The German Reproducibility Network

A Strategic Community Effort to Promote Transparent Research Practices in the Scientific System

deRSE24 – Workshop on FAIR and reproducible Code

Mission

Increasing Trustworthiness and Transparency of Scientific Research

The Goal.

Science we can trust.

One Problem.

Change coming mostly from the bottom up.

Our Approach.

Trigger system-wide change by connecting different stakeholder groups.

Activities

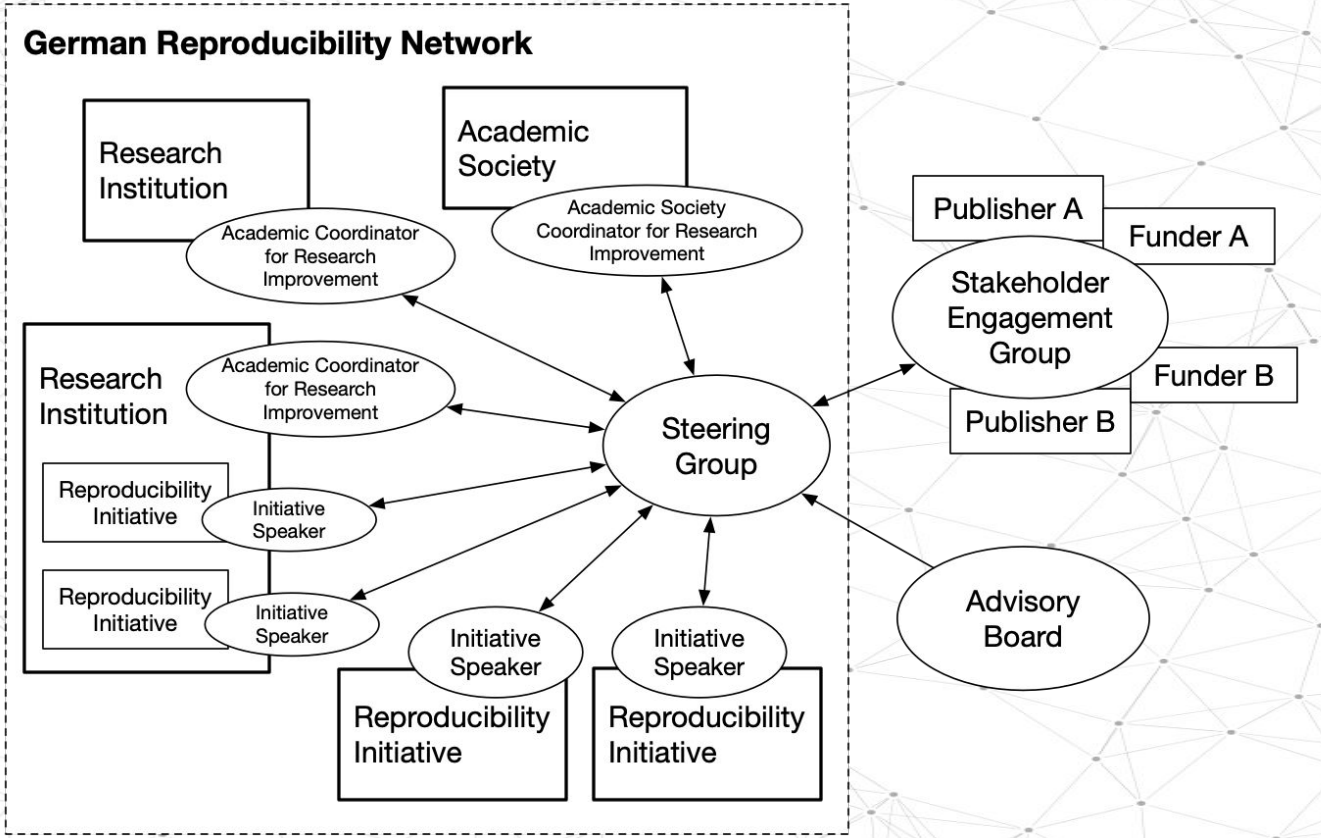
We **support researchers** in educating themselves about open science practices, and founding local open science communities.

We **connect local** or **topic-centered Reproducibility Initiatives** to a national network, and foster connections between them.

We **advise institutions** on how to embed open science practices in their work.

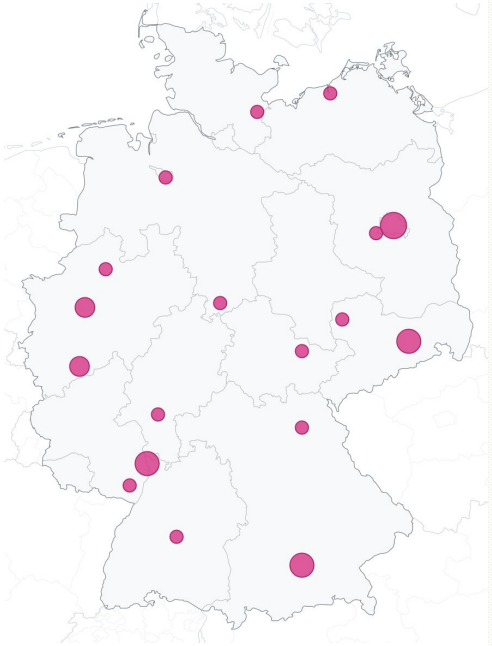
We **represent the open science community** toward other stakeholders in the wider scientific landscape.

Structure


German Reproducibility Network
Research Institution
Academic Society
Academic Coordinator for Research Improvement
Academic Society Coordinator for Research Improvement
Research Institution
Academic Coordinator for Research Improvement
Reproducibility Initiative
Initiative Speaker
Reproducibility Initiative
Initiative Speaker
Steering Group
Initiative Speaker
Reproducibility Initiative
Initiative Speaker
Reproducibility Initiative
Publisher A
Funder A
Stakeholder Engagement Group
Funder B
Publisher B
Advisory Board

Members

https://reproducibilitynetwork.de/members/



Members

Steering Group

Tracy Weissgerber

BIH QUEST-Center @ Charité Berlin

Gordon Feld

Central Institute of Mental Health in Mannheim

Maximilian Frank

LMU Munich & Psychologie-Fachschaften-Konferenz (PsyFaKo)

Rima-Maria Rahal

Network of Open Science Initiatives

Antonia Schrader

Helmholtz Association Open Science Office

Ulf Toelch

BIH QUEST-Center @

Peter Steinbach

Helmholtz AI,

Tobias Schlauch

Deutsches Zentrum

Tina Lonsdorf

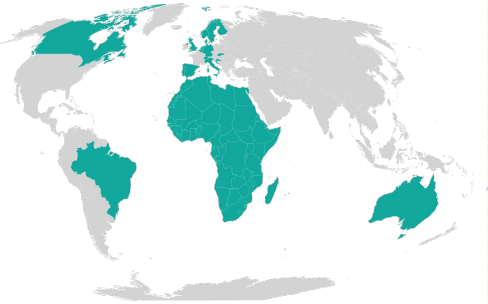
Universität Bielefeld

Heidi Seibold

Open Science

Global RNs

www.ukrn.org/global-networks



UK RN

RN UK UK Reproducibility Network

About | Community | Resources | Activities | Join UKRN | Contact

Global Networks

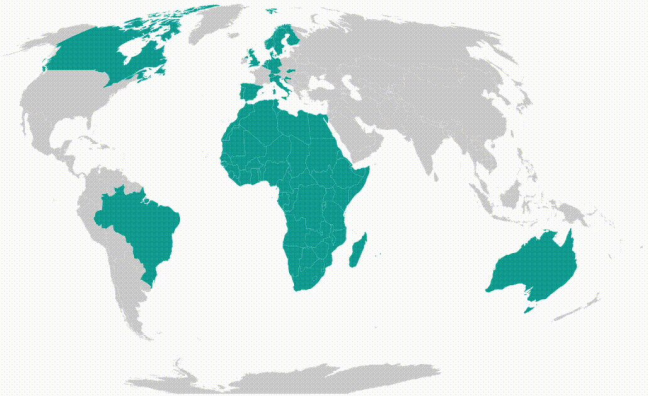
Outside the UK? Find a Reproducibility Network in your area

See full Global Networks Statement

Global Reproducibility Networks

A Reproducibility Network (RN) is a national, peer-led consortium of researchers that aims to promote and ensure rigorous research practices by establishing appropriate training activities, designing and evaluating research improvement efforts, disseminating best practice and working with stakeholders to coordinate efforts across the sector. RNs aim for broad disciplinary representation and an intensive interdisciplinary dialogue (e.g., with funding agencies, publishers, learned societies and other sectoral organisations, as well as researchers from all disciplines and across all career stages).

To reach as many researchers as possible, and to operate as efficiently as possible, we are keen to support other countries interested in creating similar networks. If you are interested in setting up a national RN, or finding out who in your country is working towards this, please contact us.



Get in touch!

http://reproducibilitynetwork.de

@GermanRepro@mastodon.world
@GermanRepro

info@reproducibilitynetwork.de

deRSE24 - Workshop on FAIR and reproducible Code

Why?

Code plays an important role when it comes to reproducibility.

Let's work together to improve the quality of research!

Topic	Details	Speaker	Duration (min)	Start
Arrival + Welcome	Welcome (from GRN+HIFIS)	Tobias	5	09:30
Introduction	The GRN and the idea of the workshop	Heidi	10	09:35
Talk session	Short talks introducing the 4 topics (5min + questions)	Heidi, Michael, Tobias, Max	40	09:45
Break	—	—	20	10:25
Intro: World Cafe	How the world cafe works	Heidi	5	10:45
World Cafe	Participants can go to tables with experts on the 4 topics (15 minutes on each table)	all	65	10:50
Wrap-Up	Wrap-up (1 minute highlight per table)	Tobias (+ Heidi, Max, Michael)	5	11:55

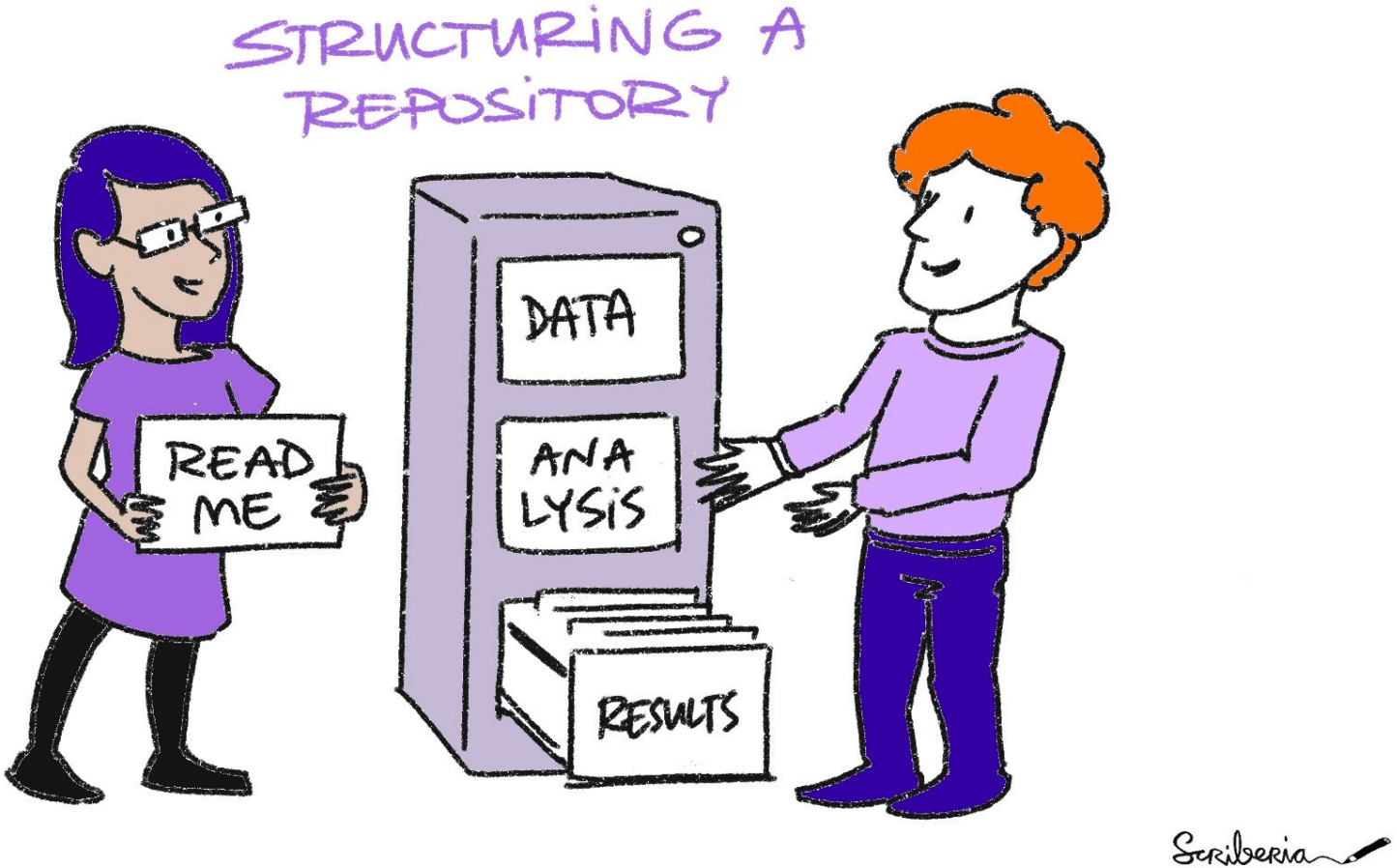
EXPERT SLIDES

Setting up a FAIR and reproducible project repository

Heidi Seibold

Topics

- The README
- Folder structure and good naming for reproducibility and seamless collaboration
- Making contributions possible
- ...



https://the-turing-way.netlify.app/project-design/project-repo

Good organisation … starts simple

- Nice file organisation
- Good naming

```
.
├── README.md
├── 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
```

See https://github.com/HeidiSeibold/research-project-template

More templates

Code-focused

- Python cookiecutter template: github.com/Materials-Data-Science-and-Informatics/fair-python-cookiecutter
- R devtools: devtools.r-lib.org

More general

- GIN-Tonic standardized research folder structure
- The Turing Way reproducible project template
- R analysistemplates from the MPIP code club
- The TIER Protocol
- R user project `template` by Frederik Aust & Marius Barth
- R project template by start your lab

Naming

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

See slides by Jenny Brian

Naming

- ➔ **Machine readable** 💻
- ➔ **Human readable** 🧐
- ➔ **Consistent** ⚙️
- ➔ Optional: **Play well with default ordering** ⬇️

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`

See slides by Jenny Brian

deRSE24 - Workshop on FAIR and reproducible Code

Stabilizing the computing environment

Michael Meinel (DLR / deRSE)

Computing Environment? Why should I care?

Your software is not on its own, but needs lots of stuff:

- A computer, where it is executed
- An operating system, that handles all the grunt stuff
- The programming language(s) of your choice
  (including Compiler, Linker, Interpreter, …)
- Third party tooling like software libraries, templates, code generation, …

… and all of this has an impact on the results, your software produces.

What is the impact of each?

Computer (ISA, usable extra hardware, …)

Good news: Compatible hardware produces compatible results … most of the time.

- Modern processors have microcode that might be used to fix bugs.
- Different accelerators (GPGPU, TPU) have different programming paradigms.
- Exploiting special hardware traits might result in unexpected behaviour…

What is the impact of each?

Computer (ISA, usable extra hardware, …)

Operating System

Portable code is a myth, even with VMs.

- While POSIX provides a standard, there is still lots of freedom for implementers.
- File systems are different, and so are the methods on how to access files.
- OS APIs change over time, get deprecated, differ in implementation details…

What is the impact of each?

Computer (ISA, usable extra hardware, …)

Operating System

Programming Language

Programming languages evolve over time, while mostly trying to retain backwards compatibility:

- The same syntax might still slightly change over time.
- Different compilers for the same language produce different machine code. The same compiler does so, too. E.g., if you apply optimizations.
- Even the semantics might change:

  Python2: print(1 / 5) → 0 # integer division
  Python3: print(1 / 5) → 0.2 # float division

What is the impact of each?

Computer (ISA, usable extra hardware, …)

Operating System

Programming Language

Third party tooling

Software libraries evolve over time, too:

- Bugs are fixed and results are now correct.
- New features are added, old features are removed.
- Programming paradigms might change so that the same code behaves different… (e.g., recent changes in Pandas indexing)

What to do about this?

Stabilize your computing environment!

Clearly declare, what your target hardware is.
Benchmark the hardware before running code.

Specify your target OS?
Try to avoid OS interaction?
Make the OS part of your software.

Be specific about your programming language.
Deliver binaries instead of code.

Lock your dependencies to specific version.
Bundle third party tools.

Software and data licenses

Tobias Schlauch (GRN / HIFIS / DLR)

Why should I care?

- **Copyright:**
  - Copyright protects the expression of an idea.
  - Copyright grants exclusive rights to the copyright holder.
  - Copyright works differently depending on the country.

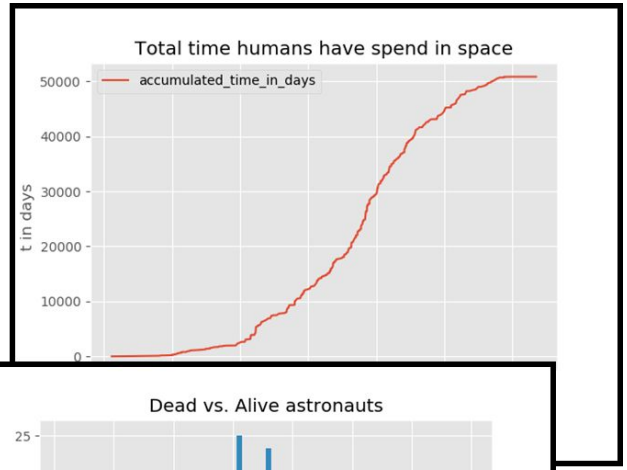
- **Software vs. data:**
  - Software: Software is protected by copyright
  - Data: Copyright requires “creativity”, data representing simple facts vs. curated datasets / databases

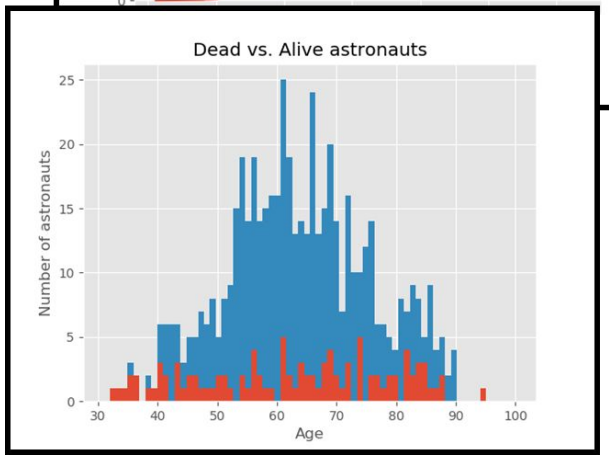
- **Without a proper license (re)-use of your software / data is problematic!**

Example: Astronaut Analysis

- **Astronaut Analysis is a data publication consisting of:**
  - Data set
  - Analysis script written in Python using pandas and matplotlib
  - Result plots

- **Scenario:**
  - I created it on my own as part of my job.
  - I want to publish it with my research paper.
  - I want to make its reuse as easy as possible and make it available under an open source license.





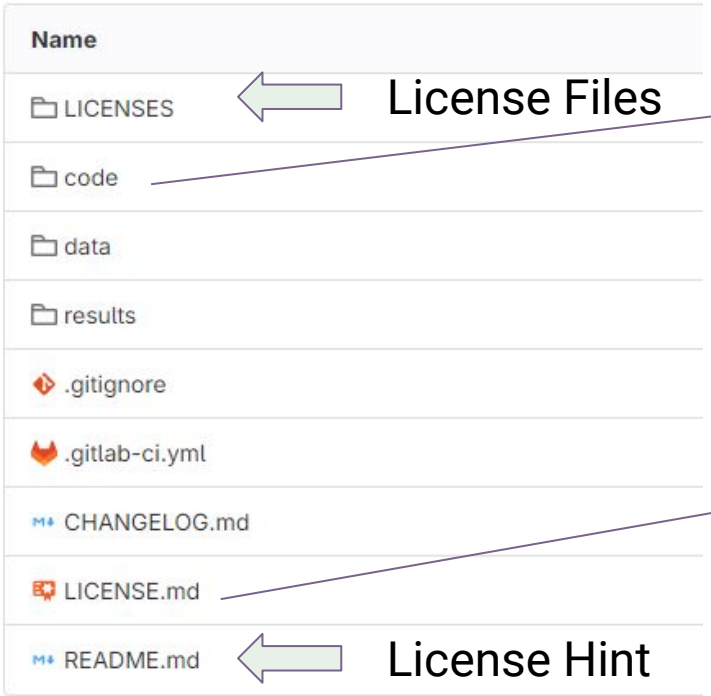
Example: Overall process

- **1.) Choose proper software and data licenses:**
  - Consider strategic aspects, https://choosealicense.com can help with software licenses
  - Check compatibility with the software license of your dependencies!
- **2.) Ask my boss for permission**
- **3.) Document copyright and license information**
  - Minimum: Add license file(s) and state the copyright holder(s)
  - Recommended: REUSE Software

Choices for the example:

Source code:	Apache-2.0
Data set:	CC0-1.0
Docs and plots:	CC-BY-4.0
Insignificant files:	CC0-1.0

Example: Result could look like this …




```
"""
SPDX-FileCopyrightText: 2024 German Aerospace Center (DLR)
SPDX-License-Identifier: Apache-2.0


This script analysis the astronaut data set and creates different
plots as result.
"""
```

License

Copyright © 2024 German Aerospace Center (DLR)

This work is licensed under multiple licenses:

- The data set is licensed under CC0-1.0.
- The source code and the accompanying material are licensed under Apache-2.0.
- The documentation and the resulting plots are licensed under CC-BY-4.0.
- Insignificant files are licensed under CC0-1.0.

Please see the individual files for more accurate information.

> **Hint:** We provided the copyright and license information in accordance to the REUSE Specification 3.0.

Some final recommendations

Make sure that every code that you use is covered by a license!

Make the license a criteria for your code dependencies selection!

Find out about your organizational processes!

Take care when combining code under different licenses!

Attach proper license information when publishing your code / data!

Ask for legal advice if you are unsure!

Topics

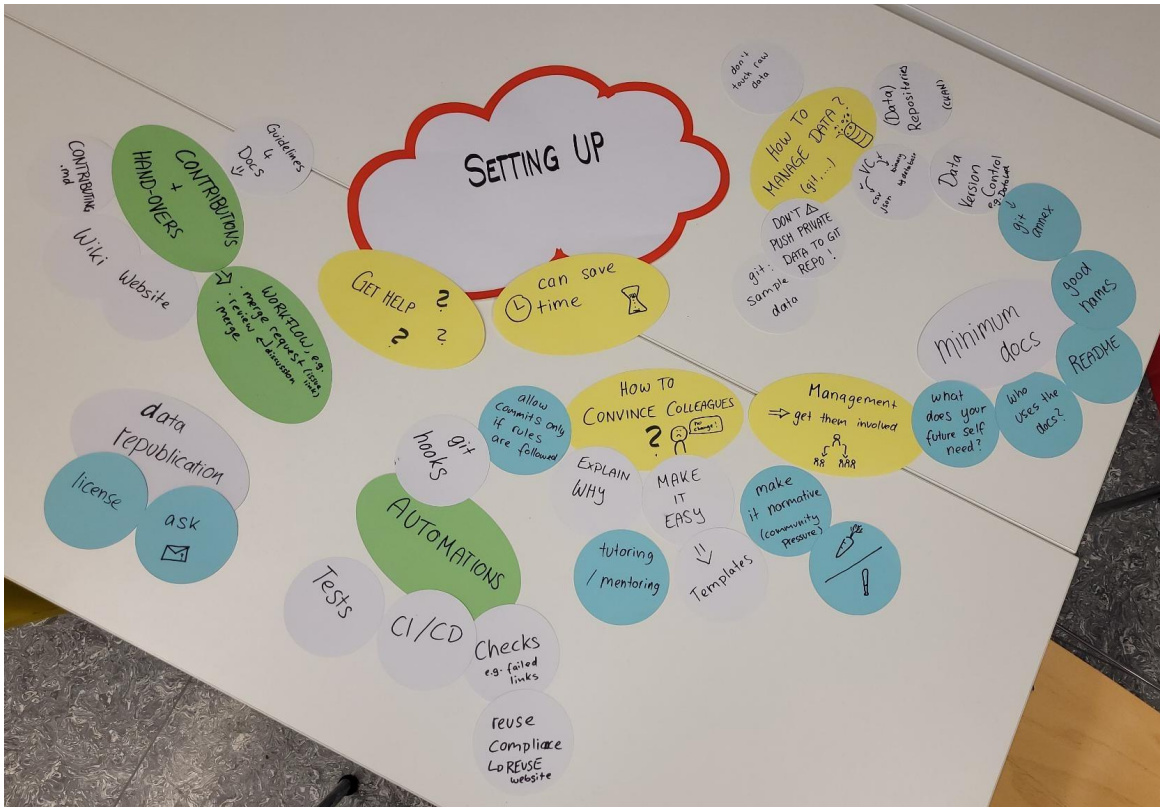
- Aspects of copyright
- Software vs. data licenses
- Organizational support processes
- Tools
- Challenges in context of AI models
- ...

World cafe

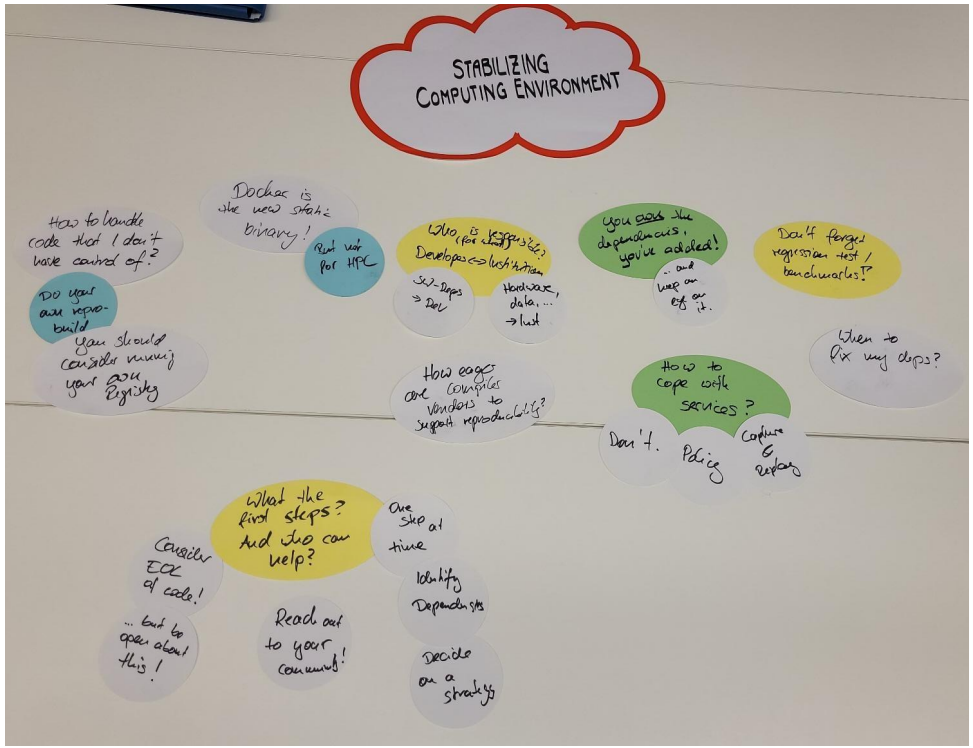
World cafe

- 3 topics = 3 world cafe tables
- 20 min per round / per table
- Bring your questions

Setting up a FAIR and reproducible project repository



Stabilizing the computing environment



Software and data licenses