

# 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](mailto: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](https://wuqui.github.io/opensciws/opensciws_slides.html)  
**website version** [https://wuqui.github.io/opensciws/opensciws\\_website.html](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 fields

- 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**

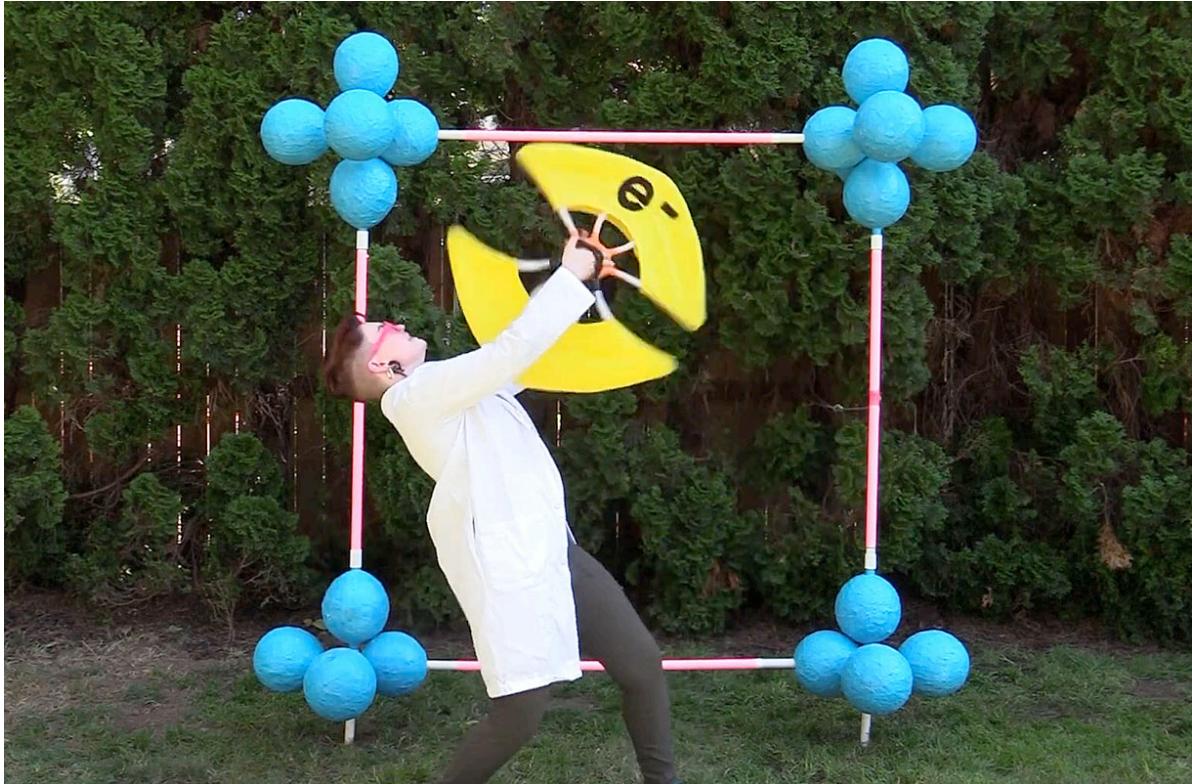
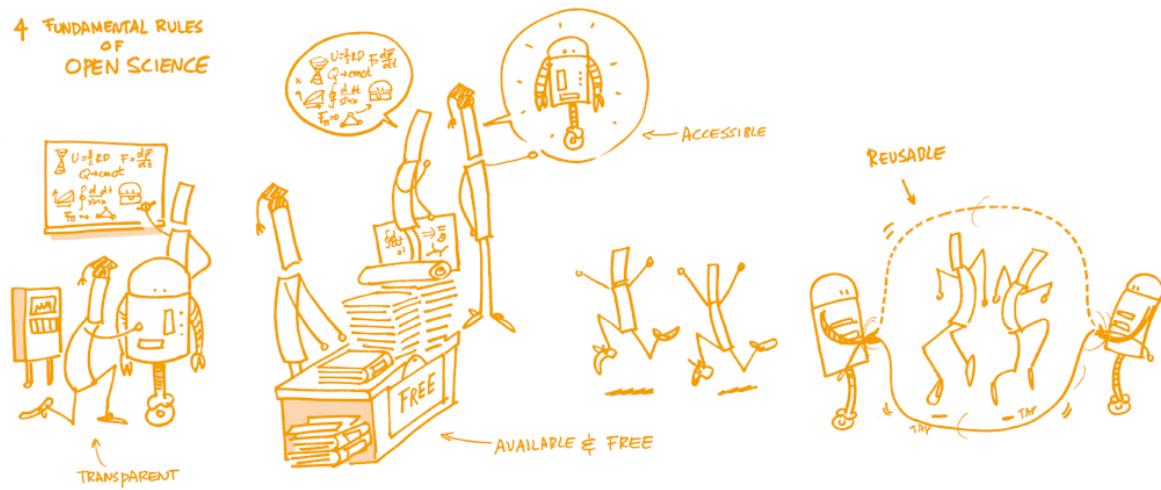


Figure 1: [Dance your PhD](#)

## What is Open Science?



---

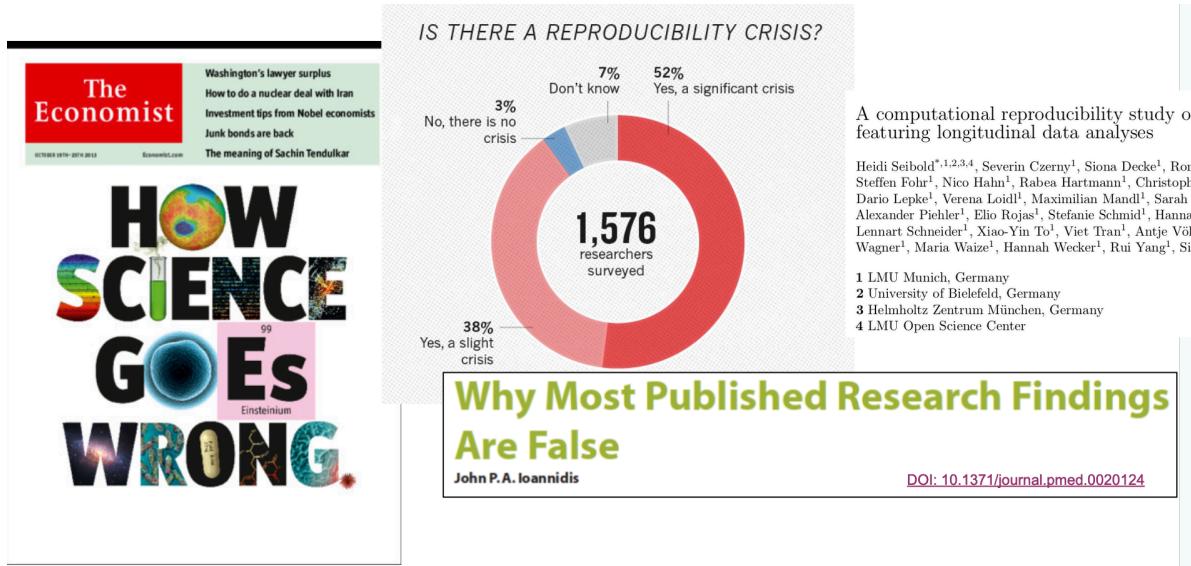
## Why should we do Open Science?

- 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?

---

---



## Is this what we want?

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.

## *Scientists: Mostly Hackers*

“The case against science is straight-forward: much of the scientific literature, perhaps half, may simply be untrue. Afflicted by studies with small sample sizes, tiny effects, invalid exploratory analyses, and flagrant conflicts of interest, together with an obsession for pursuing fashionable trends of dubious importance, **science has taken a turn towards darkness.**”

Richard Horton



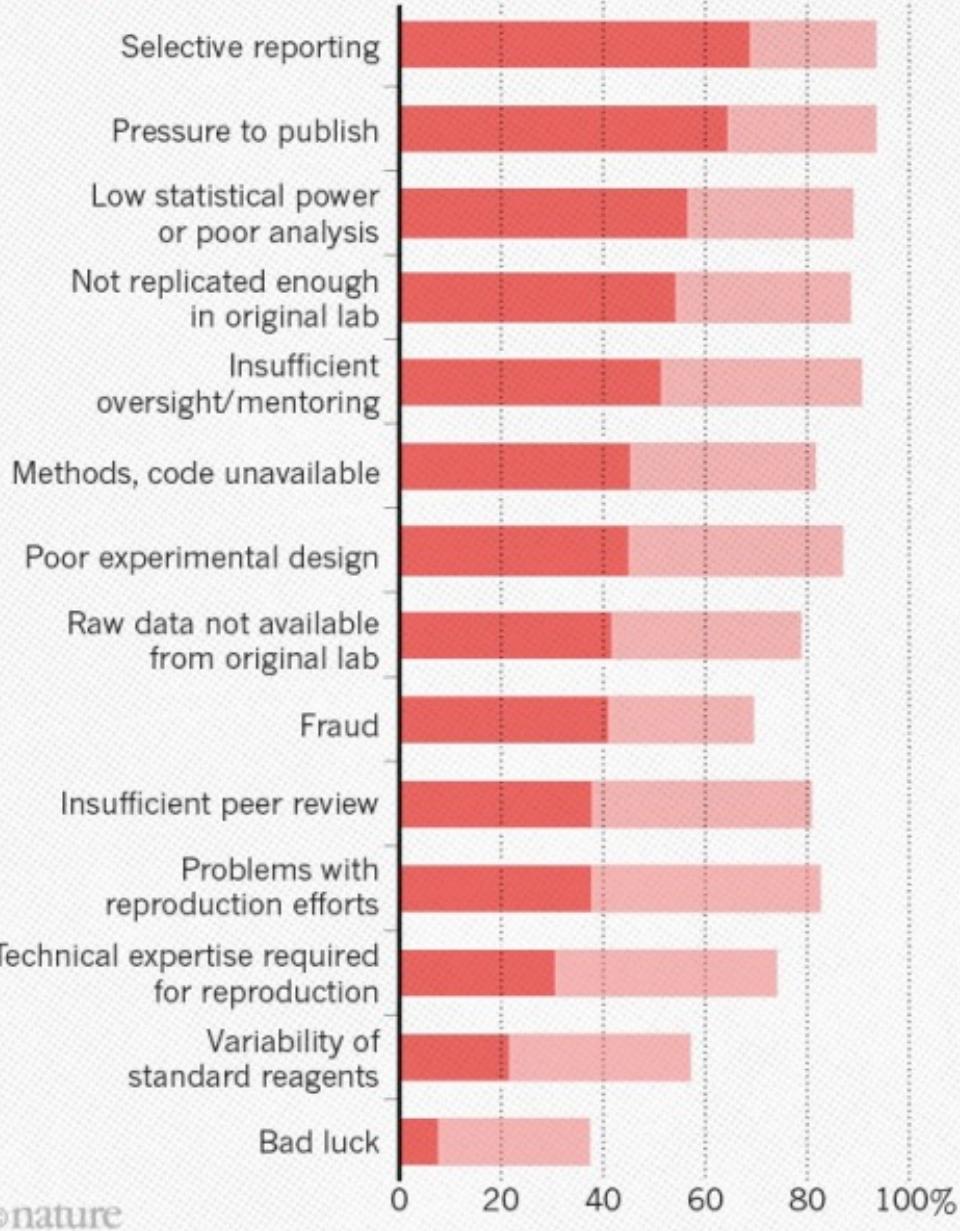
Richard Horton, United Kingdom  
Editor-in-Chief  
*The Lancet*

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

## WHAT FACTORS CONTRIBUTE TO IRREPRODUCIBLE RESEARCH?

Many top-rated factors relate to intense competition and time pressure.

● Always/often contribute   ● Sometimes contribute



©nature

Figure 4: [source](#)

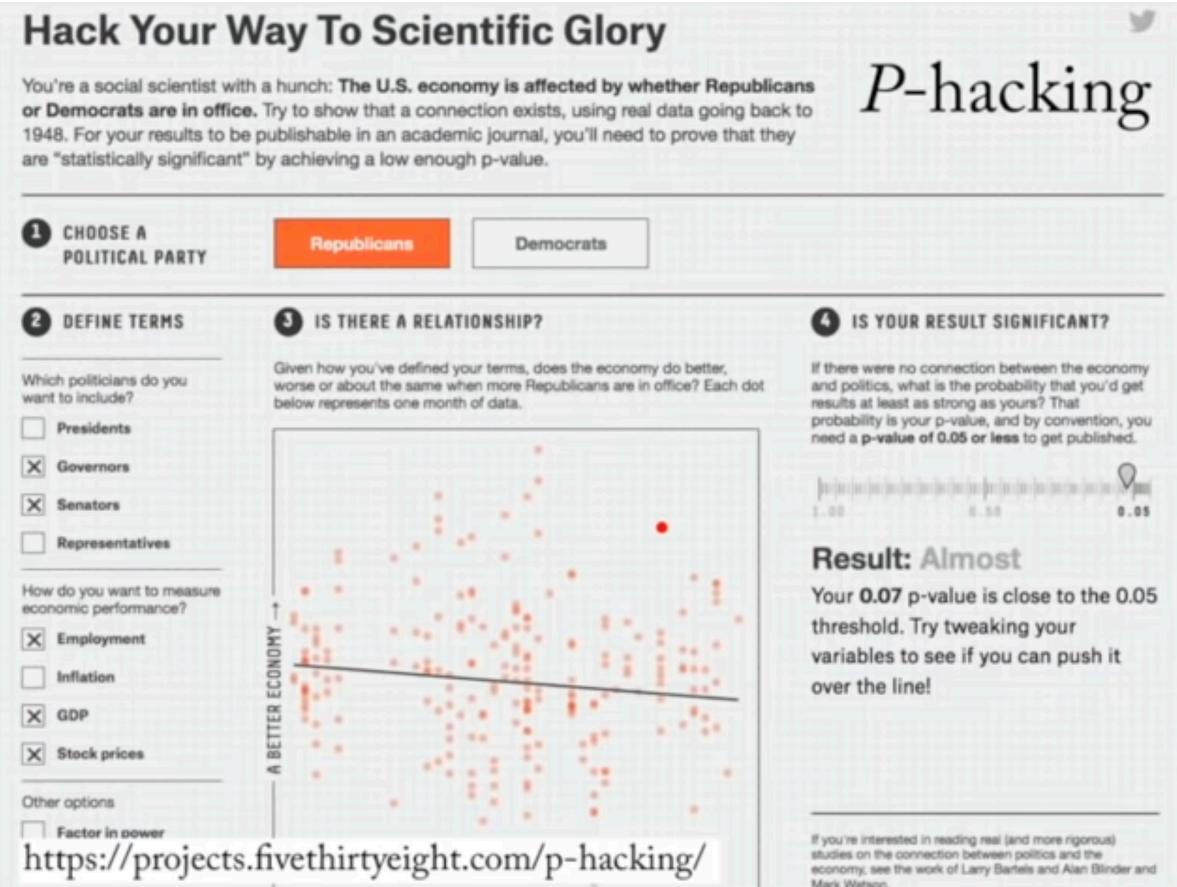


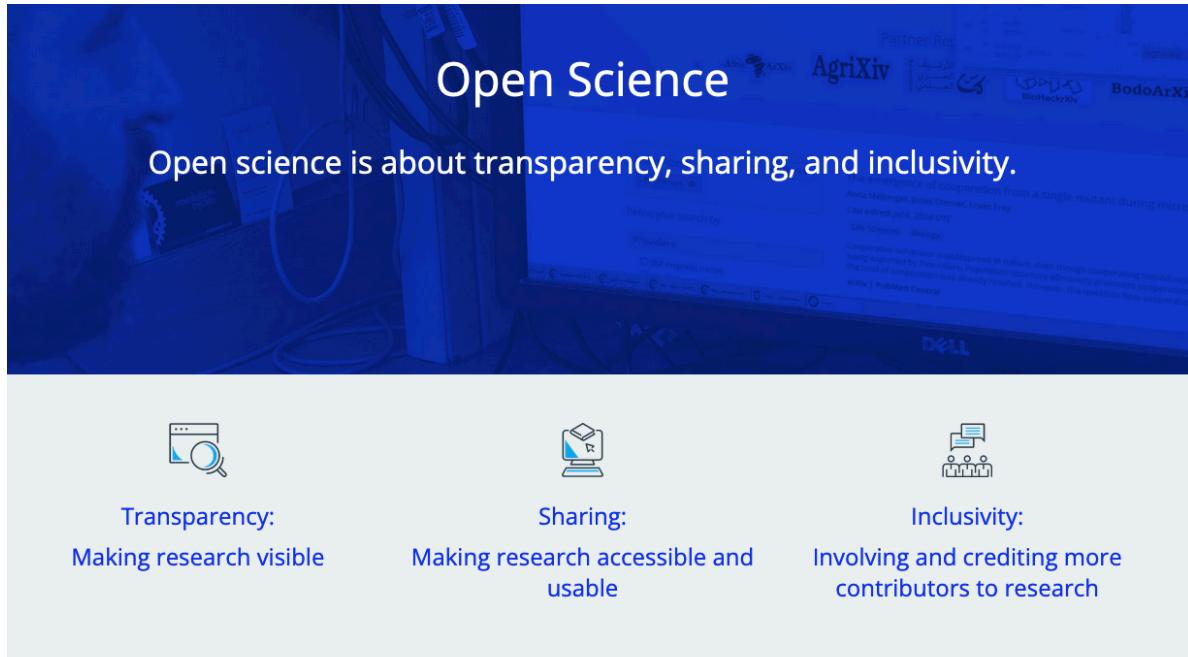
Figure 5: [source](#)

## *Skills to Pay the Bills*

- Professors make professors
- How to get funding
- How to get published
- How to get cited
- How to give credit (citation)
- Research skills often *informally* transmitted

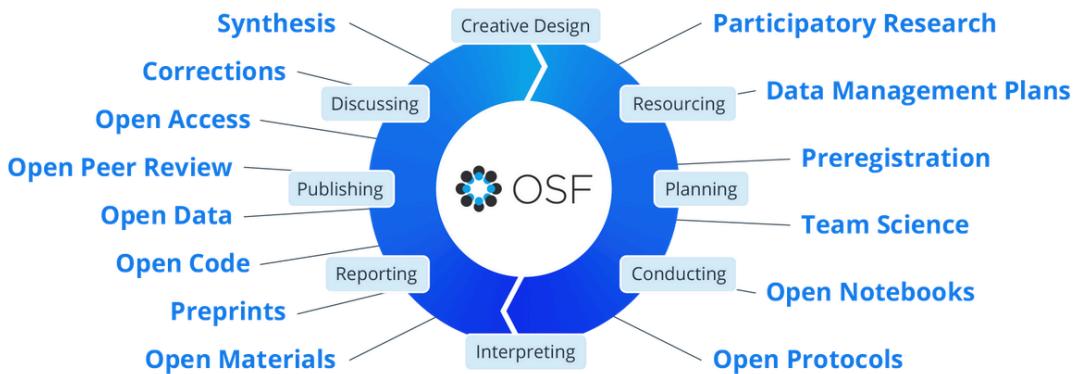


Figure 6: [source](#)



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



We advocate for lifecycle open science. There are open scholarship activities at every stage of the research lifecycle (see figure above) that individually and collectively contribute to improving science, with everyone playing a role:

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>

---

you pay my salary,  
but you don't get access to my  
work.



Figure 9: [source](#)

## Learning outcomes

- \* Open Science = Good science in a digitized world



- \* Open Science impacts all steps in the research cycle  
=> change in practices in planning, data collection, analysis, presentation, ...



- \* Open Science = social change

→ makes it difficult (social hurdles)  
→ it is possible, if we understand mechanisms  
+ support each other



CC-BY 4.0 Heidi Seibold  
@HeidiBoya

---

## 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?



paper\_draft.tex



paper\_update.tex



paper\_final.tex



paper\_final2.tex



paper\_final3.tex



paper\_please\_let\_this\_be\_the\_final.tex



paper\_please\_let\_this\_be\_the\_final123.tex



paper\_ultrafinal.tex



paper\_I\_will\_kill\_myself\_if\_this\_will\_go\_on.tex

Dear colleagues,

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

► ① attachment: StudyProposal█████\_Validation\_V1\_250918.docx

source

---

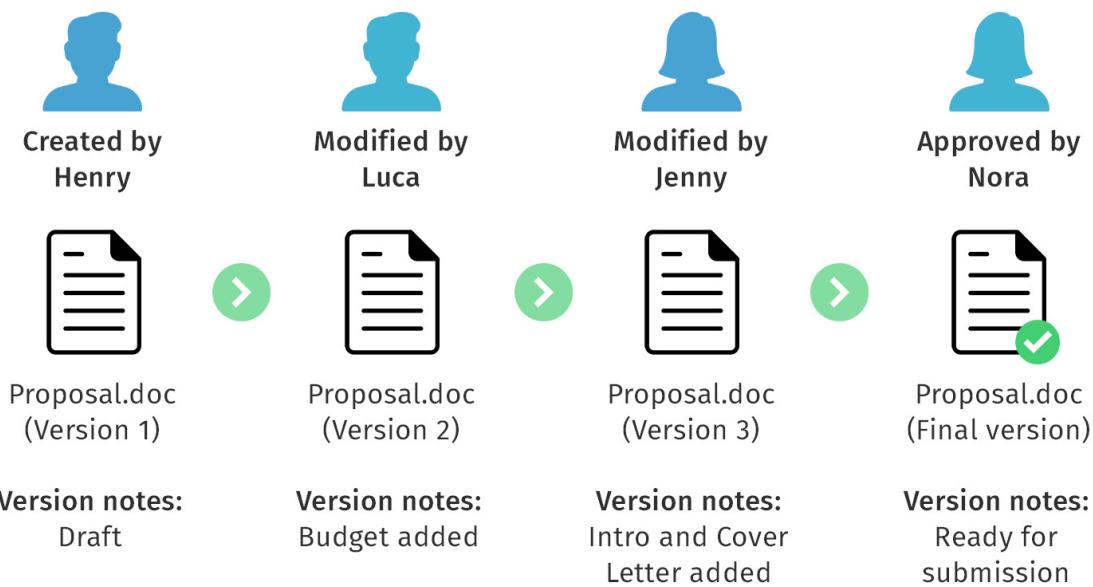


Figure 10: [source](#)

---

---

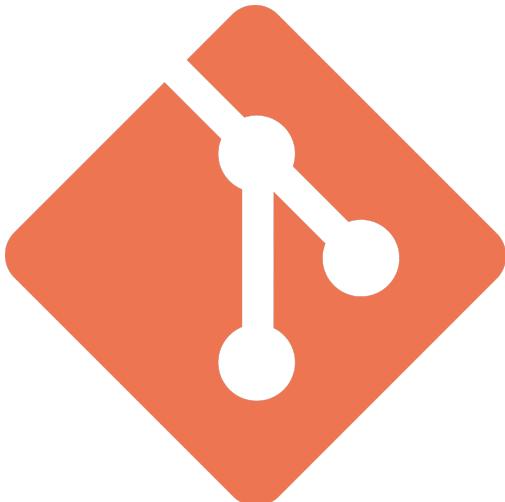
## git and GitHub/GitLab

**git** software on your machine

# BENEFITS OF DOCUMENT VERSION CONTROL



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

---

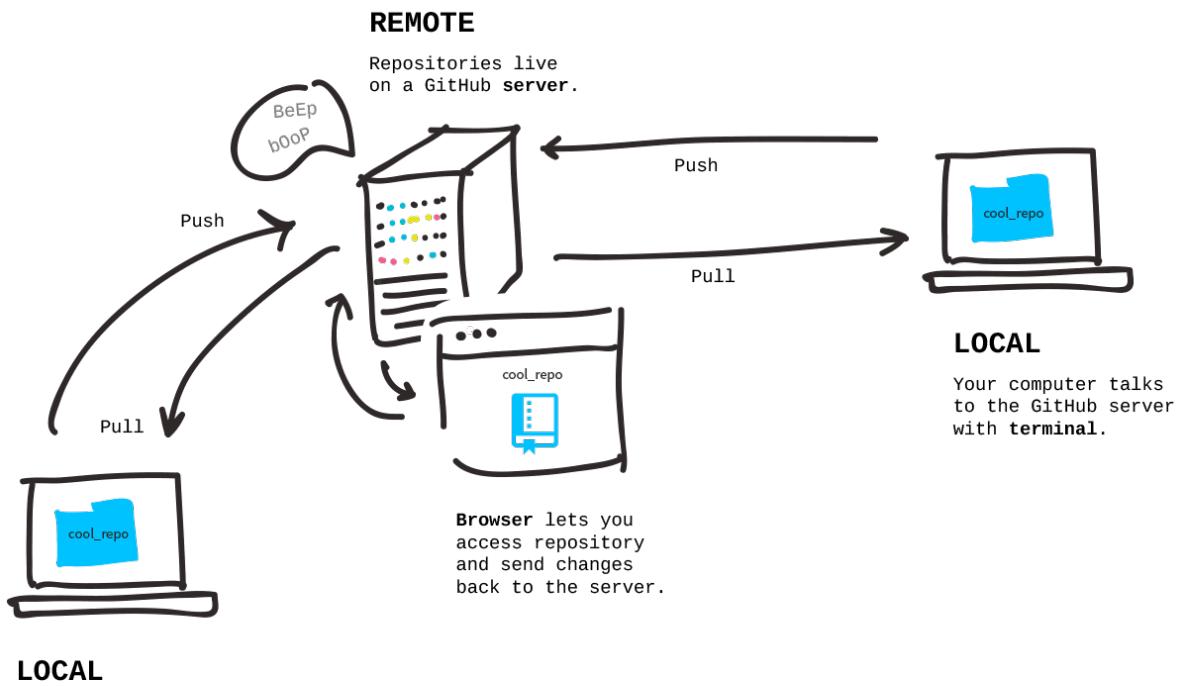


Figure 12: ([source](#))

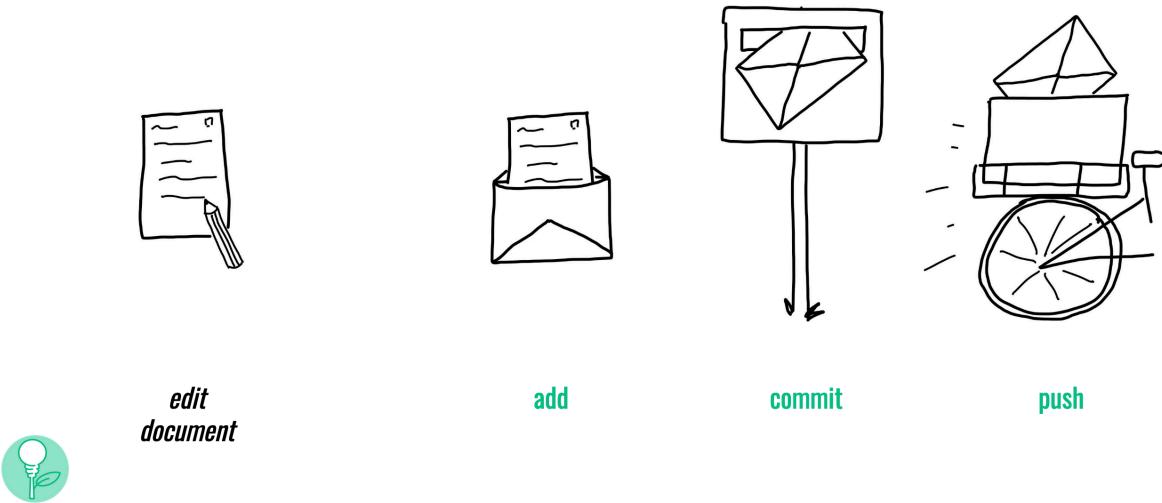


Figure 13: ([source](#))

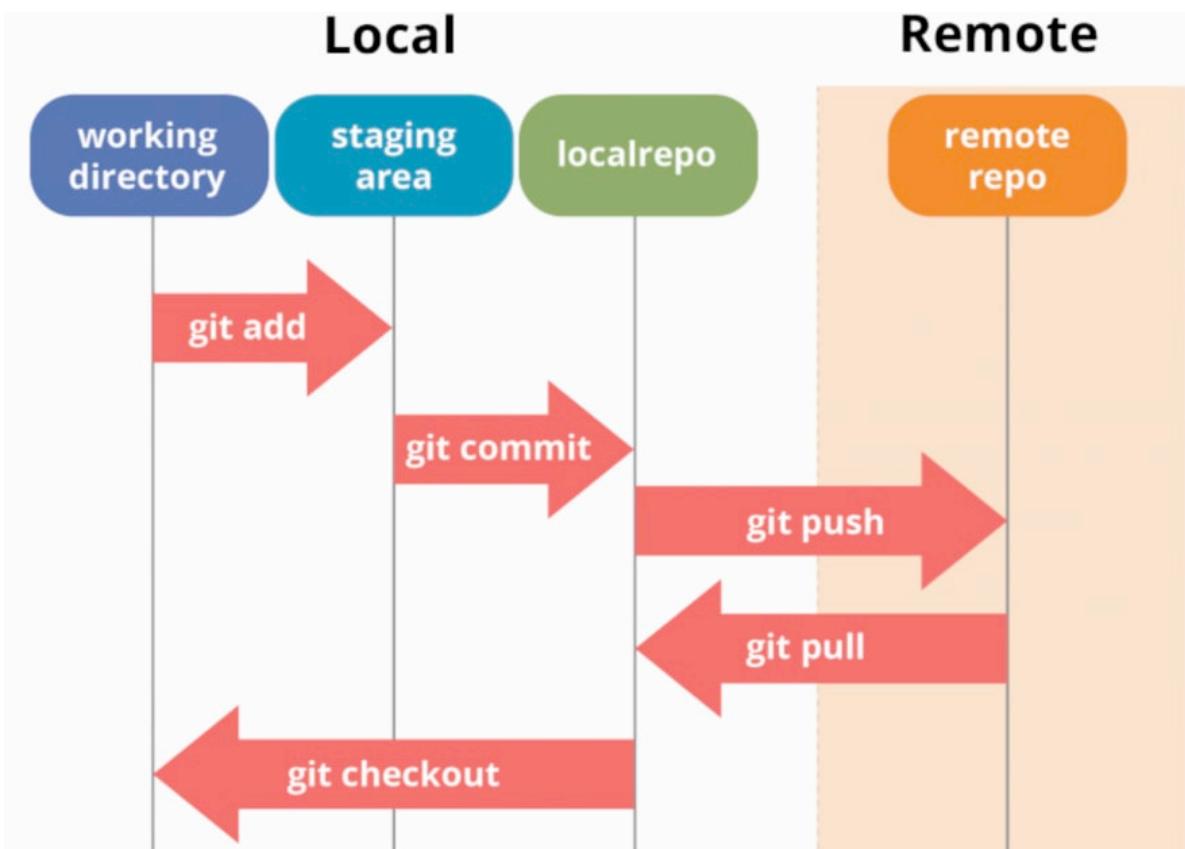


Figure 14: [\(source\)](#)

## GitHub workflow

---

### Example

The screenshot shows a GitHub pull request titled "Improve grammar in compendia.md (#1822)". The pull request has been merged into the "main" branch. The commit was made by "inwaves and malvikasharan" on April 24, 2021, and is verified. It has one parent commit, 328d54a, and a commit hash of 72039372a2803a4ec15764b2c01017039320f096. The diff shows changes between two code versions:

```
132 - In the future, the research compendium may even be the publication itself which is being peer reviewed  
      (rather than just peer reviewing the paper, why not review the entire research project).  
133 -  
133 + In the future, the research compendium may even be the publication itself allowing peer review of the entire  
      research project.
```

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<sup>1</sup> tool; e.g.:
  - [GitKraken](#)
  - [GitHub Desktop](#)
- from within your editor/IDE<sup>2</sup>; e.g.:

---

<sup>1</sup>Graphical User Interface

<sup>2</sup>Integrated Development Environment

- RStudio
  - VSCode
- 

## **set up GitHub**

tutorial

- setting up git user information (name, password)
  - 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 <sup>3</sup>
4. specify description
5. specify visibility: private or public
6. select Add a README file
7. specify licence <sup>4</sup>

---

<sup>3</sup>safe: lowercase alphabet characters

<sup>4</sup>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 \*

wuqui ▾

Repository name \*

/ 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 ▾

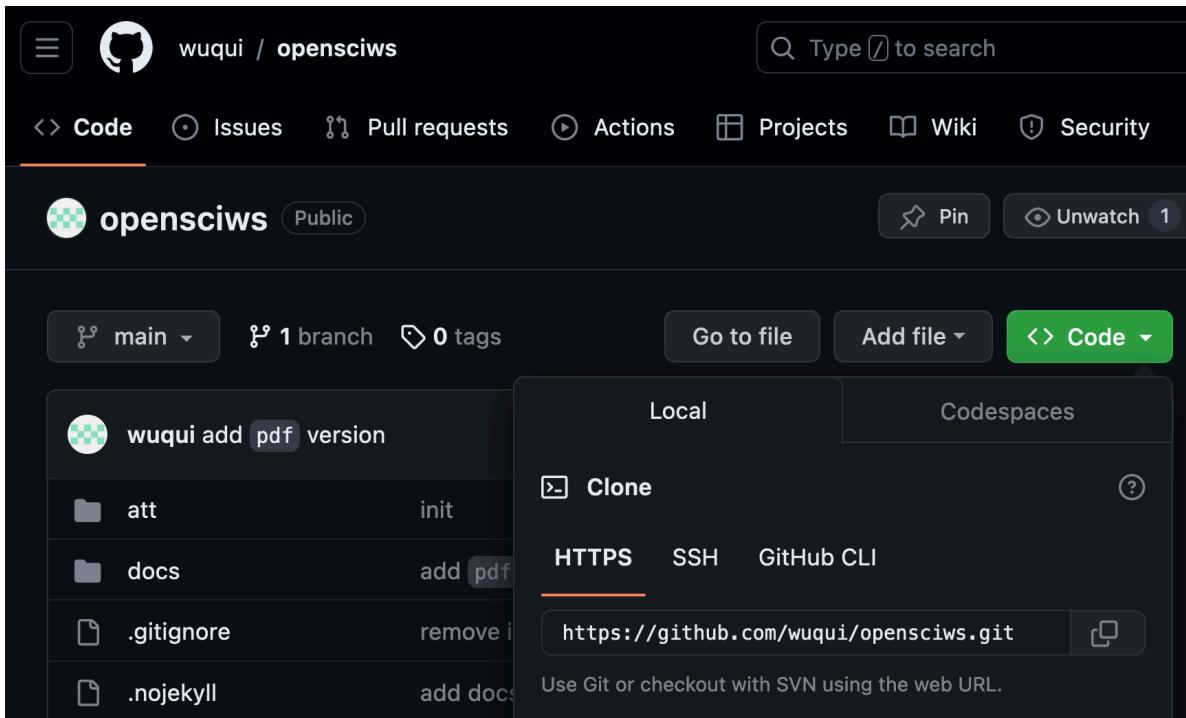
Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

## 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, committing, and pushing changes

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

---

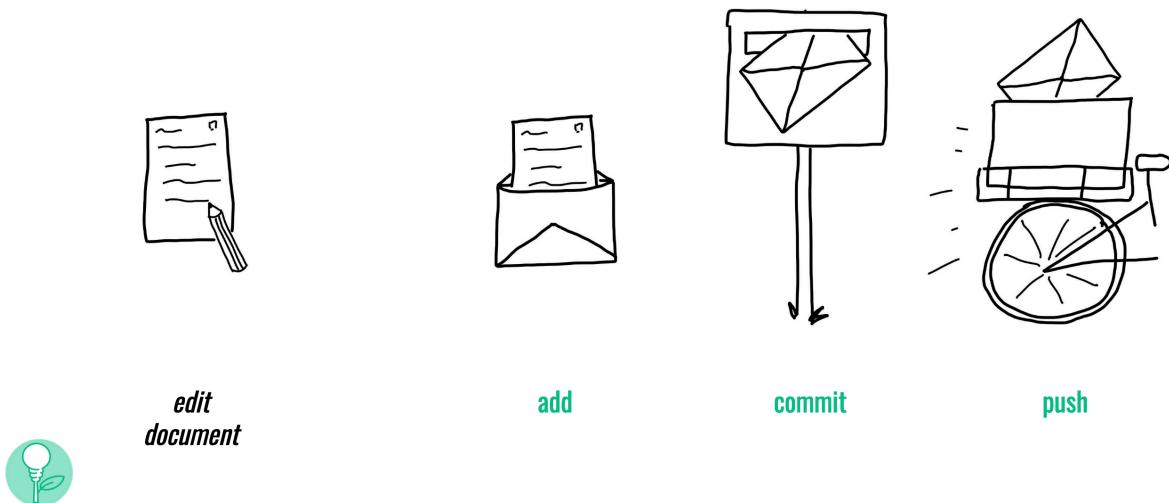
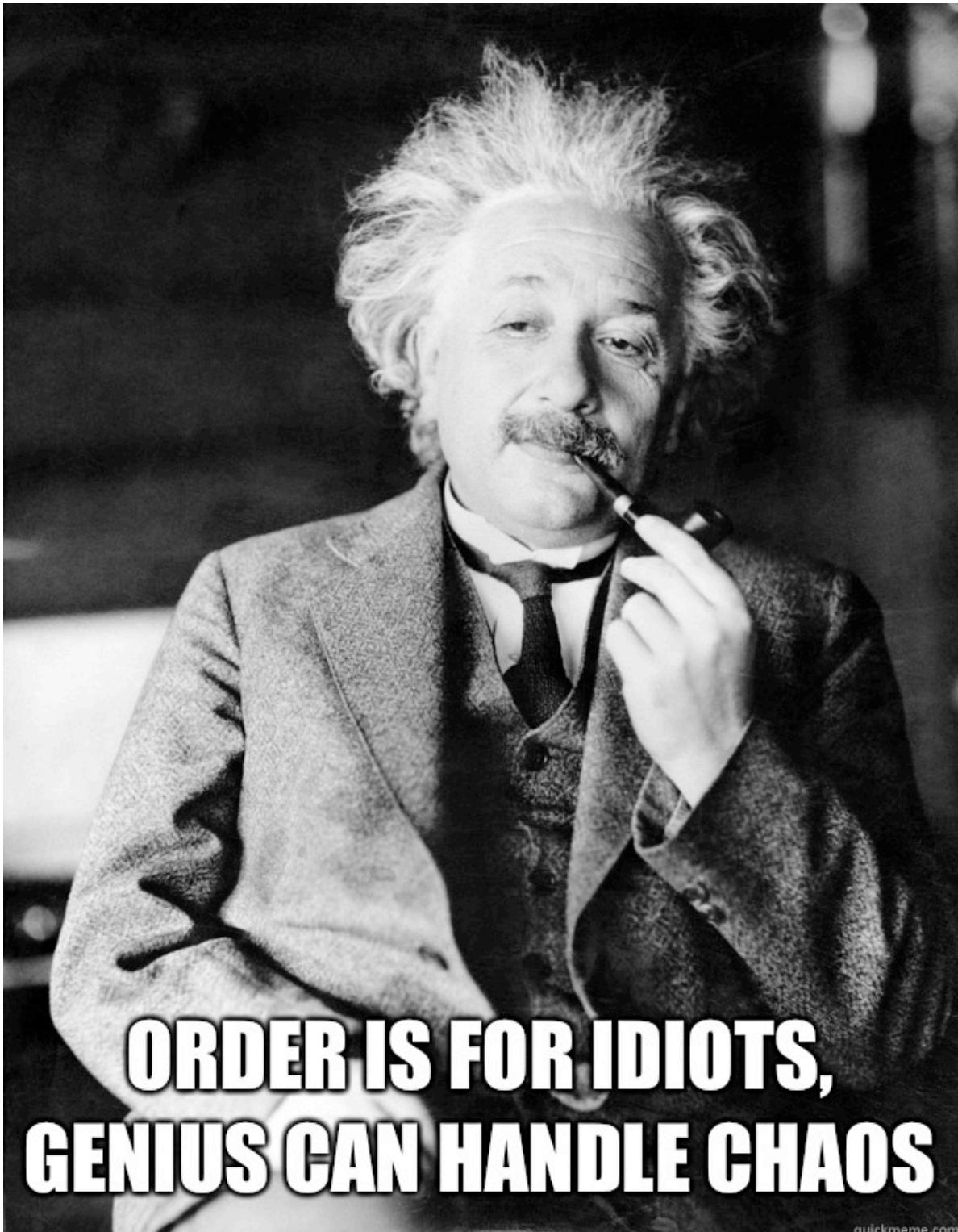


Figure 16: (source)

## Project structure

---



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<sup>5</sup>

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

---

## Code

---

## Reproducibility

---

## Reproducibility (crisis)

---

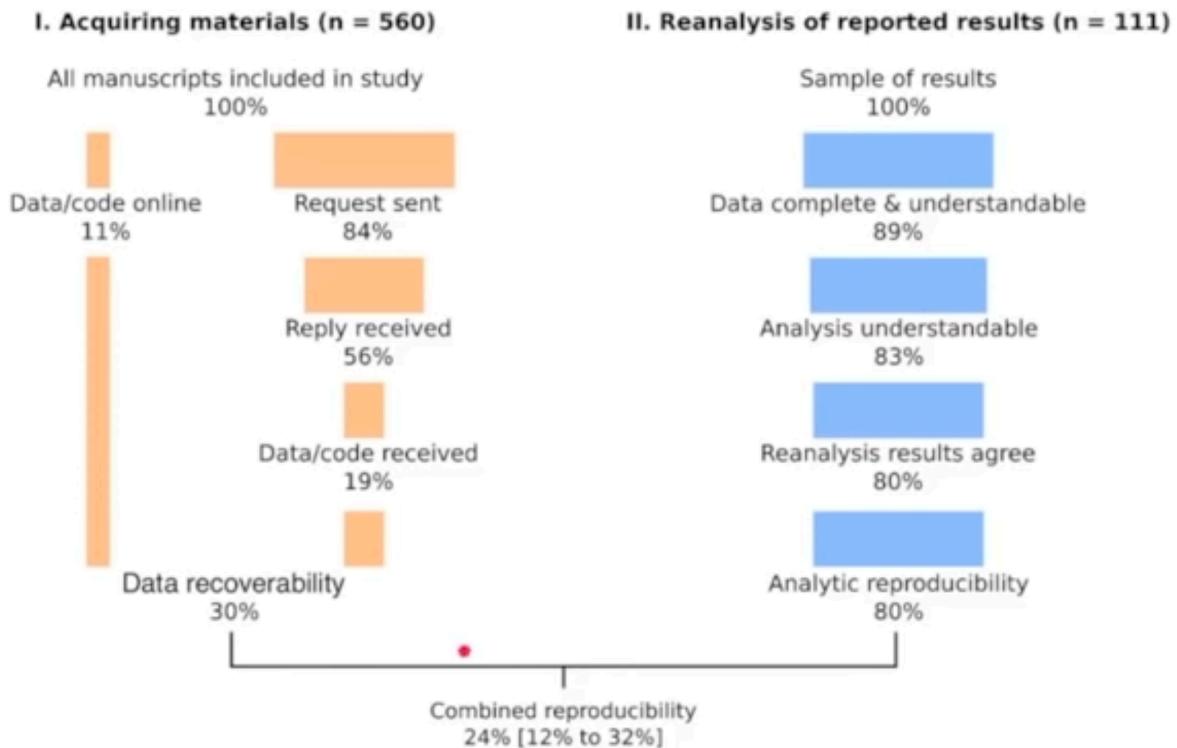


Figure 17: [source](#)

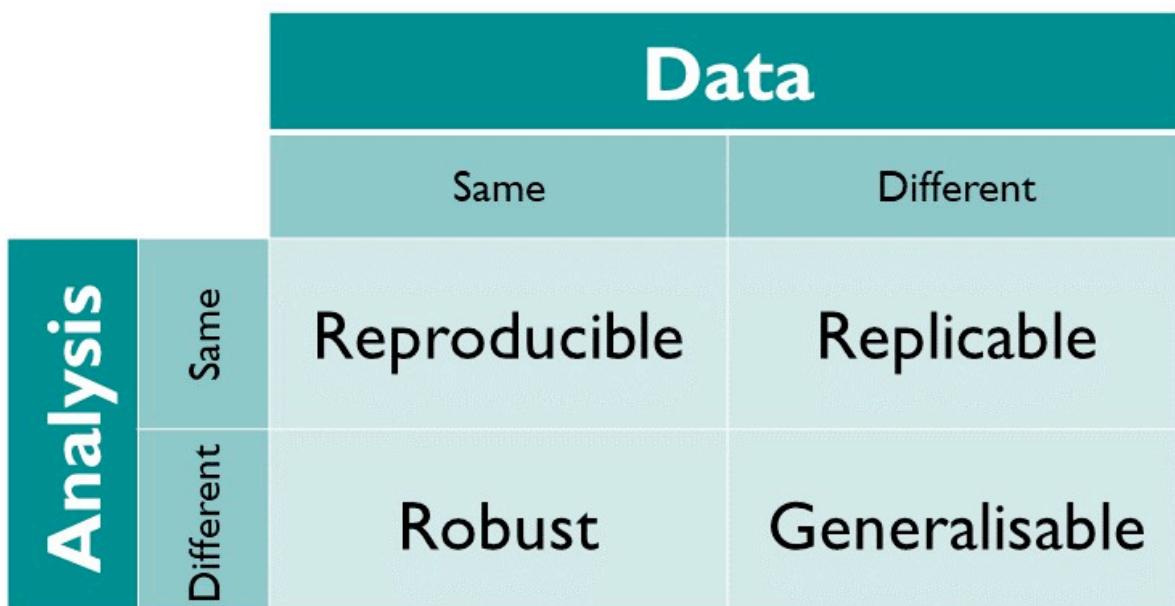
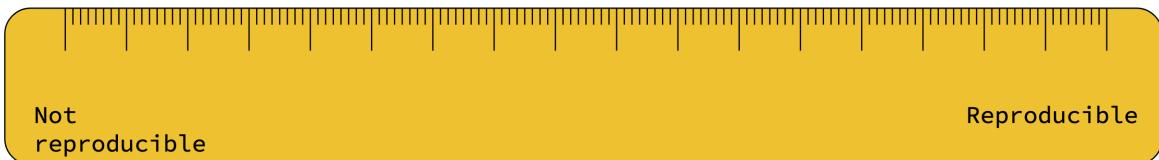


Figure 18: [The Turing Way](#)

## Reproducibility et al.



## The quality of tools

SCIENCE / TECH / MICROSOFT

# Scientists rename human genes to stop Microsoft Excel from misreading them as dates



/ Sometimes it's easier to rewrite genetics than update Excel

By James Vincent, a senior reporter who has covered AI, robotics, and more for eight years at The Verge.

Aug 6, 2020, 2:44 PM GMT+2 | □



Figure 19: [source](#)

## Testing code

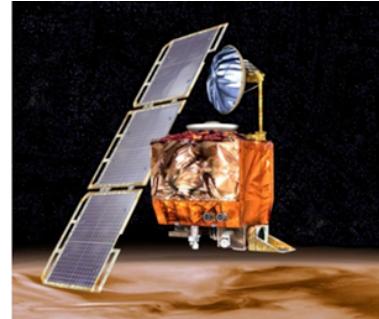
### Why we should test code

---

<sup>5</sup>make a backup first

LISA GROSSMAN 11.10.10 7:00 AM

# NOV. 10, 1999: METRIC MATH MISTAKE MUFFED MARS METEOROLOGY MISSION



The **\$125 million satellite** was supposed to be the first weather observer on another world. But as it approached the red planet to slip into a stable orbit Sept. 23, the orbiter vanished. Scientists realized quickly it was gone for good. “It was pretty clear that morning, within half-an-hour, that the spacecraft had more or less **hit the top of the atmosphere and burned up**,” recalled NASA engineer Richard Cook, who was project manager for Mars exploration projects at the time.

A NASA review board found that the problem was in the software controlling the orbiter’s thrusters. **The software calculated the force the thrusters needed to exert in pounds of force. A separate piece of software took in the data assuming it was in the metric unit: newtons.**

<https://www.wired.com/2010/11/1110mars-climate-observer-report/>

Figure 20: [source](#)

## Professional testing

### math

The Stan Math Library is a C++ template library for automatic differentiation of any order using forward, reverse, and mixed modes. It includes a range of built-in functions for probabilistic modeling, linear algebra, and equation solving.



math    automatic-differentiation    stan    stan-math-library

● C++    BSD-3-Clause    127    449    214 (20 issues need help)    23  
Updated 5 minutes ago

3.6 MB of library code

7.6 MB of test code



<https://github.com/stan-dev/math>

Figure 21: [source](#)

## Types of tests

### Testing frameworks

- Python: `pytest`
- R: `testthat`

### Analogy

- during the process of manufacturing a ballpoint pen, the cap, the body, the tail, the ink cartridge and the ballpoint are produced separately and **unit tested** separately.

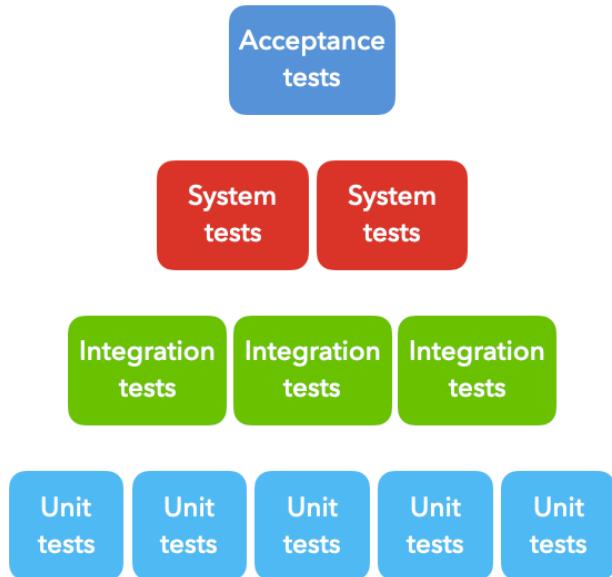


Figure 22: source

- When two or more units are ready, they are assembled and **integration testing** is performed, for example a test to check the cap fits on the body.
  - When the complete pen is integrated, **system testing** is performed to check it can be used to write like any pen should.
  - **Acceptance testing** could be a check to ensure the pen is the colour the customer ordered.

source

## Testing example

## using pytest for Python

## Python

```
# test_assert_examples.py

def test_uppercase():
    assert "loud noises".upper() == "LOUD NOISES"

def test_reversed():
    assert list(reversed([1, 2, 3, 4])) == [4, 3, 2, 1]

def test_some_primes():
    assert 37 in {
        num
        for num in range(2, 50)
        if not any(num % div == 0 for div in range(2, num))
    }
```

---

## Documenting code

### Literate programming

- 'Literate programming is a methodology that combines a programming language with a **documentation language**,
- thereby making programs **more robust, more portable, more easily maintained**,
- and arguably **more fun** to write than programs that are written only in a high-level language.
- The main idea is to treat a program as a piece of literature, **addressed to human beings** rather than to a computer.
- The program is also viewed as a **hypertext document**, rather like the World Wide Web. (Indeed, I used the word WEB for this purpose long before CERN grabbed it!)

Donald Knuth

---

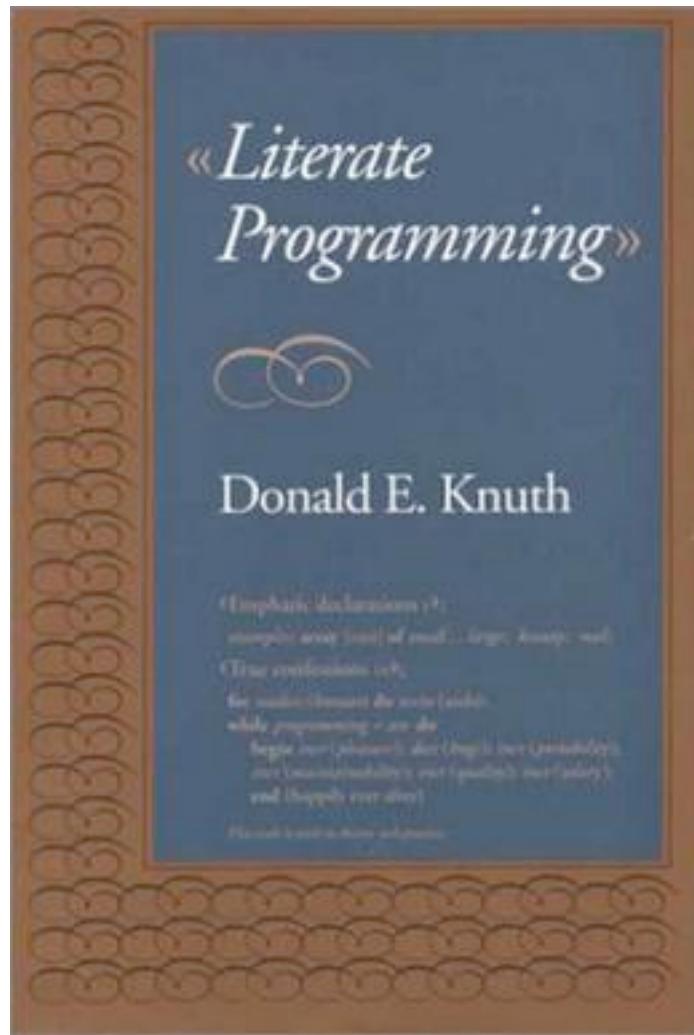


Figure 23: [source](#)

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** "jupyter covid\_19\_dashboard" Last Checkpoint: Last Friday at 11:45 PM (unsaved changes)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help
- Cells:**
  - In [13]: 

```
# importing libraries
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML
```
  - In [14]: 

```
# loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_confirmed.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_recovered.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_recovered.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')
```
  - In [15]: `confirmed_df.head()`
  - In [16]: `recovered_df.head()`
  - In [17]: `death_df.head()`
  - In [18]: `country_df.head()` (highlighted with a green border)
- Right Side:** A large orange "jupyter" logo watermark.

Figure 24: [source](#)

## Notebooks

Python

R

→ both work with [Quarto](#)

## Example using nbdev for Python

Programming a deck of cards: [https://github.com/fastai/nbdev\\_cards/](https://github.com/fastai/nbdev_cards/)

```
---
```

```
title: "Components of a Quarto document"
output: html_document
date: "2022-08-24"
execute:
  echo: false
---
```

## Human-readable text

This is a Quarto document based on an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. When you click the **Render** button a document will be generated that includes both content as well as the output of any embedded code chunks within the document. You can embed an code chunk like this:

```
{r cars}
summary(cars)
```

## Including code chunks

You can embed plots, for example:

```
{r pressure}
#| echo = FALSE
plot(pressure)
```

Figure 25: [source](#)

```
In [ ]: #| export
suits = ["♣", "♦", "♥", "♠"]
ranks = [None, "A"] + [str(x) for x in range(2,11)] + ["J", "Q", "K"]
```

We will be using numbers to represent playing card clubs and ranks. These are the suits:

```
In [ ]: suits
```

```
Out[ ]: ['♣', '♦', '♥', '♠']
```

For instance the suit at index `0`:

```
In [ ]: suits[0]
```

```
Out[ ]: '♣'
```

These are the ranks:

```
In [ ]: ranks
```

```
Out[ ]: [None, 'A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']
```

For instance the rank at index `1` (note that there isn't a playing card at position `0`, since we want the ranks to match the indices where possible):

---

## Literate testing with nbdev

For instance, here's a test of equality...

```
In [ ]: test_eq(Card(suit=1, rank=3), Card(suit=1, rank=3))
```

```
In [ ]: #| hide  
test_ne(Card(suit=2, rank=3), Card(suit=1, rank=3))  
test_ne(Card(suit=1, rank=2), Card(suit=1, rank=3))
```

...and a test of < ...

```
In [ ]: assert Card(suit=1, rank=3)
```

...and finally of > :

```
In [ ]: assert Card(suit=3, rank=3)>Card(suit=2, rank=3)  
assert not Card(suit=1, rank=3)>Card(suit=2, rank=3)
```

---

## Additional benefits of nbdev

- publishing documentation via Quarto
  - simple, integrated testing
  - continuous integration
  - dependency management
  - publishing code for PyPI and conda
-

## R: Quarto and RMarkdown

```
---
```

```
title: "Components of a Quarto document"
output: html_document
date: "2022-08-24"
execute:
  echo: false
---
```

### Human-readable text

This is a Quarto document based on an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. When you click the **Render** button a document will be generated that includes both content as well as the output of any embedded code chunks within the document. You can embed an code chunk like this:

```
{r cars}
summary(cars)
```

### Including code chunks

You can embed plots, for example:

```
{r pressure}
#| echo = FALSE
plot(pressure)
```

---

## Licensing

### Code

often appropriate: [MIT license](#)

### Other materials

---

## { Which of the following best describes your situation? }



### I need to work in a community.

Use the [license preferred by the community](#) you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to [add a license](#).



### I want it simple and permissive.

The [MIT License](#) is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

[Babel](#), [.NET](#), and [Rails](#) use the MIT License.



### I care about sharing improvements.

The [GNU GPLv3](#) also lets people do almost anything they want with your project, *except* distributing closed source versions.

[Ansible](#), [Bash](#), and [GIMP](#) use the GNU GPLv3.

## { What if none of these work for me? }

### My project isn't software.

[There are licenses for that.](#)

### I want more choices.

[More licenses are available.](#)

### I don't want to choose a license.

[Here's what happens if you don't.](#)

Figure 26: <https://choosealicense.com>

# License Features

Your choices on this panel will update the other panels on this page.

**Allow adaptations of your work to be shared?**



Yes     No     Yes, as long as others share alike

**Allow commercial uses of your work?**



Yes     No

Figure 27: <https://creativecommons.org/choose/>

## Data and methods

---

### Diversity in data and methods

CAIS: Forschung zu Digitalisierung und Digitale Gesellschaft  
research fields

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

### data and methods

- qualitative interviews
- text analysis
- quantitative surveys
- experimental designs

- social media studies
  - ...
- 

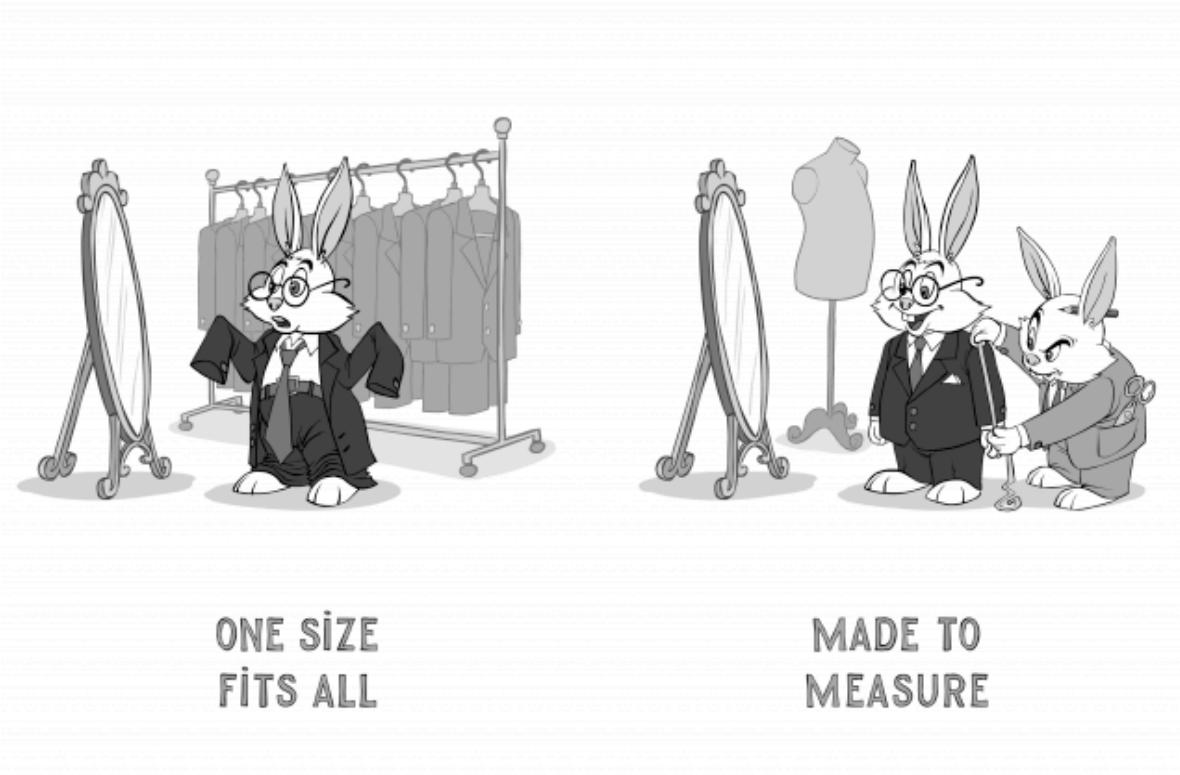


Figure 28: [source](#)

---

### Reasons to share your data

- To allow the possibility to fully **reproduce** a scientific study.
- To prevent duplicate efforts and **speed up scientific progress**. Large amounts of research funds and careers of researchers can be wasted by only sharing a small part of research in the form of publications.
- To facilitate **collaboration** and increase the impact and quality of scientific research.
- To make results of research openly available as a **public good**, since research is often publicly funded.

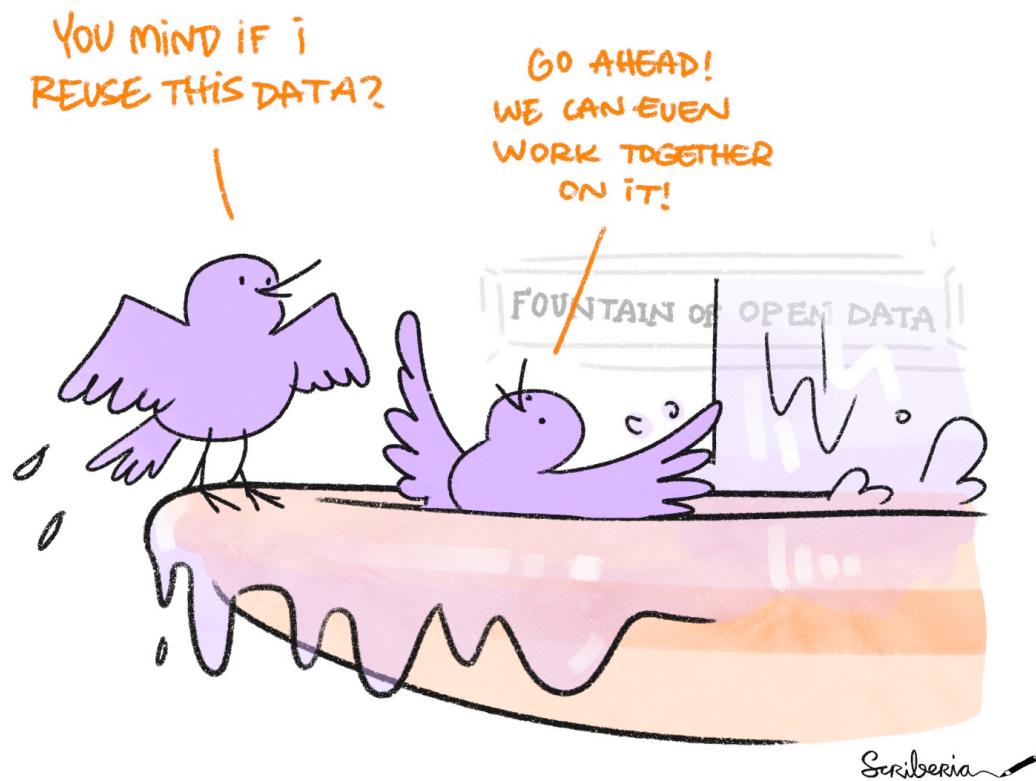


Figure 29: [source](#)

## FAIR data

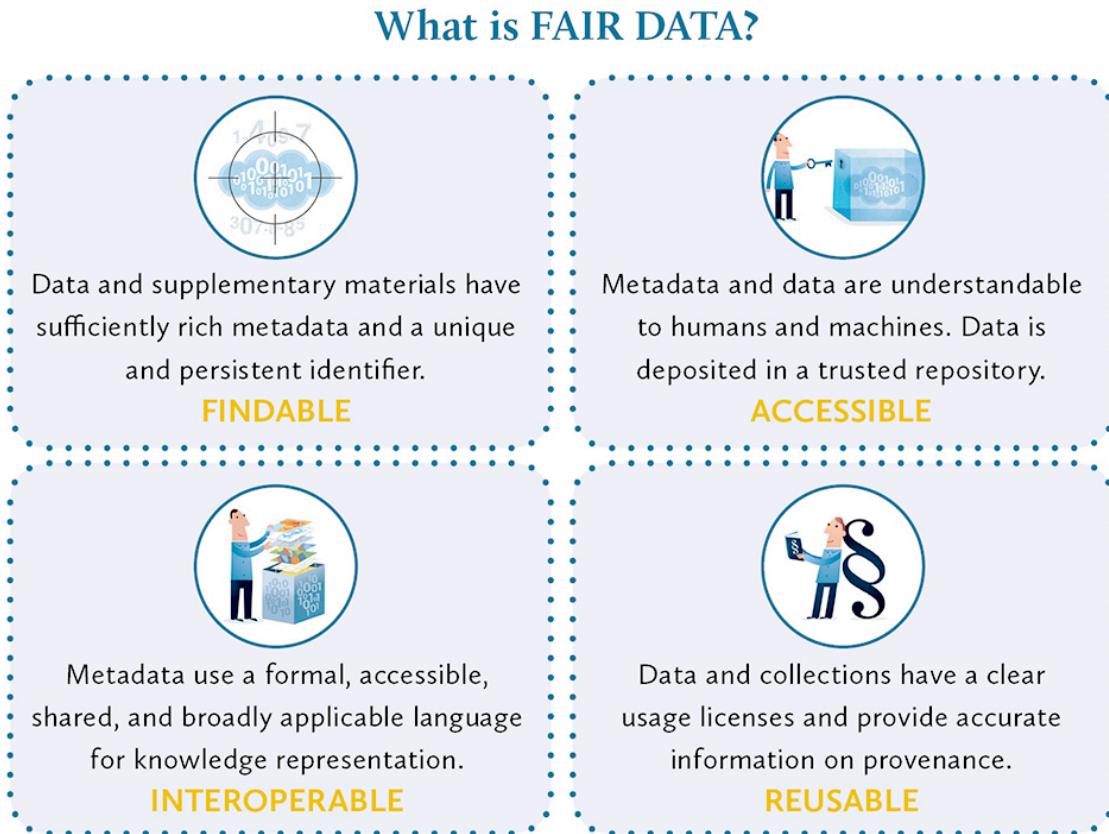


Figure 30: [source](#)

## How to share your data

### [Turing Way tutorial](#)

- Step 1: Select **what data** you want to share; eg.:
  - ethical concerns
  - commercial concerns

- Step 2: Choose a **data repository** or other sharing platform
    - overview: [re3data](#), [NIH](#), [FAIRsharing](#)
    - examples: [Zenodo](#), [Dryad](#)
  - Step 3: Choose a **licence** and link to your paper and code; e.g.:
    - [Creative Commons](#)
    - [Open Data Commons](#)
  - Step 4: **Upload** your data and documentation
    - good file organisation
    - appropriate file formats (e.g. `csv` > `xlsx`)
- 

## Sharing social media data

**ORIGINAL RESEARCH article**

Front. Big Data, 16 January 2023  
Sec. Data Mining and Management  
Volume 5 - 2022 | <https://doi.org/10.3389/fdata.2022.971974>

This article is part of the Research Topic  
Social Recommendations and Applications for Metaverse  
[View all 3 Articles >](#)

**Download Article**

**1,676** Total views   **480** Downloads   **1** Citations   ⓘ

[View article impact >](#)

**4** View altmetric score >

**Edited by**

 **Ingmar Weber**  
Saarland University, Germany

**Reviewed by**

 **Katja Mayer**  
University of Vienna, Austria

 **Lucia Ratiu**  
Babeş-Bolyai University, Romania

 **Tamara Gajic**  
Geographical Institute, Ljubljana University

**Sharing social media data: The role of past experiences, attitudes, norms, and perceived behavioral control**

Esra Akdeniz<sup>1\*</sup>  Kerrin Emilia Borschewski<sup>1</sup>  Johannes Breuer<sup>2,3</sup>  Yevhen Voronin<sup>1†</sup> 

<sup>1</sup> Data Services for the Social Sciences, GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany  
<sup>2</sup> Survey Data Curation, GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany  
<sup>3</sup> Center for Advanced Internet Studies (CAIS), GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany

Social media data (SMD) have become an important data source in the social sciences. The purpose of this paper is to investigate the experiences and practices of researchers working with SMD in their research and gain insights into researchers' sharing behavior and influencing factors for their decisions. To achieve these aims, we conducted a survey study among researchers working with SMD. The questionnaire covered different topics related to accessing, (re)using, and sharing SMD. To examine attitudes toward data sharing, perceived subjective norms, and

Figure 31: [source](#)

## Obstacles to data-sharing

- Reason 1: Preparing data for sharing is **resource-intensive**
- Reason 2: Not enough **credit** for data sharing
- Reason 3: Lack of **confidence** and **knowledge**
- Reason 4: Data protection **laws**
- Reason 5: Platform **terms of service**
- Reason 6: **Copyright**
- Reason 7: Informed **consent**
- Reason 8: **Ethical** challenges
- Reason 9: Lack of common **standards**

[source](#)

paper on incest

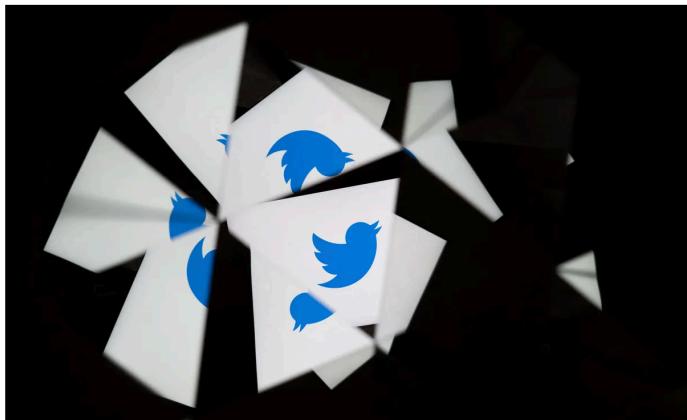
---

## The case of Twitter

- stage 1: access costly & legal grey area for scraping
- stage 2: Research API
- stage 3: Elon Musk → X → ...

SCIENCE / TWITTER - X / TECH

## Twitter just closed the book on academic research



/ Twitter was once an indispensable resource for academic research. That's changed under Elon Musk.

By [Justine Calma](#), a science reporter covering the environment, climate energy with a decade of experience. She is also the host of the Hell podcast.

May 31, 2023, 3:19 PM GMT+2 | □

Figure 32: [source](#)

---

## Data and methods

In groups of shared interests and expertise, make a **digital poster** about challenges and possibilities of working with data in your field of study.

For example, this could tackle issues like collecting, publishing, and sharing data.

Try to make it as concrete and constructive as possible.

You have until **14:20 h** to make the poster.

After that, each group will briefly present their poster.

---

## 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 33: [source](#)

---

---

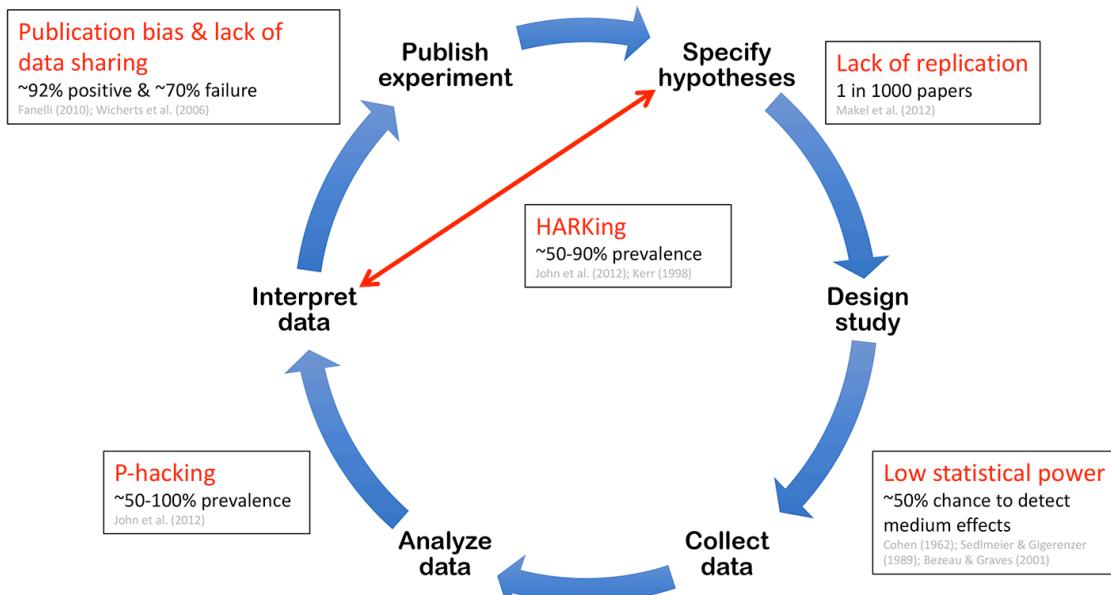


Figure 34: [source](#)

## 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

...

## 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
- 

## **Resources**

- DRA
- The Turing Way
- Data Carpentries