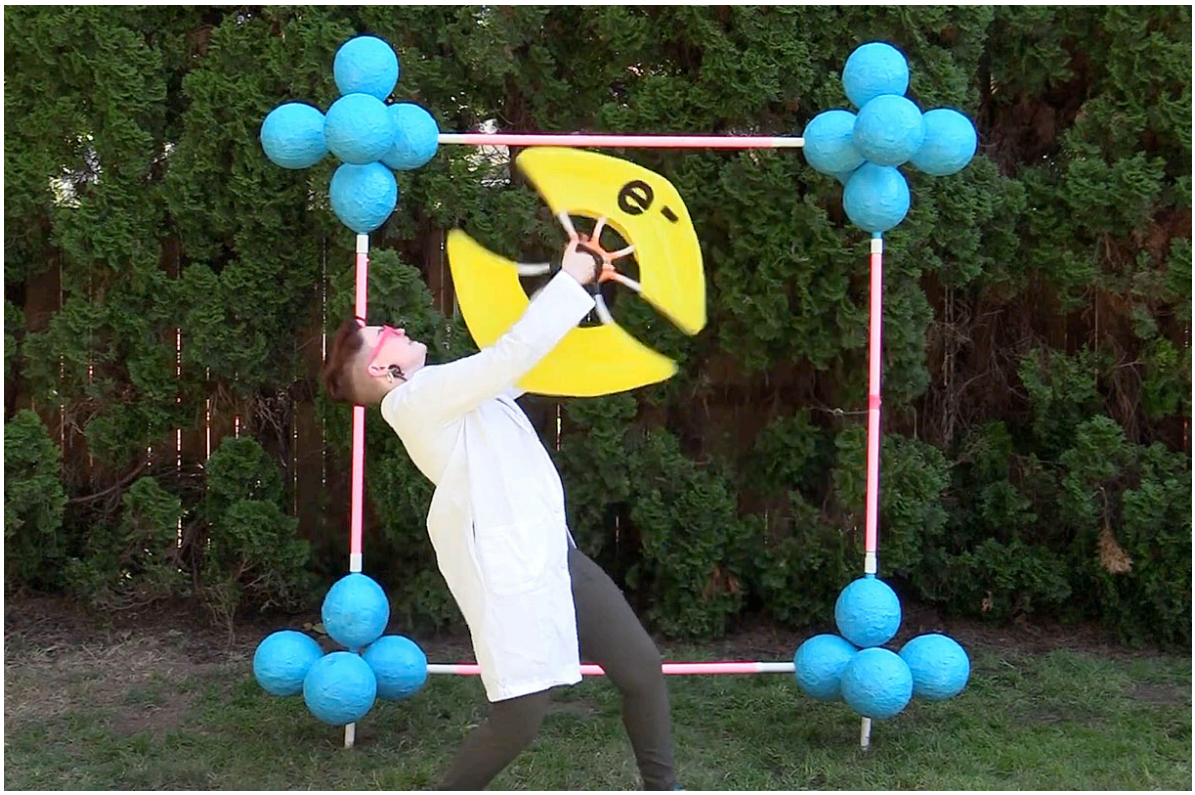
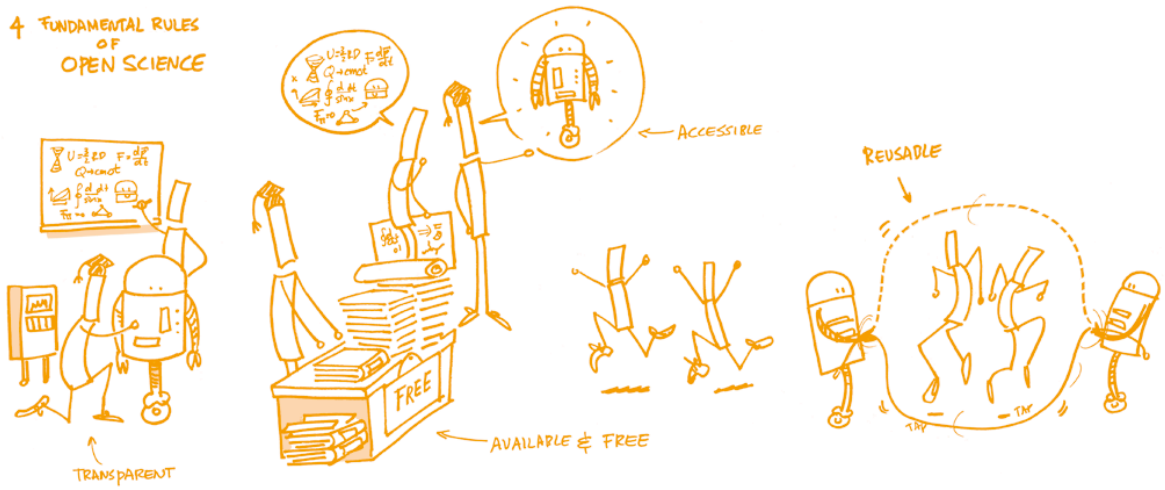


Figure 1: Dance your PhD

What is Open Science?



Why should we do Open Science?

- dataset/sample size
- effect sizes
- selection/number of relationships
- flexibility in design
- financial interests
- hype around topic/field

What are the reasons why science can go wrong?

Figure 2: source

Principles of Open Science

Open Science lifecycle

Roles in Open Science

Funders make open science part of the selection process, and conditions for grantees conducting research.

Publishers make open science part of the review process, and conditions for articles published in their journals.

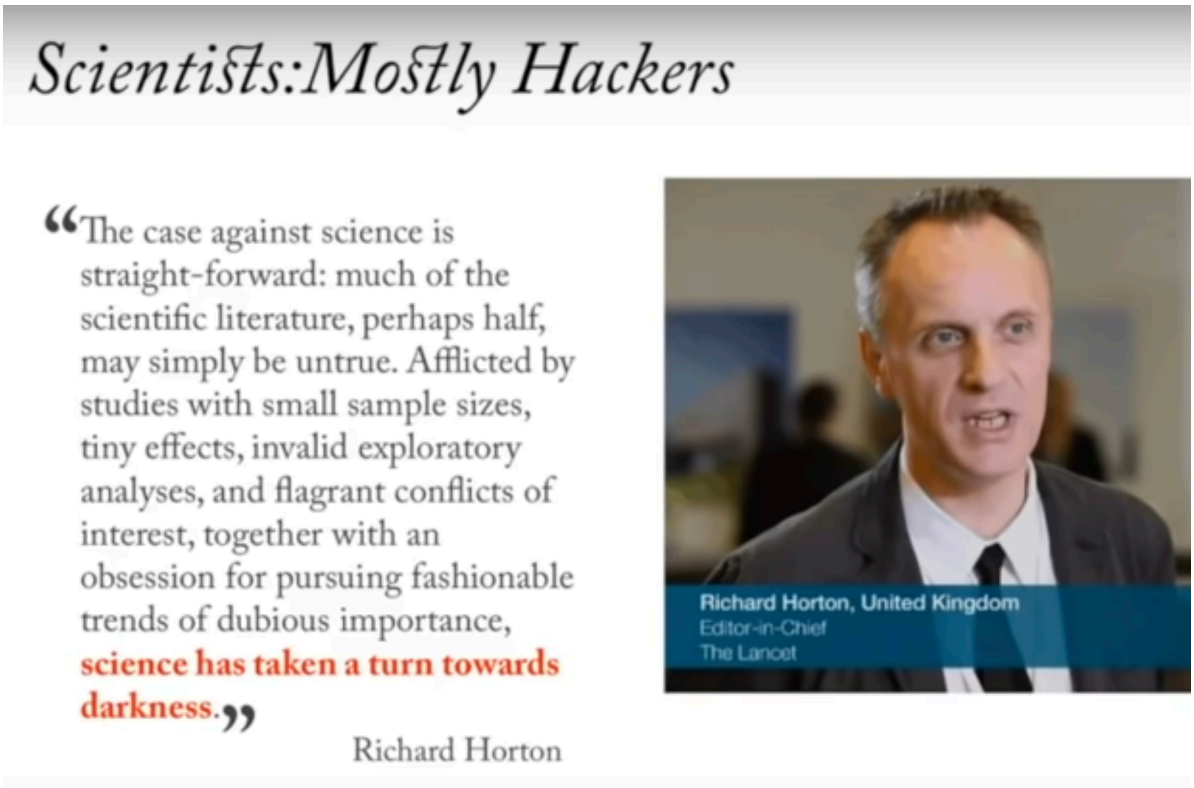


Figure 3: Richard McElreath: *Science as Amateur Software Development*

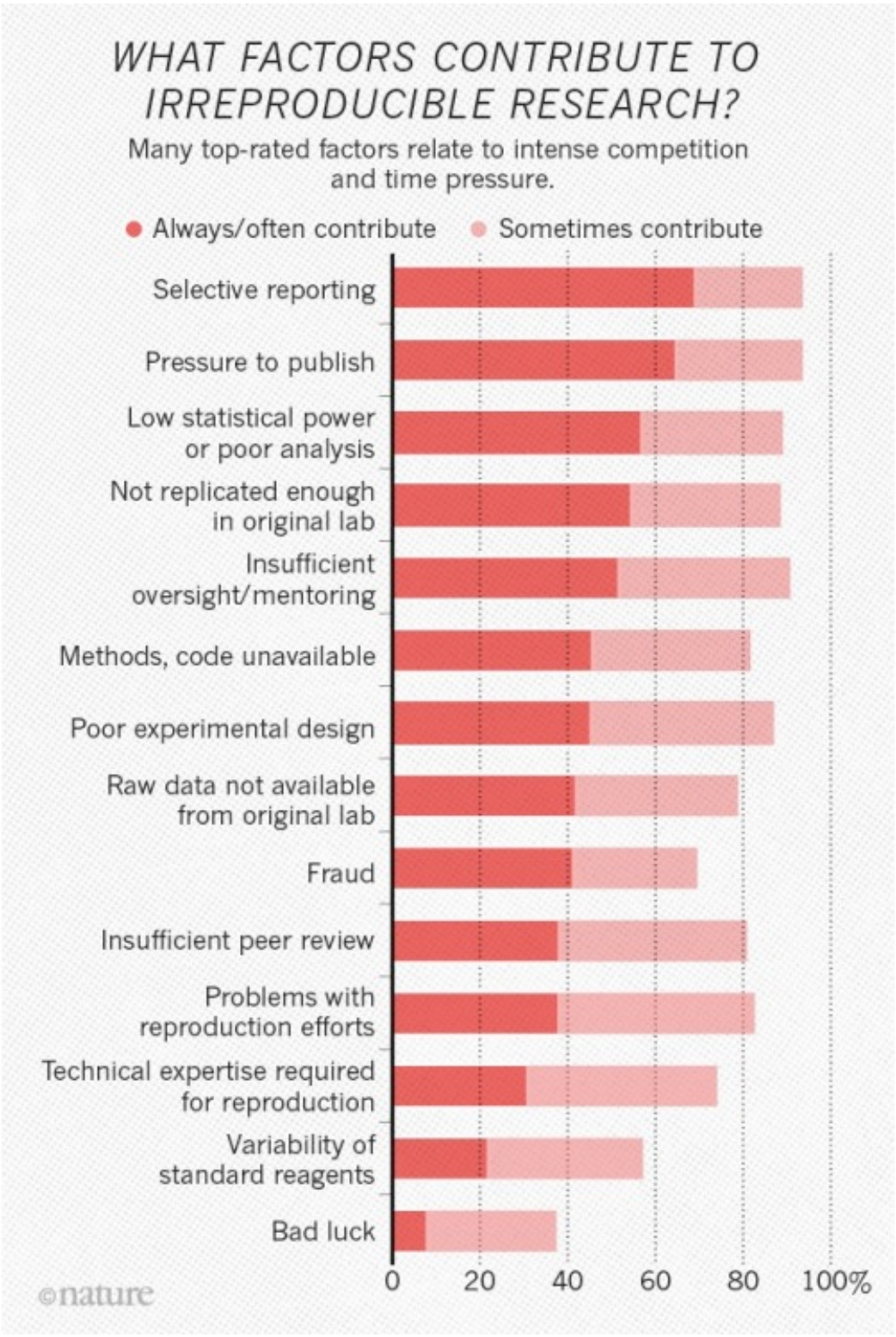


Figure 4: source

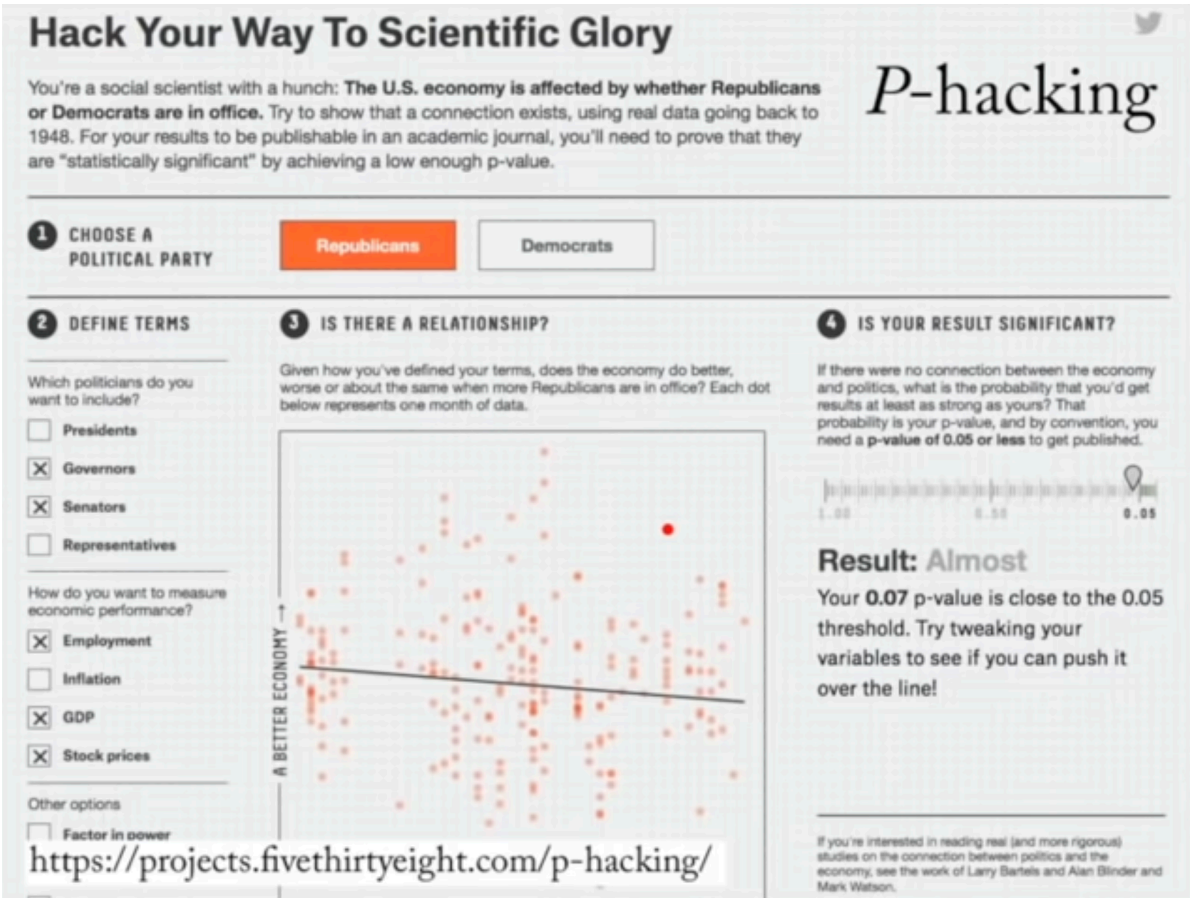


Figure 5: source

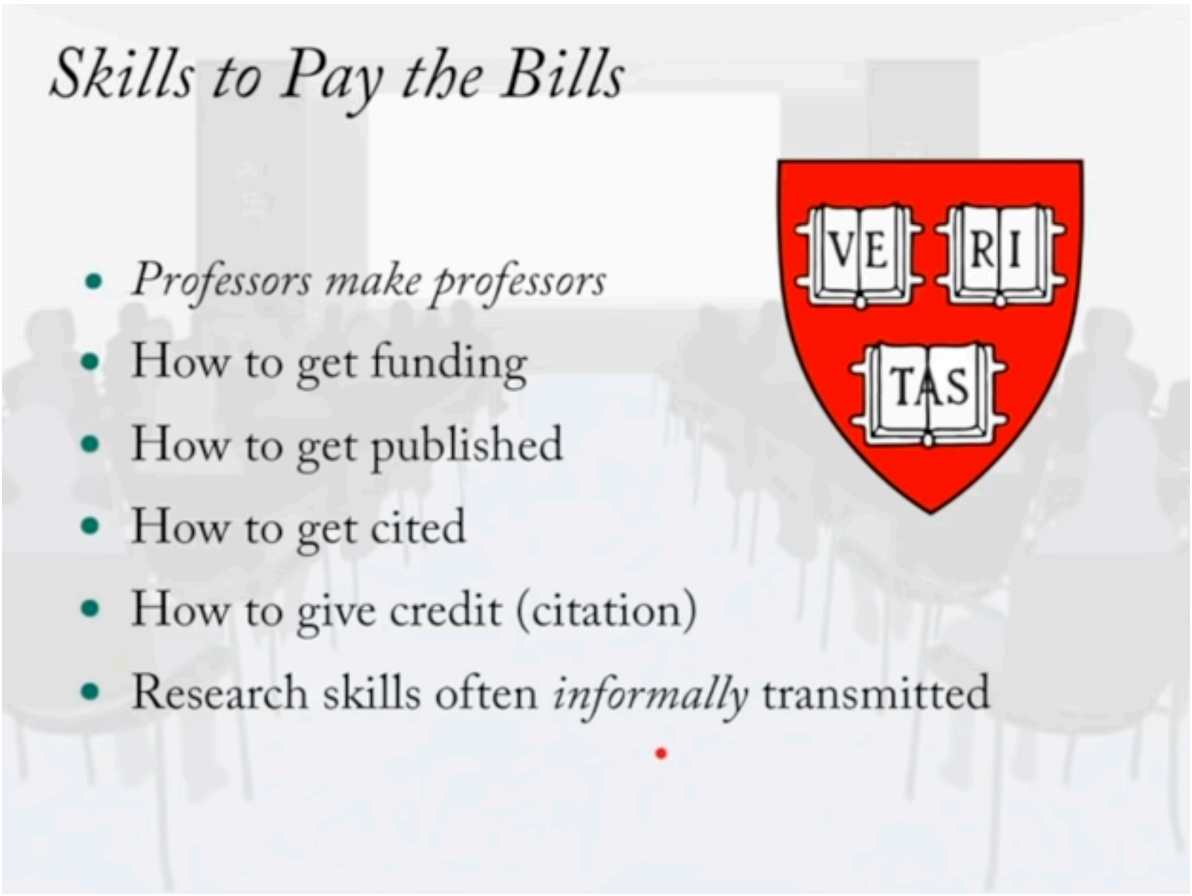
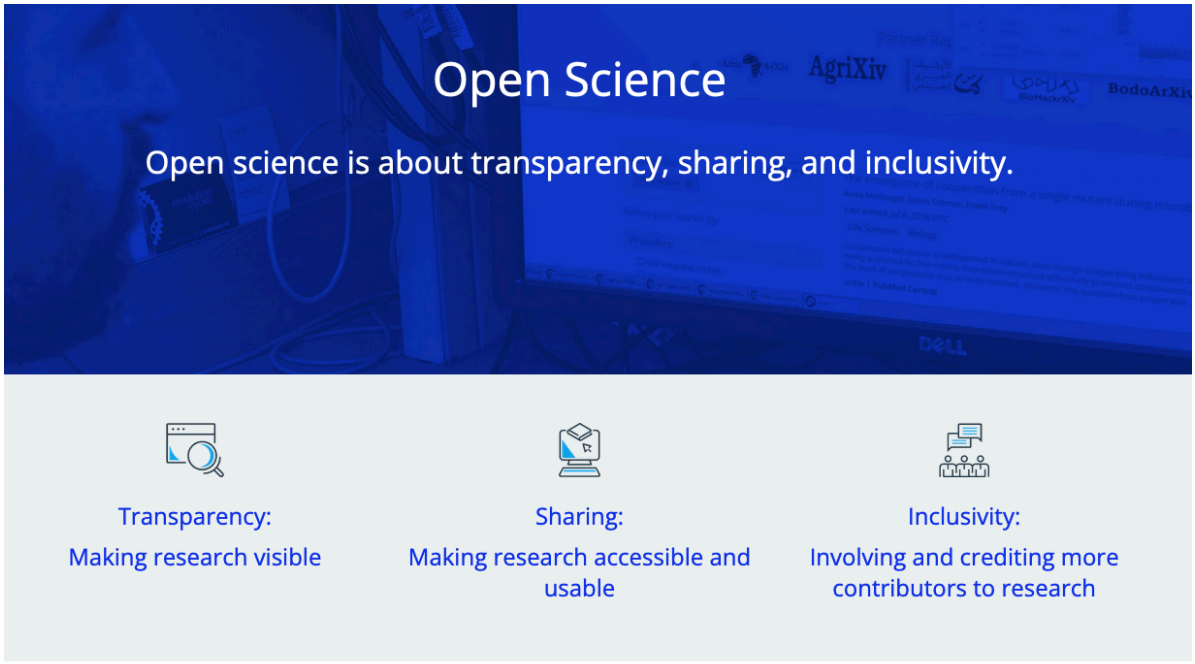


Figure 6: source



These principles aim to democratize access to research, promote equitable resource distribution, foster accountability and trustworthiness, accelerate self-correction, and improve rigor and reproducibility.

Figure 7: Center for Open Science

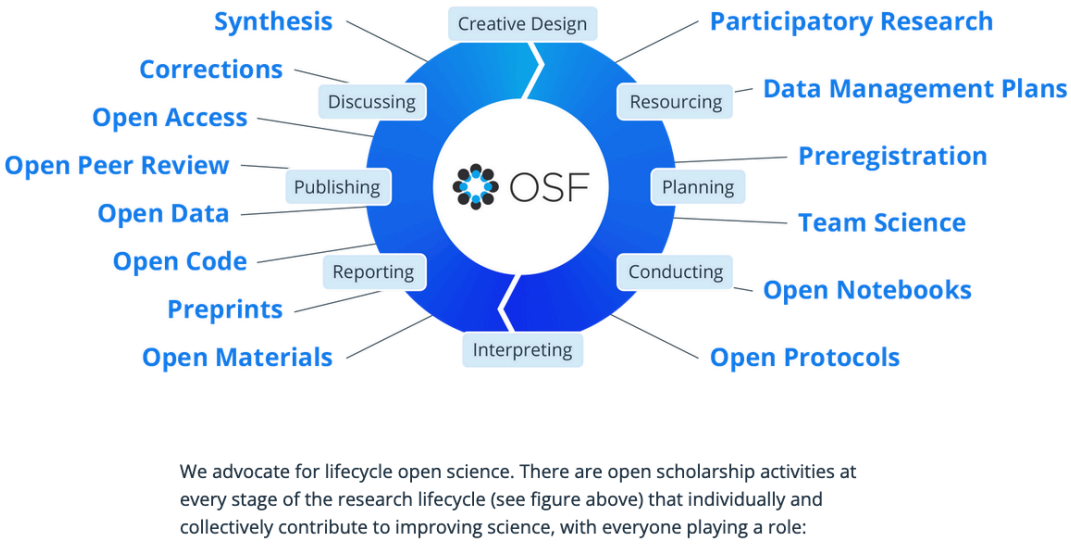


Figure 8: Center for Open Science

Institutions make open science part of academic training, and part of the selection process for research positions and evaluation for advancement and promotion.

Societies make open science part of their awards, events, and scholarly norms.

Researchers enact open science in their work and advocate for broader adoption in their communities.

[Center for Open Science]

Who profits from Open Science?

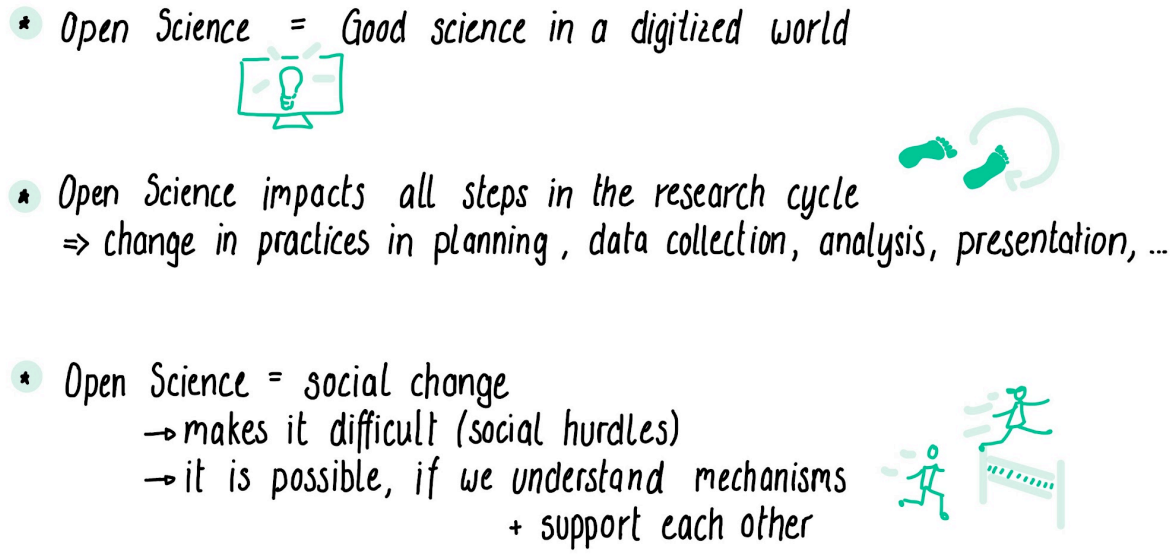
What is Open Science to you?

What do you find interesting, important, or attractive about Open Science?

https://tinyurl.com/opnsci

Figure 9: source

Learning outcomes



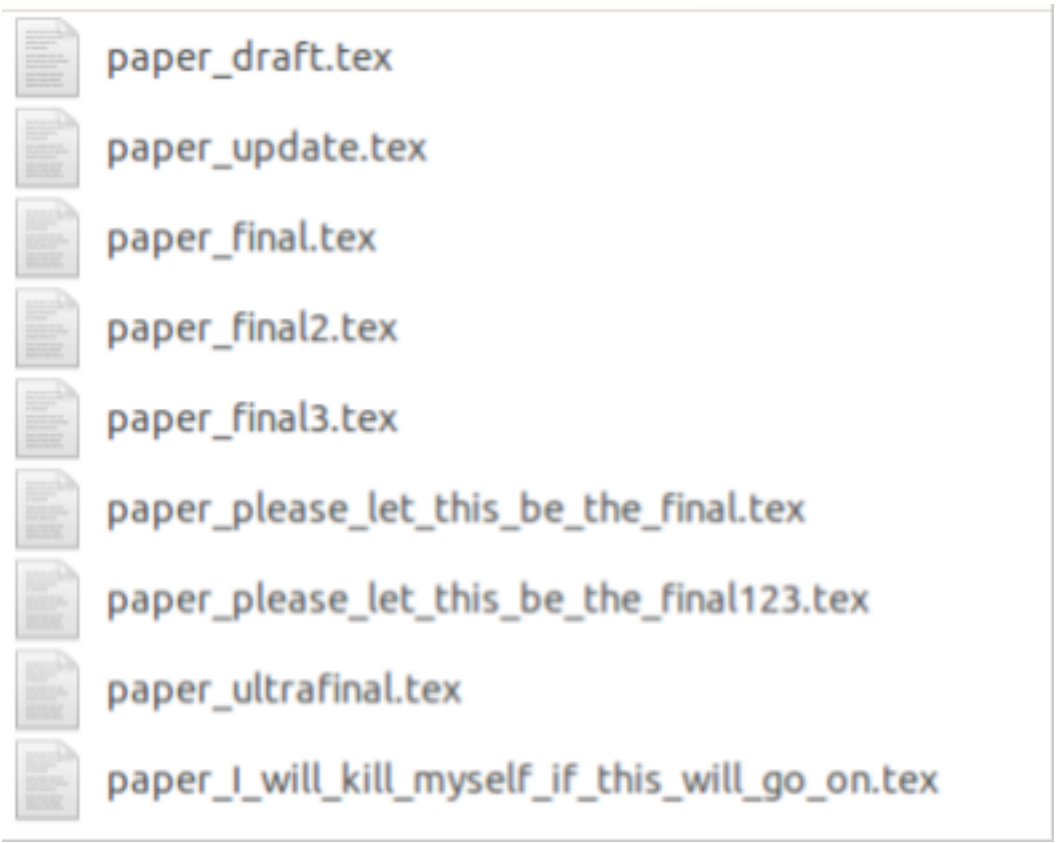
Implementing an open and reproducible workflow

1. version control
2. project structure
3. data
4. methods
5. code
6. authoring
7. publishing

Break

Version control

Why use version control?



Dear colleagues,

attached you find the first public version of the [redacted] protocol.
Please have a look and do comment. We can also meet to aggregate our reviews.

1 attachment: StudyProposa[redacted]_Validation_V1_250918docx.docx

source

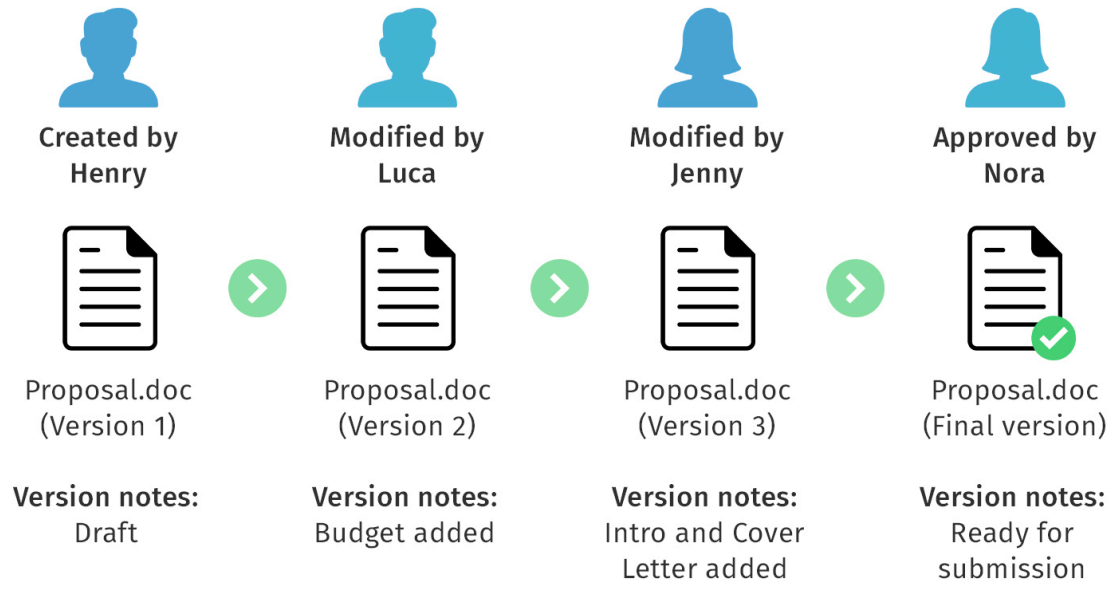


Figure 10: source

---

git and GitHub/GitLab

git software on your machine

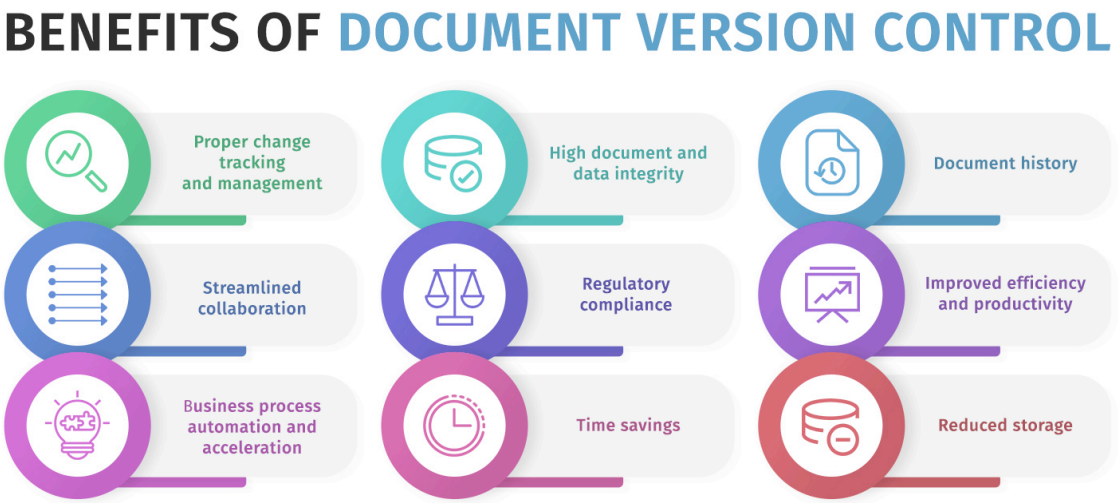


Figure 11: source

```
git add src/tests.py
git commit -m 'add tests'
git push
```

GitHub and GitLab services on a remote server

Collaborating using GitHub

git commands

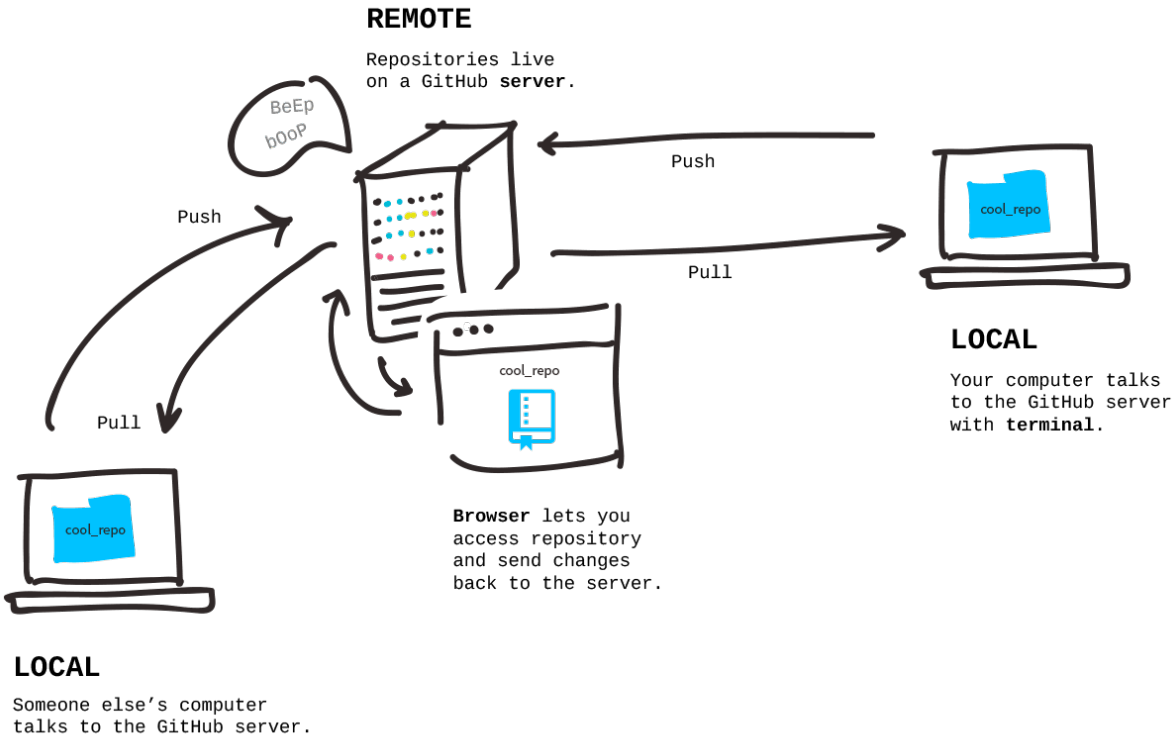


Figure 12: (source)

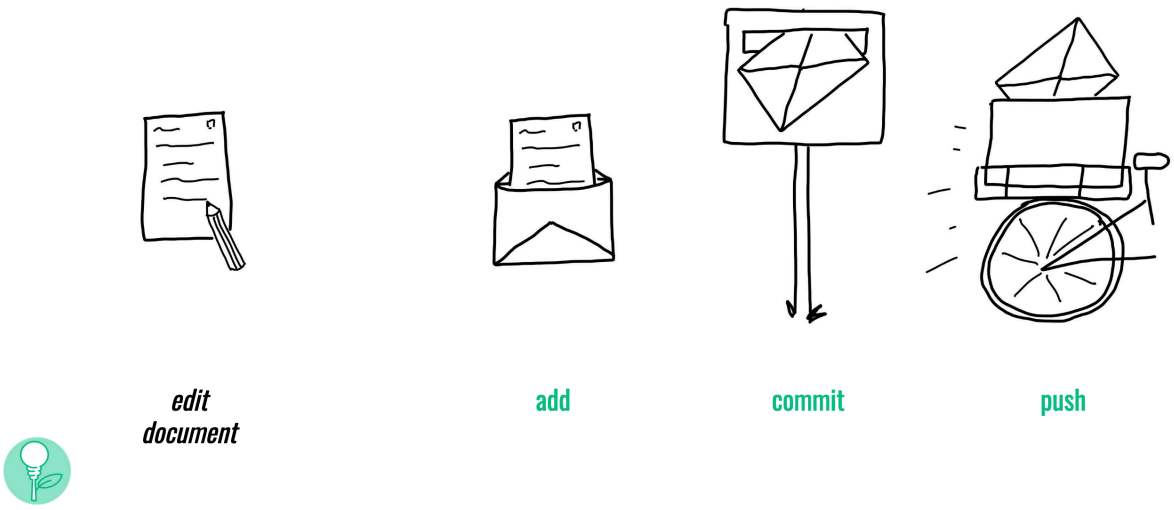


Figure 13: (source)

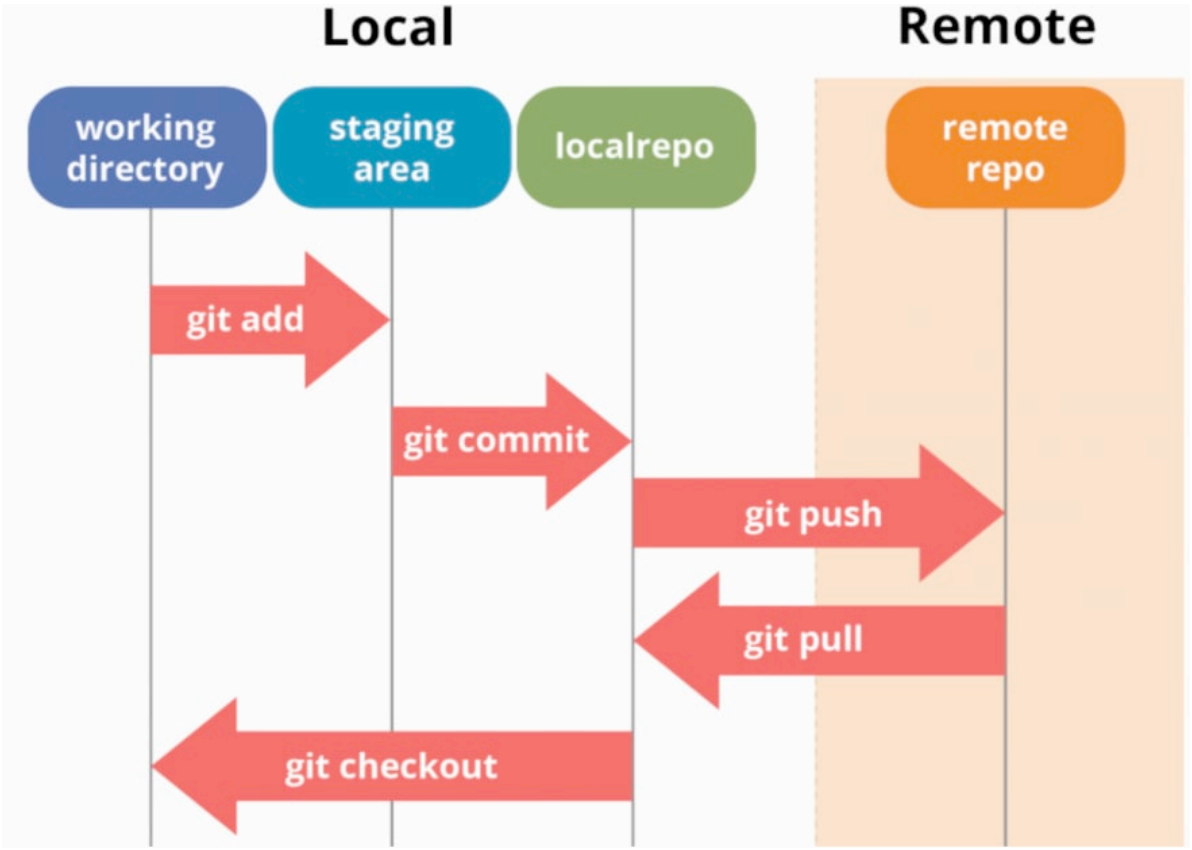


Figure 14: (source)

GitHub workflow

Example

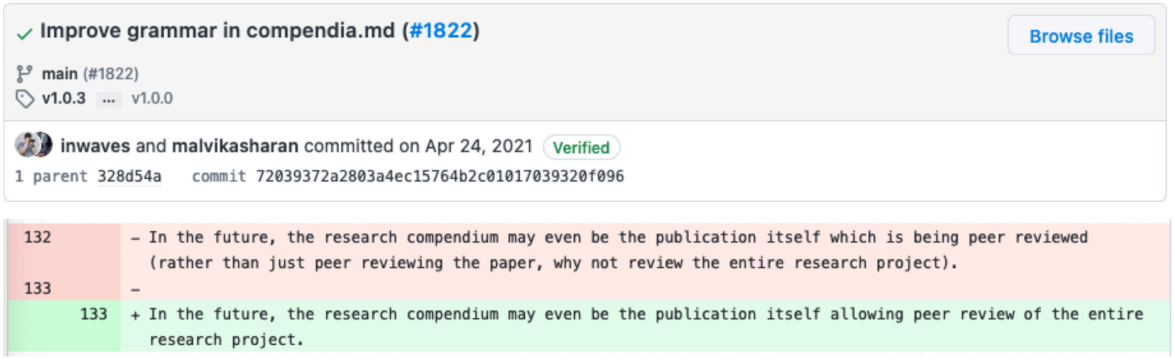


Figure 15: source

How to set up a GitHub repository

---

set up `git`

Installing `git`: see tutorial

Using `git`:

- from the command line
- using a standalone GUI[1] tool; e.g.:
  - GitKraken
  - GitHub Desktop
- from within your editor/IDE[2]; e.g.:

[1] Graphical User Interface
[2] Integrated Development Environment

- RStudio
- VSCode

---

set up `GitHub`

tutorial

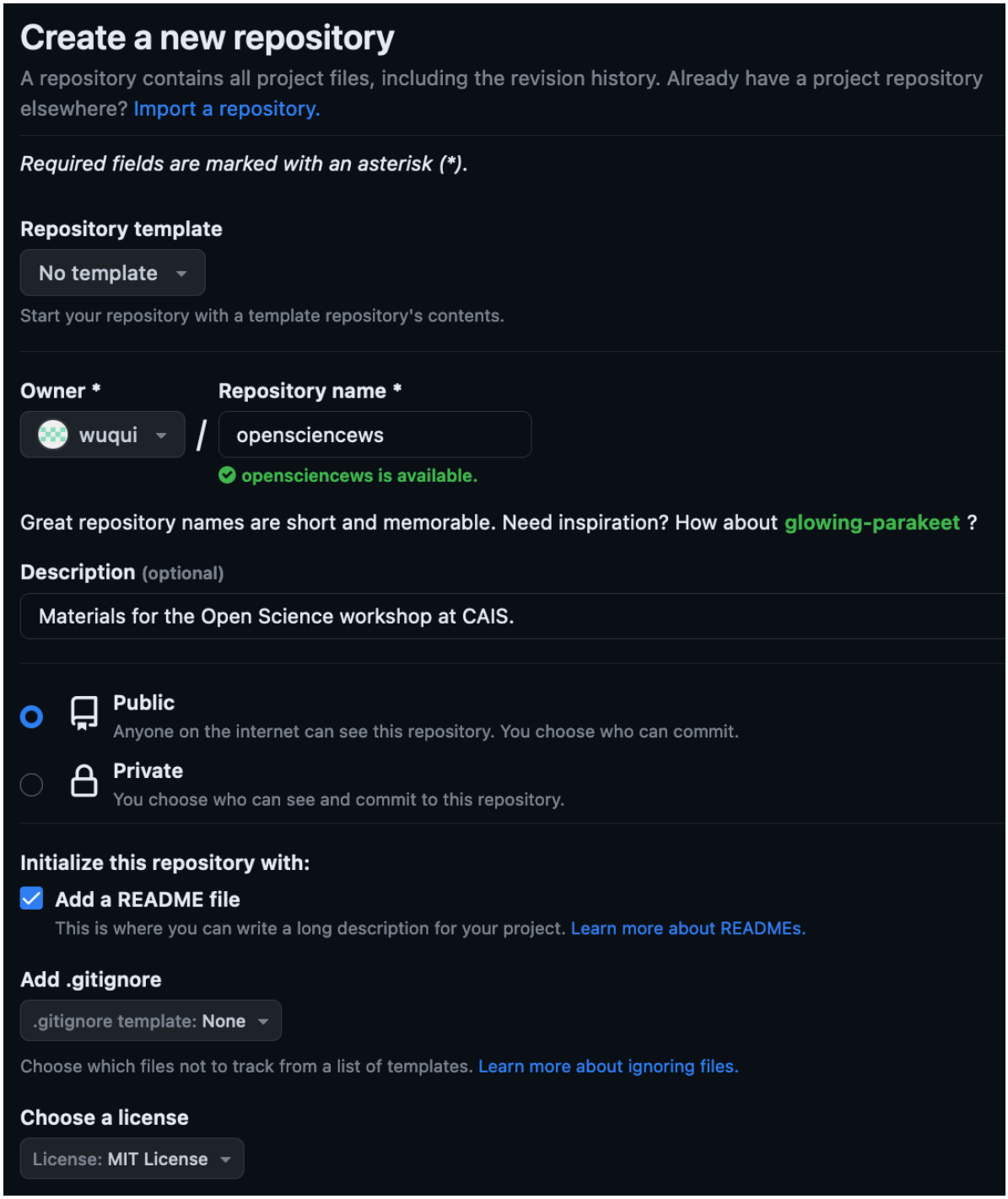
- setting up git user information (name, passwort)
- setting up GitHub authentication
- setting and storing authentication ('token')

---

create a repository on GitHub

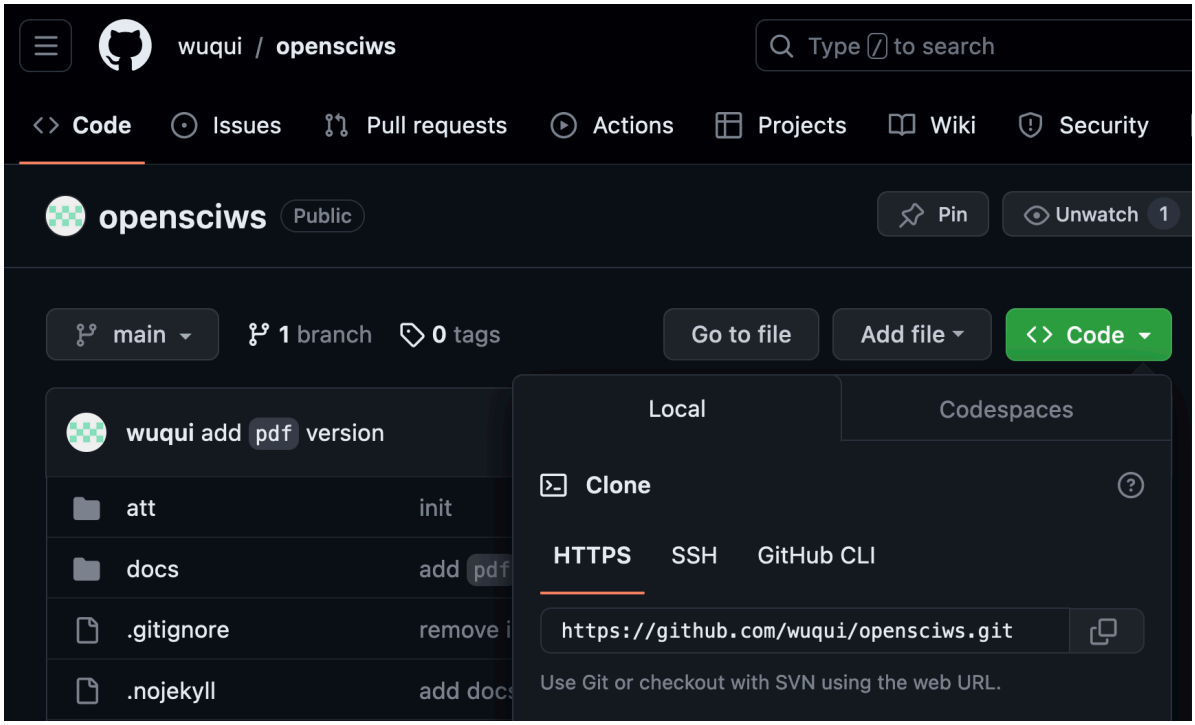
1. (create GitHub account)
2. click on `New` (https://github.com/new)
3. specify repo name [3]
4. specify description
5. specify visibility: private or public
6. select `Add a README file`
7. specify licence [4]

[3] safe: lowercase alphabet characters
[4] good choice for many purposes: MIT license


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 *
Repository name *
wuqui
/
opensciencews
opensciencews is available.
Great repository names are short and memorable. Need inspiration? How about glowing-parakeet ?
Description (optional)
Materials for the Open Science workshop at CAIS.
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: MIT License

`clone` repositories

go to the folder where you want your project to live




```
git clone https://github.com/wuqui/opensciws.git
```

---

adding, `commit`ing, and `push`ing changes

```
git add src/tests.py
git commit -m 'add tests'
git push
```

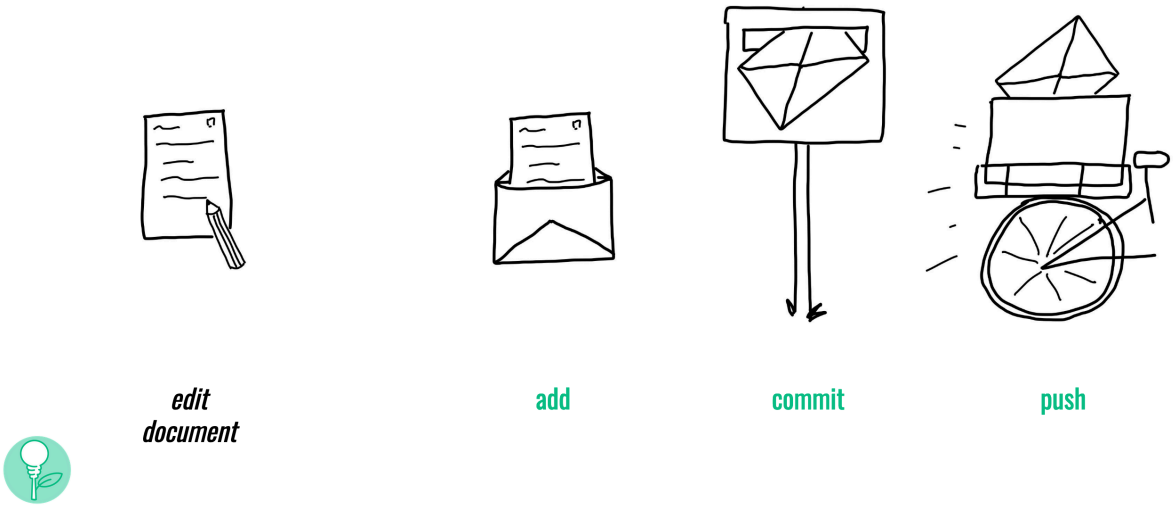
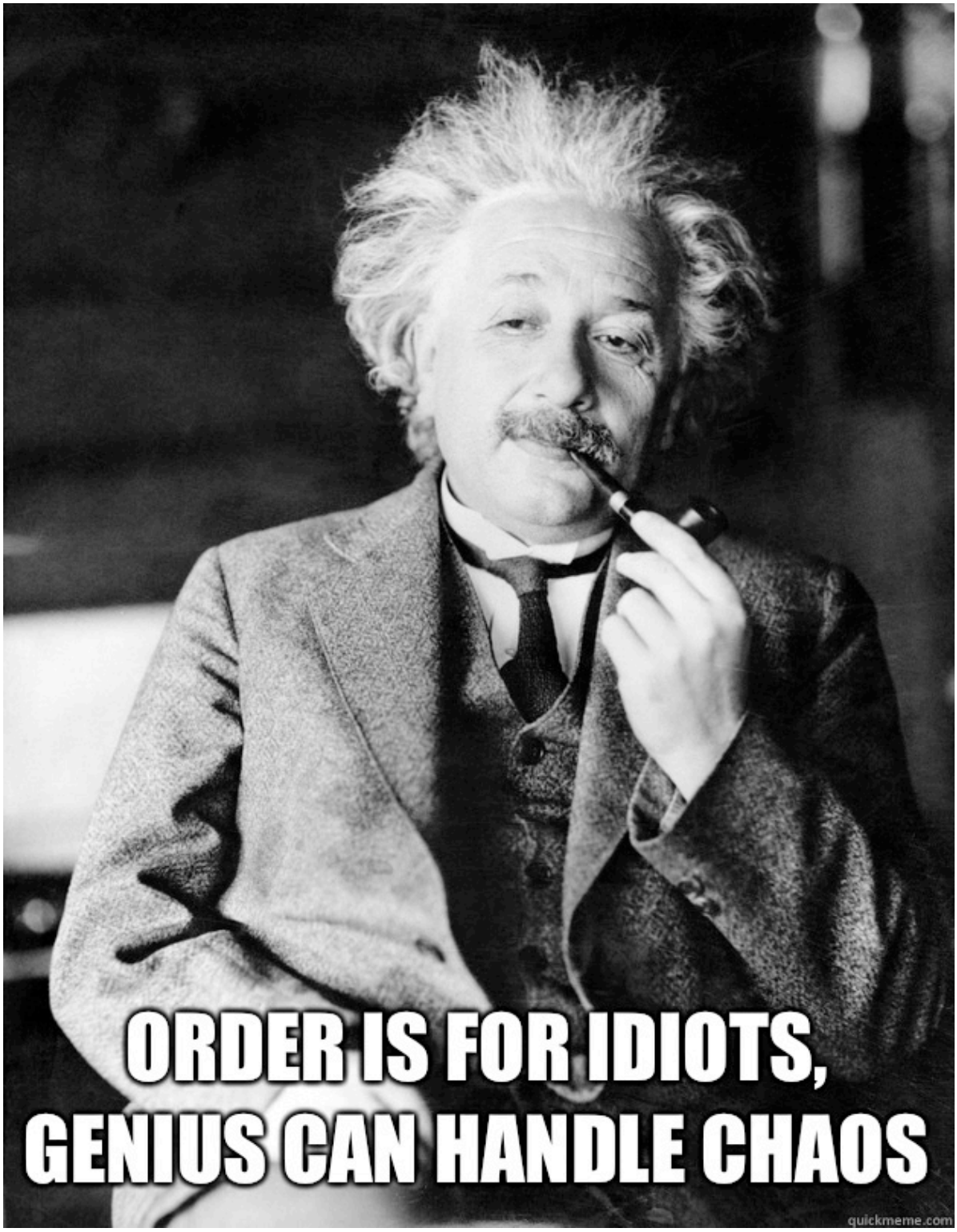


Figure 16: (source)

Project structure


ORDER IS FOR IDIOTS,
GENIUS CAN HANDLE CHAOS
quickmeme.com

Let's not pretend we're all geniuses ...

File names

NO

- `Myabstract.docx`
- `Joe's Filenames Use Spaces and Punctuation.xlsx`
- `figure 1.png`
- `fig 2.png`
- `JW7d^(2sl@deletethisandyourcareerisoverWx2*.txt`

YES

- `2014-06-08_abstract-for-sla.docx`
- `Joes-filenames-are-getting-better.xlsx`
- `Fig01_scatterplot-talk-length-vs-interest.png`
- `Fig02_histogram-talk-attendance.png`
- `1986-01-28_raw-data-from-challenger-o-rings.txt`

File names should be:

- machine-readable
- human-readable
- consistent
- optional: play well with default ordering (e.g. include timestamps)

File structure

```
.
  analysis            <- all things data analysis
     src              <- functions and other source files
  comm
     internal-comm    <- internal communication such as meeting notes
     journal-comm     <- communication with the journal, e.g. peer review
  data
     data_clean       <- clean version of the data
```

```
    data_raw        <- raw data (don't touch)
dissemination
    manuscripts
    posters
    presentations
documentation       <- documentation, e.g. data management plan
misc                <- miscellaneous files that don't fit elsewhere
```

Practice: project management

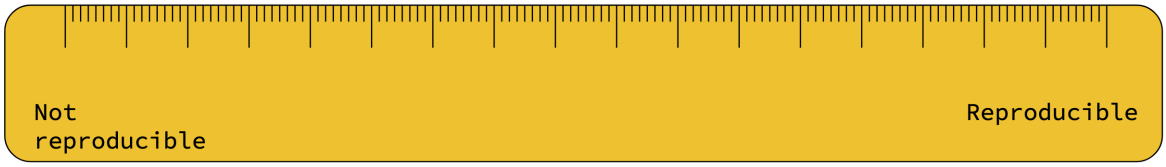
You have until 11:50 h to work on either …

1. developing a project structure for your needs from scratch
2. refactoring/cleaning an existing project[5]

Optionally: set up version control via git/GitHub for this project.

Code

Reproducibility et al.



---

[5] make a backup first

		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

Figure 17: The Turing Way

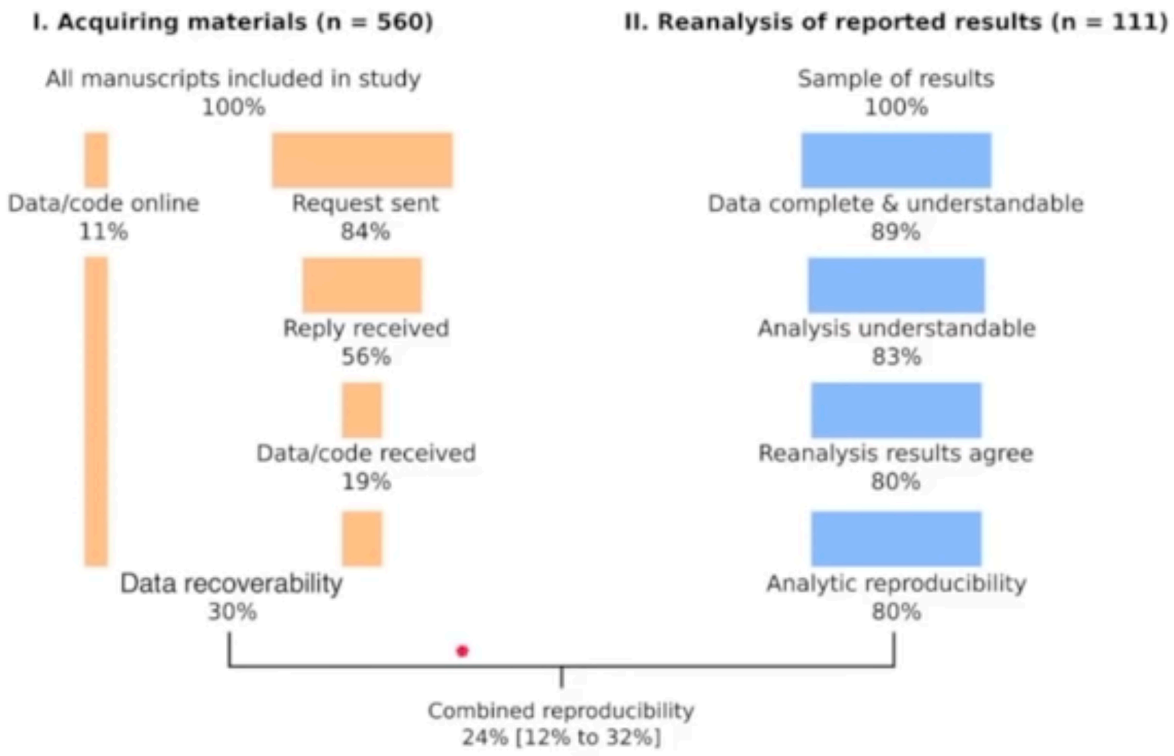


Figure 18: source

The quality of tools

Figure 19: source

---

notebooks and literate programming

- Knuth
  - application
    - R
    - Python: nbdev

Licensing

Code

Other materials

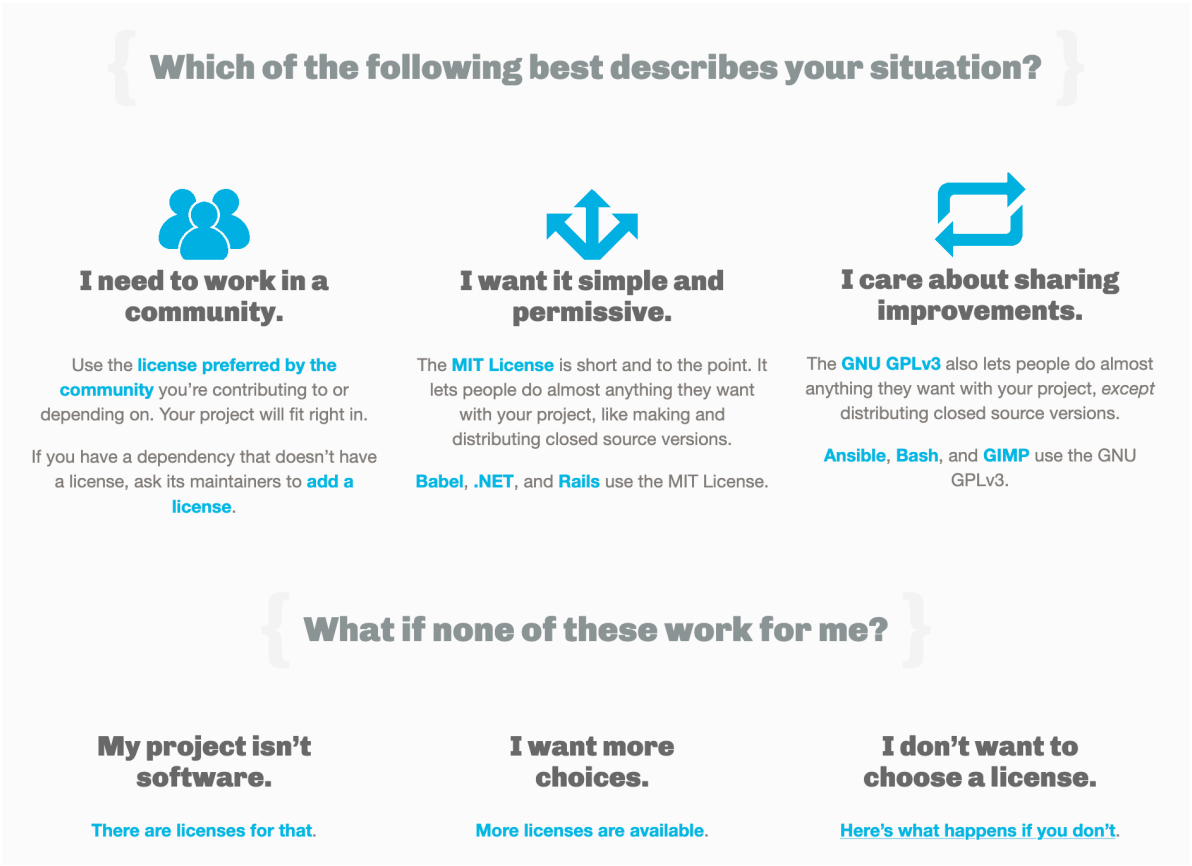


Figure 20: https://choosealicense.com

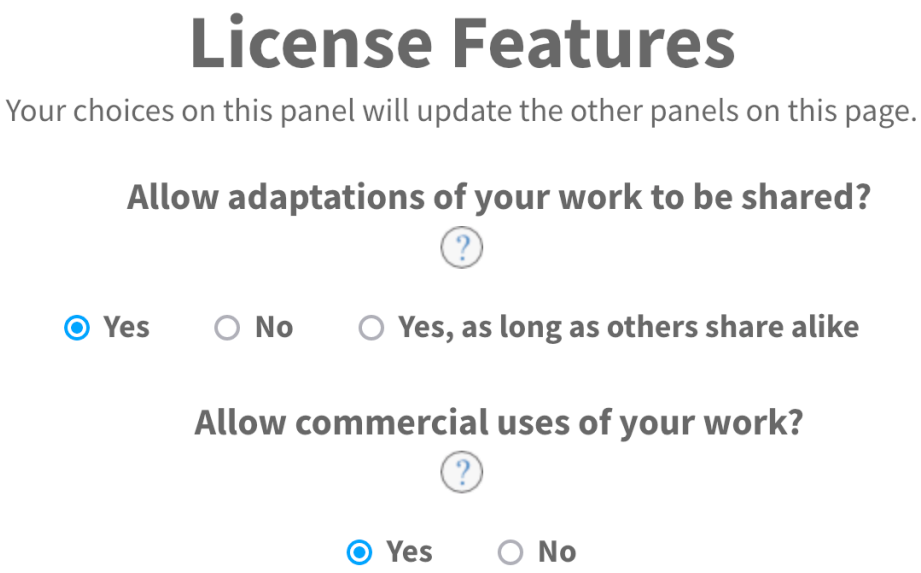


Figure 21: https://creativecommons.org/choose/

Data and methods

FAIR data

Types of data

- interviews
- questionnaires
- web
- social media

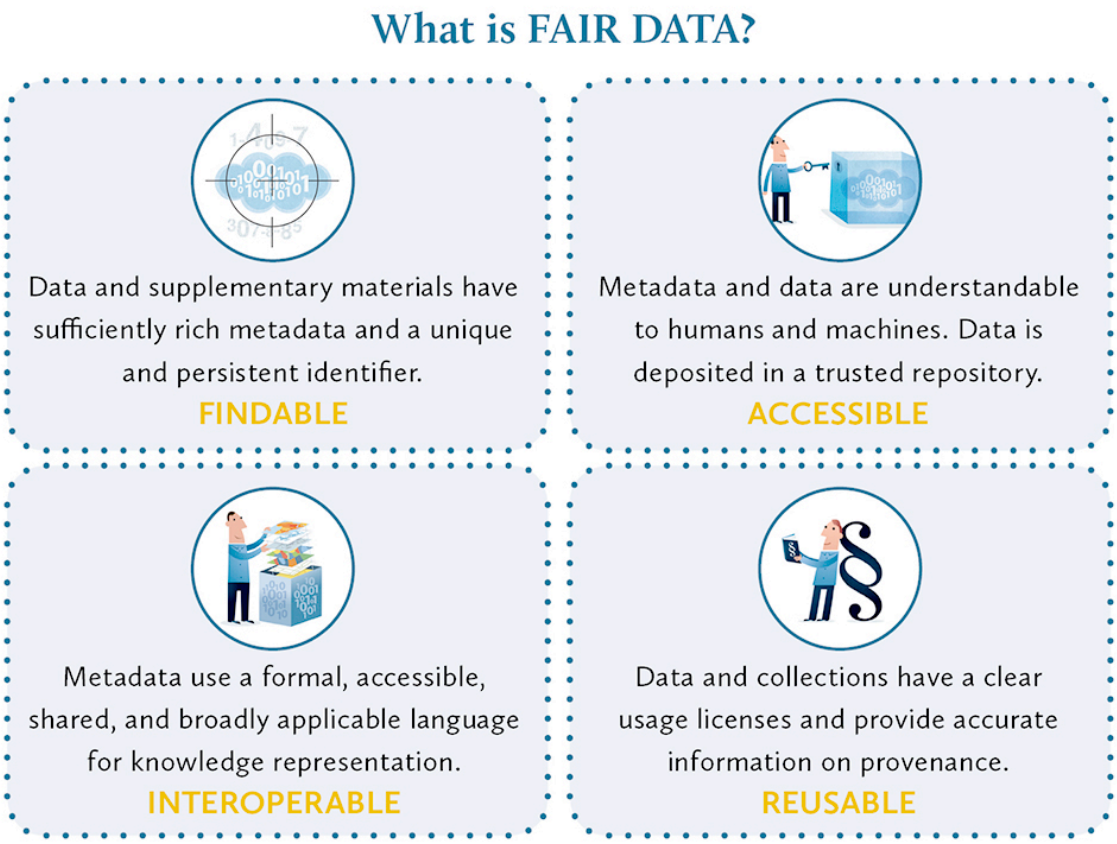


Figure 22: source

Authoring

Authoring **How** can we organise our project from the beginning so that we can publish outputs in the end?

Publishing **Where** can I publish my work (platforms, research centers infrastructure, …)?

Plain text

Quarto

- single source → multiple output formats
  - PDF for publication outlets
  - blog
  - website

Publishing

How: How can we organise our project from the beginning so that we can publish outputs in the end? Where: Where can I publish my work (platforms, research centers infrastructure, …)?

Pre-registration

Figure 23: source

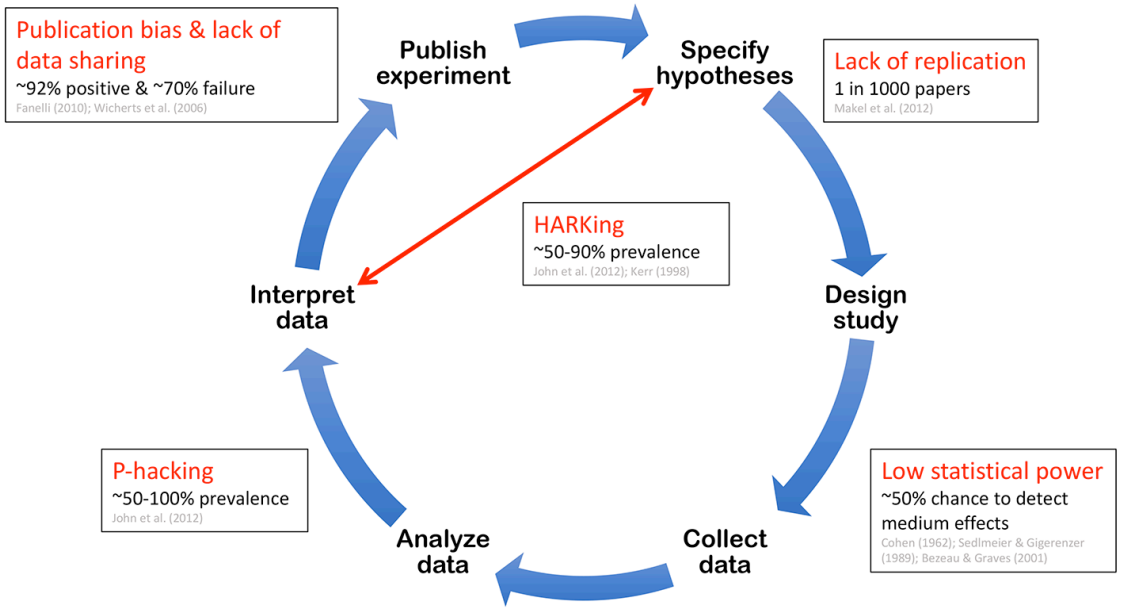


Figure 24: source

Outlets

ArXiV preprints
Zenodo all kinds including data, code, preprints, etc.
GitHub and GitLab code, software
Open Science Framework all kinds including data, code, preprints, preregistration, etc.
Software Heritage archival of code (long-term)
Papers with Code code and data for and with papers, mostly Machine Learning
...

Resources

- DRA
- The Turing Way
- Data Carpentries