Open source, open data.

The two pillars of this course

Winter 2026/27 · Timo Breuer · TH Köln

```
$ git clone https://github.com/owid/co2-data
Cloning into 'co2-data'...
$ cd co2-data
$ head -3 owid-co2-data.csv | cut -c1-40
country,year,iso_code,population,gdp,cem
Afghanistan,1750,AFG,2802560,,0.000,0.00
Afghanistan,1751,AFG,,,0.000,,,,,,,,,,,
$ wc -l owid-co2-data.csv
   50412 owid-co2-data.csv
$ grep -o "Creative Commons BY license" README.md
Creative Commons BY license
```

Who am I? Who are you?

WHO AM I?

Timo Breuer

Postdoctoral researcher in Prof. Schaer's team at TH Köln

Web	breuert.github.io
E-mail	timo.breuer@th-koeln.de
GitHub	github.com/breuert

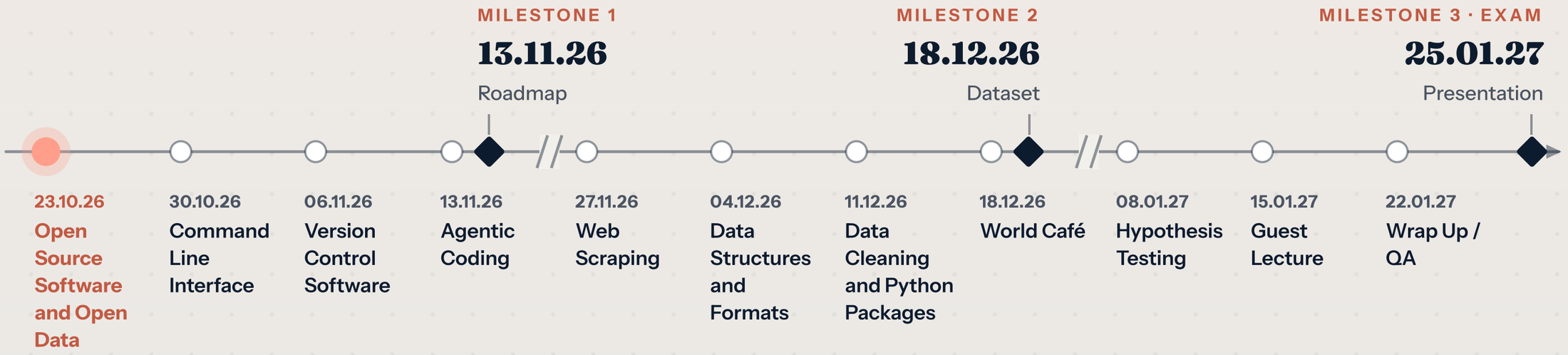
STUDENT JOBS IN OUR TEAM

Programming, data annotation and cleaning, literature research — 8–10 hours a week, flexible, maybe with a thesis topic. Apply: **apply@cir-group.org**

WHO ARE YOU? PLEASE RAISE YOUR HANDS.

1. Who studies Data and Information Science — and who studies something else?
2. Who wants to work as a data scientist, a data engineer, or a data analyst?
3. Who has written code before — in Python, R or any other language?
4. Who has used the command line or Git before?
5. Who owns a laptop — and who needs one for the labs?

Timeline


MILESTONE 1
13.11.26
Roadmap
MILESTONE 2
18.12.26
Dataset
MILESTONE 3 · EXAM
25.01.27
Presentation
23.10.26 Open Source Software and Open Data
30.10.26 Command Line Interface
06.11.26 Version Control Software
13.11.26 Agentic Coding
27.11.26 Web Scraping
04.12.26 Data Structures and Formats
11.12.26 Data Cleaning and Python Packages
18.12.26 World Café
08.01.27 Hypothesis Testing
15.01.27 Guest Lecture
22.01.27 Wrap Up / QA
Today
Upcoming lecture
Milestone / exam

The two pillars of this course

YOUR DATA PROJECT

Open source software

Tools anyone can inspect, change and share

- Linux
- bash
- Git
- Python
- Jupyter
- pandas
- NumPy

Open data

Data anyone can use, change and share

- GovData
- data.europa.eu
- OpenStreetMap
- Wikidata
- Our World in Data

Openness: transparent · reproducible · reusable

THE IDEA

Open tools and open data let anyone check, repeat and build on your results.

Almost every tool in this course is open source, and every project starts from open data. Today we look at both: what makes them open, which licences they use, and where to find them.

Why open matters: checking a famous result

2010 · REINHART & ROGOFF

The claim

Countries with public debt above 90% of GDP grow at **−0.1%** a year on average. The finding is quoted widely in the debate on austerity.

2013 · HERNDON, ASH & POLLIN

The replication

A graduate student gets the spreadsheet for a class assignment and finds a formula that skips five countries, plus debatable exclusions and weights. Corrected: **+2.2%**.

THE LESSON

Share data and code

Everyone makes mistakes. Hidden mistakes are the problem: with open data and open code, others can find them.



Average growth above 90% debt, published vs. corrected

Reinhart & Rogoff (2010), Growth in a Time of Debt, AER 100(2) · Herndon, Ash & Pollin (2014), Cambridge Journal of Economics 38(2)

Scope of today's lecture and roadmap

01
About this course

02
Open source software

03
Software licences

04
GitHub & Markdown

05
Open data

06
Data licences

07
FAIR data

08
Finding open data

What is data integration?

> **Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data.**
>
> Maurizio Lenzerini (2002), Data Integration: A Theoretical Perspective

IN THIS COURSE YOU WILL

- collect data from open sources and the web
- clean, structure and combine it
- analyse it, test hypotheses and present results

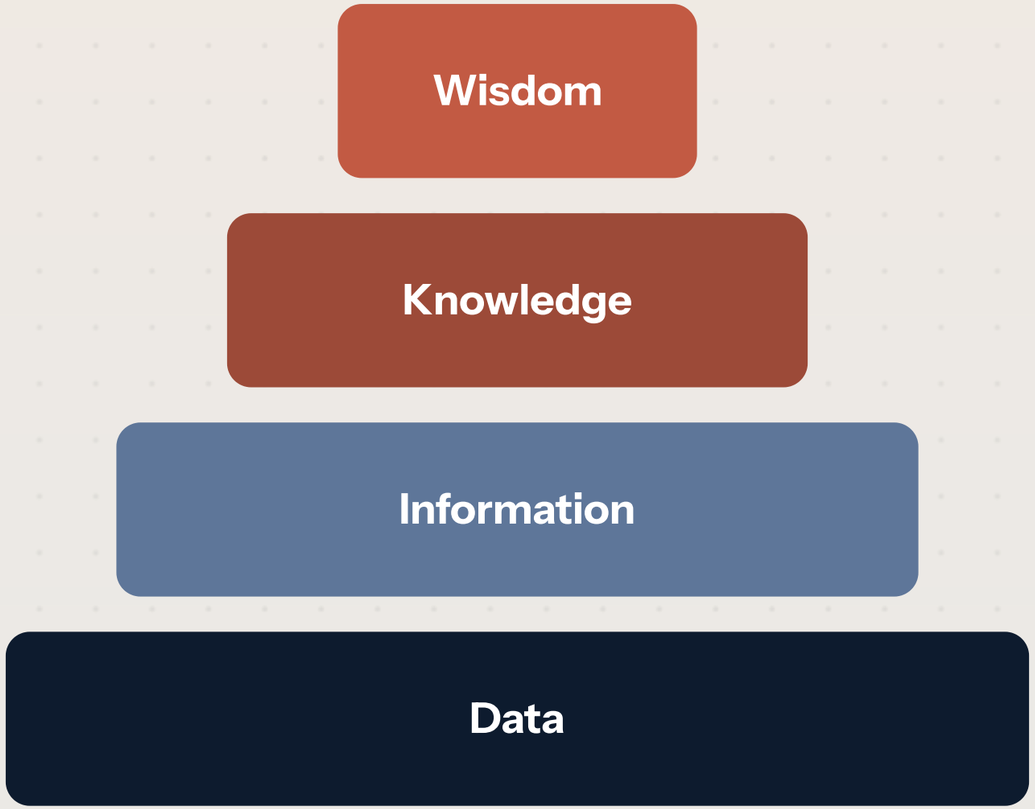
Real questions need more than one dataset.

Traffic accidents and weather, medals and GDP: the answer lies in combining sources.

THE CLASSIC PATTERN: ETL



From data to wisdom: the DIKW pyramid



AN EXAMPLE

Wisdom

Build bike lanes where people ride on warm days.

Knowledge

People cycle more when it is warm.

Information

Daily bike counts at one station, next to the temperature.

Data

`412, 380, 1024 · 7.5, 6.1, 18.2`

After Russell L. Ackoff (1989), From data to wisdom, Journal of Applied Systems Analysis 16 · the numbers are illustrative

Your pipeline, lecture by lecture

O
Obtain
Find and collect data

Open data · today

Web scraping · 27.11.

→

S
Scrub
Clean and structure it

Data cleaning and data structures · 04.12.

→

E
Explore
Look for patterns

Data science packages · 11.12.

→

M
Model
Test your hypotheses

Hypothesis testing · 08.01.

→

N
iNterpret
Explain and present

Your presentation · 25.01.

TOOLS FOR EVERY STEP

Command line 30.10. **Version control with Git** 06.11. **Agentic coding** 13.11.

OSEMN after Hilary Mason and Chris Wiggins (2010), A Taxonomy of Data Science · see also datascience-pm.com/osemn

How this course works

1

Lectures

Nine lectures in three blocks, a guest lecture and a wrap-up with Q&A — on Fridays from 23.10.26 to 22.01.27.

2

Labs

One lab per lecture, nine in total. Practise what you learned — completed labs earn bonus points.

3

Group project

Your own data pipeline on a topic of your choice. Three milestones — the presentation on 25.01.27 is the exam.

THREE BLOCKS, THREE MILESTONES

23.10. – 13.11.26

Tools, data and software engineering

Open source and open data · command line · Git · agentic coding

◆ **Milestone 1: roadmap**

27.11. – 18.12.26

Data engineering

Web scraping · data cleaning and structures · data science packages · World Café

◆ **Milestone 2: dataset**

08.01. – 25.01.27

Data analysis and presentation

Hypothesis testing · guest lecture · wrap-up and Q&A

◆ **Milestone 3: presentation**

Where to find everything

COURSE REPOSITORY

github.com/breuert/data-integration

Schedule, slides and labs for every lecture. **Watch** the repository to get notified about updates.

`README.md · schedule` `slides/` `labs/01/ … labs/09/`

QUESTIONS?

Ask early, ask in the open

- Open an **issue** on GitHub — others may have the same question.
- Discuss with your peers in the lab sessions.
- In case of emergency: **timo.breuer@th-koeln.de**

LANGUAGE MATTERS

Slides, labs and materials are in English. Your answers, documentation and presentation may be in **English or German**.

LANDING PAGE

`breuert.github.io/data-integration/`

MOODLE COURSE

`elearning.iws.th-koeln.de/moodle/course/view.php?id=3051`

Group project: your own data pipeline

In a team, you choose a topic you care about — politics, the environment, public health, sports, culture. You start from **open datasets**, enrich them with data you collect from the web, and analyse them with the tools of this course.

WHAT YOU DELIVER

Dataset
open data, enriched and combined

Pipeline
code from collection to analysis

Repository
documented and reproducible

Presentation
your findings, in the exam

TOPICS FROM PREVIOUS YEARS

Football
Market value vs. Champions League games · wins vs. weather · when do most goals fall?

Olympics
Medals vs. gross domestic product

Video games
Sales across platforms

Movies
Trends and patterns in IMDb data

Milestones and deliverables

Date	Milestone	Deliverables	Format
13.11.26	**1 · Roadmap**	Topic and dataset selection · research questions · roadmap	Markdown documentation in your code repository
18.12.26	**2 · Dataset**	Enriched and augmented dataset · pipeline prototype (skeleton code)	New dataset hosted online · Jupyter notebook in your repository
25.01.27	**3 · Presentation**	Presentation of your results · code repository and demo · project documentation	Exam: presentation · running notebook · open repository and documentation

> **Each milestone builds on the previous one — and today's topics are the start: an open dataset, a GitHub repository and a README in Markdown.**

Working on your project

THE TIMELINE

This project lasts the whole semester.

- Start now: form a team and look for a topic.
- Work on it continuously — commit early and often.
- Your repository is graded as it is at the deadline before the exam. Later commits don’t count.

YOUR RESULTS LIVE IN A GIT REPOSITORY ON GITHUB

1. One team member creates the repository — preferably in a GitHub organisation.
2. Every team member is invited as a collaborator and contributes.
3. Invite me as a collaborator: **breuert**.
4. Send me an e-mail with the repository link and the names of all team members.

Grading

Your grade is based on the group project — code, documentation and presentation — and is determined by the total number of points:

Points	<50	50	55	60	65	70	75	80	85	90	95
Grade	n.b.	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0

TEN CRITERIA — DETAILS ON THE NEXT SLIDE

- Dataset and motivation
- Documentation
- Repository organisation
- Version control and collaboration
- Web scraping
- Data cleaning
- Data analysis
- Hypothesis testing and modelling
- Visualisation and presentation
- Q&A

BONUS POINTS

+10

for passing the World Café on 18.12.26

Grading rubric

01 **Dataset and motivation**
Why this dataset? Relevance for the research question, limitations, a sensible size and scope.

02 **Documentation**
A structured README with the report: objectives, methods, data sources, how to reproduce.

03 **Repository organisation**
Logical folders (data/, src/, notebooks/, docs/), raw vs. processed data, reusable code.

04 **Version control and collaboration**
Frequent, meaningful commits and messages; branches, issues and pull requests.

05 **Web scraping**
Robust and ethical scraping that handles errors and changing pages, well documented.

06 **Data cleaning**
Missing values, duplicates, outliers and formats handled — with reasons for each choice.

07 **Data analysis**
Beyond summaries: descriptive statistics, comparisons and patterns tied to your questions.

08 **Hypothesis testing and modelling**
Clear hypotheses, suitable tests, systematic trends; assumptions and limits discussed.

09 **Visualisation and presentation**
Clear, labelled charts that support the story; a logical, polished presentation.

10 **Q&A**
Confident answers that show you understand methods, assumptions and limitations.

Generative AI in this course

You may use large language models and coding agents — I encourage you to. But you are learning to code, so there are rules.

1. Try to solve the problem on your own first — and document what you tried.
2. Then ask the model for a solution or a code review.
3. Document the tool and model, your prompt and the output in `AI_USAGE.md`.
4. Compare its answer with yours: what did you accept, change or reject — and how did you check it?
5. You own every line: each team member must be able to explain the code.

WHAT IS VIBE CODING?

You describe what you want and accept the generated code without reading it. The feedback loop is the program's output only.

Fun — but not how you learn programming.

Good prompts need the right concepts and terms: the better you code, the better your results. More in lecture 4 — **Agentic Coding**, 13.11.26.

How to succeed in this course

Use GitHub

Raise issues and discuss your questions there.

Collaborate

Communicate and work together with your peers.

Be curious

Your ideas and suggestions are very welcome.

Take part

Join lectures and labs actively.

Keep up

Prepare and follow up every session.

FREE BOOKS, OPENLY LICENSED

The Linux Command Line — CC BY-NC-ND
William Shotts · Lecture 2
`linuxcommand.org/tlcl.php`

Pro Git — CC BY-NC-SA
Scott Chacon and Ben Straub · Lecture 3
`git-scm.com/book`

Automate the Boring Stuff with Python — CC BY-NC-SA
Al Sweigart · Python basics
`automatetheboringstuff.com`

Python Data Science Handbook — CC BY-NC-ND
Jake VanderPlas · Lectures 6, 7
`jakevdp.github.io/PythonDataScienceHandbook`

Creative Commons licences: BY = attribution, NC = non-commercial, ND = no derivatives, SA = share-alike — more on those later today.

What is open source software?

> **Open source software comes with its source code and a licence that lets anyone use, study, change and share it.**

SOURCE CODE: WHAT PEOPLE WRITE

```
printf("Hello, world!\n");
```

BINARY: WHAT MACHINES RUN

```
7f 45 4c 46 02 01 01 00 00 00
```

Without the source, you can run a program — but not see or change what it does.

THE LICENCE GRANTS

- Use it for any purpose
- Study how it works
- Change it
- Share it — changed or not

Open ≠ free of charge ≠ public

The licence makes software open. Code you can read on GitHub is not open source unless a licence says so — and open source may be sold.

Free software: the four freedoms

0

FREEDOM 0

Run

The freedom to run the program as you wish, for any purpose.

1

FREEDOM 1

Study and change

The freedom to study how the program works, and change it so it does your computing as you wish.

2

FREEDOM 2

Share

The freedom to redistribute copies so you can help others.

3

FREEDOM 3

Share your changes

The freedom to distribute copies of your modified versions to others.

Freedoms 1 and 3 need access to the source code.

> **“You should think of ‘free’ as in ‘free speech,’ not as in ‘free beer.’”**

Free Software Foundation, gnu.org/philosophy/free-sw.html · the FSF was founded by Richard Stallman in 1985

The Open Source Definition

1. **Free redistribution**
   Anyone may give it away or sell it, royalty-free.
2. **Source code**
   The source must be included or easy to get.
3. **Derived works**
   Changes and derived works must be allowed.
4. **Integrity of the author's source code**
   Changes may have to ship as patches or under a new name.
5. **No discrimination against persons or groups**
   The licence applies to everyone.
6. **No discrimination against fields of endeavor**
   Business, research, anything: all uses allowed.
7. **Distribution of license**
   Everyone who receives the program gets the same rights.
8. **License must not be specific to a product**
   The rights don't depend on a particular bundle.
9. **License must not restrict other software**
   Software shipped alongside may be closed.
10. **License must be technology-neutral**
    No clause may require a specific technology or interface.

Open Source Initiative (OSI), opensource.org/osd — derived from the Debian Free Software Guidelines

From proprietary to public domain

← closed open →



Proprietary

Closed source, use restricted by a licence you buy or accept.

Microsoft Office, Adobe Photoshop

Freeware

Free of charge — but closed. You may use it, not study or change it.

Adobe Acrobat Reader, Google Chrome

Source-available

You may read the code, but its use is restricted.

MongoDB (SSPL), Terraform (BSL)

Open source

Use, study, change and share — under a licence with conditions.

Linux, Python, Firefox, LibreOffice

Public domain

No rights reserved at all.

SQLite

Chrome is freeware — but it is built on **Chromium**, which is open source.

A short history of open source

1983

GNU project

Richard Stallman sets out to build a free Unix.

1985

Free Software Foundation

The home of free software and the GPL.

1989

GNU GPL

The first copyleft licence.

1991

Linux

Linus Torvalds posts a hobby kernel.

1998

“Open source”

The term is coined; Netscape opens its browser’s code; the OSI is founded.

2005

Git

Torvalds writes a version control system for Linux.

2008

GitHub

Code hosting for everyone — pull requests go mainstream.

2018

Microsoft buys GitHub

For US$ 7.5 billion — once an opponent of open source.

2024

Open Source AI Definition

The OSI defines what open means for AI models.

Eric S. Raymond’s essay “The Cathedral and the Bazaar” (1997) describes the bazaar-style development of Linux — and inspired Netscape’s decision.

What does open mean for AI models?

AN AI MODEL IS MORE THAN CODE

Code	to train the model and to run it
Parameters	the weights learned in training
Data information	enough detail about the training data to build a substantially equivalent system

The **Open Source AI Definition 1.0** (OSI, 2024) asks for all three under open terms — so that anyone may use, study, modify and share the system.

FROM CLOSED TO FULLY OPEN

Closed

ChatGPT, Claude, Gemini

Use it through an app or an API — nothing to download.

Open weights, restricted

Meta's Llama

Its licence restricts who may use it — not open source, says the OSI.

Open weights, open licence

Mistral's Mixtral

Weights under Apache-2.0, but the training data stays undisclosed.

Fully open

Ai2's OLMo, EleutherAI's Pythia

Data, code and weights: they pass the definition.

opensource.org/ai · validation results: OSI, 2024 end-of-year review of the Open Source AI Definition

Why open source?

01

Transparency

Read the code: see what a program really does — with your data, too.

02

Reproducibility

Anyone can rerun your analysis with the same tools and versions.

03

No lock-in

Your work does not depend on one vendor, its prices or its future.

04

Community

Documentation, forums, fixes: millions of people help each other.

05

Security

Many eyes can find bugs — if someone actually looks.

06

Learning

Read code written by experts, and contribute back.

> For science and data work, **reproducibility** is the killer feature.

Open source is everywhere

98%

of commercial codebases contain open source

Black Duck OSSRA report 2026

$8.8 trn

value of widely used open source — firms would spend 3.5× more on software without it

Hoffmann, Nagle & Zhou (2024), Harvard Business School

100%

of the world's 500 fastest supercomputers run Linux

TOP500 list, since November 2017

180 m+

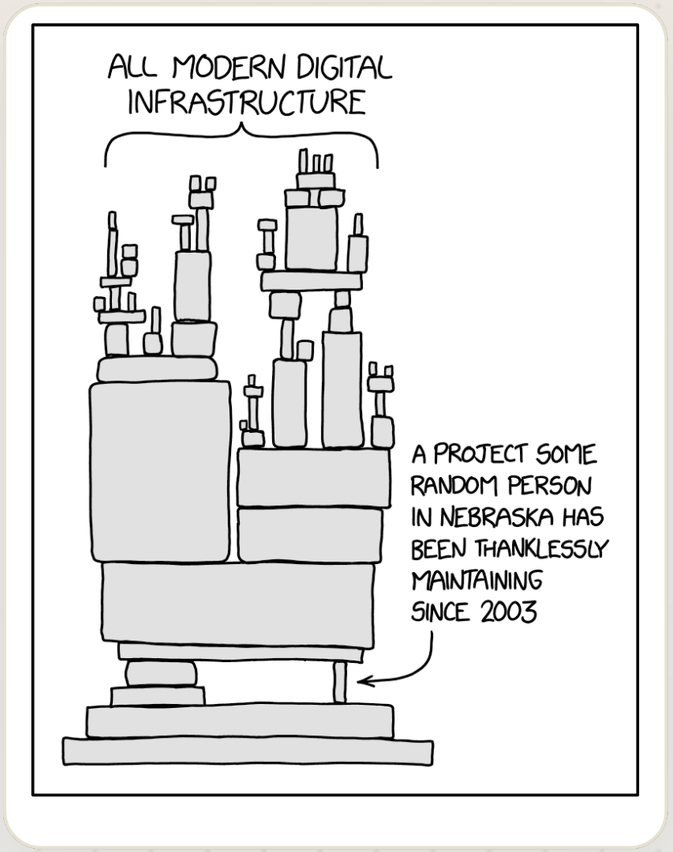
developers on GitHub

GitHub Octoverse 2025

EVEN THESE SLIDES

Made with **bento** (MIT licence), which bundles **reveal.js** (MIT) and the fonts **Fraunces** and **Instrument Sans** (SIL Open Font Licence). Open the HTML file: the licence notices are in there.

When a tiny project holds up the world



xkcd.com/2347 · CC BY-NC 2.5

2014 **Heartbleed**

A bug in OpenSSL — run by volunteers, with one full-time developer — leaked passwords and keys from servers all over the web.

2016 **left-pad**

A developer removed an 11-line package from npm — and the builds of countless projects failed for hours.

2021 **Log4Shell**

A flaw in Log4j, a Java logging library, made millions of servers attackable.

2024 **xz utils**

A fake contributor spent two years building trust, then planted a backdoor. Found by chance: SSH logins took half a second longer.

> **Open source is free to use — but somebody has to maintain it.**

Who keeps open source alive?

Volunteers	Most projects are run by individuals in their spare time.
Companies	Firms pay developers to work on the open source they depend on.
Foundations	Linux Foundation, Apache Software Foundation, NumFOCUS — home of NumPy, pandas and SciPy.
Donations	GitHub Sponsors, Open Collective: money straight to maintainers.
Public funding	Germany's Sovereign Tech Agency invests in the maintenance of open digital infrastructure.

WHAT YOU CAN DO

- Report bugs — with a minimal example.
- Improve the documentation: typos count.
- Answer questions of other users.
- Cite the software you use in your work.
- Say thank you. Maintainers rarely hear it.

How open source projects work

PEOPLE

Users

use the software, report bugs

Contributors

send fixes, docs, features

Maintainers

review, merge, release, decide

FILES EVERY PROJECT HAS

`README.md`

what it is and how to use it

`LICENSE`

what you may do with it

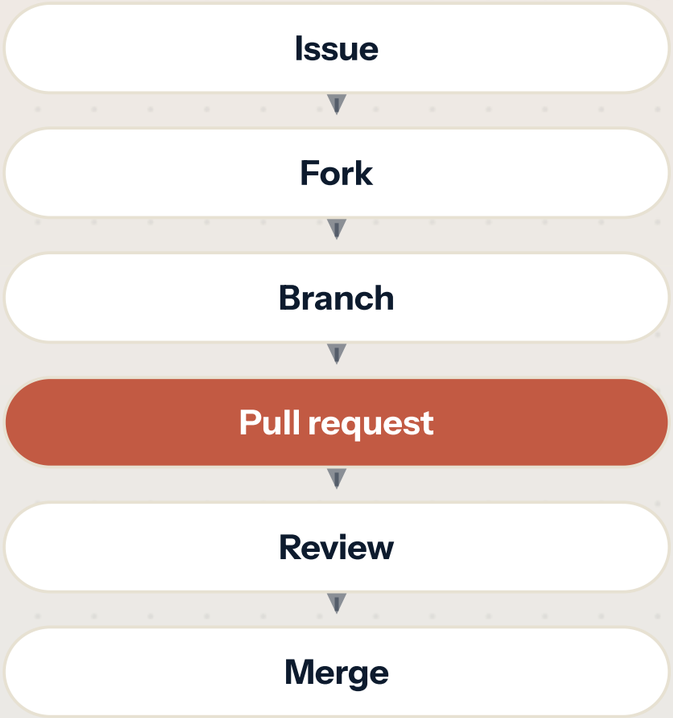
`CONTRIBUTING.md`

how to help

`CODE_OF_CONDUCT.md`

how we treat each other

HOW A CHANGE GETS IN



The Git side of this — branches, forks, pull requests — is lecture 3 on 06.11.26.

The open source stack of this course

Tool	What for	Lecture	Licence
bash	the shell	2	GPL-3.0-or-later
Git	version control	3	GPL-2.0-only
Python	programming language	all	PSF-2.0
Jupyter	notebooks	5 – 9	BSD-3-Clause
requests · Beautiful Soup	web scraping	5	Apache-2.0 · MIT
pandas · NumPy · SciPy	data analysis and statistics	6 – 9	BSD-3-Clause
VS Code	editor	all	MIT source · proprietary build
GitHub	hosting, collaboration	all	proprietary platform

THE FINE PRINT

Not everything is open.

- VS Code's source is MIT-licensed, the download from Microsoft is not. **VSCodium** is the fully open build.
- GitHub hosts open source, but the platform itself is closed. **Codeberg** and **GitLab** are open alternatives.
- Coding agents: some are open source, many are not.

No licence, no permission

Copyright is automatic. Without a licence, all rights are reserved — even if the code is public.

Whoever writes code owns the copyright from the first line — no registration needed. A licence is the owner's permission for everyone else.

PUBLIC CODE WITHOUT A LICENCE

- ✓ Read it on GitHub
- ✓ Fork it on GitHub — the terms of service allow this
- ✗ Copy it into your own project
- ✗ Change it and share your version
- ✗ Use it in a product

Add a `LICENSE` file to every repository you want others to use — choosealicense.com helps you pick one.

Permissive or copyleft?

fewer conditions for reusers

more protection for openness

Public domain

No conditions at all.

`CC0 · Unlicense`

Do anything.

Permissive

Keep the copyright and licence notice.

`MIT · BSD`
`Apache-2.0`

Reuse it in anything — even closed products.

Weak copyleft

Changes to the component itself stay open.

`LGPL · MPL-2.0`

Link it from closed code; share fixes to it.

Strong copyleft

Derived works stay open, under the same licence.

`GPL-2.0`
`GPL-3.0`

Distribute a derived program? Share all of its source.

Network copyleft

Also applies when you only offer a web service.

`AGPL-3.0`

Run a modified version online? Share its source.

Four common licences at a glance

	MIT	Apache-2.0	GPL-3.0	AGPL-3.0
Commercial use, change, share	✓	✓	✓	✓
Explicit patent grant	—	✓	✓	✓
Include copyright and licence	✓	✓	✓	✓
State your changes	—	✓	✓	✓
Disclose source when you distribute	—	—	✓	✓
Derived work under the same licence	—	—	✓	✓
Network use counts as distribution	—	—	—	✓

WHO USES WHAT?

MIT
React, Vue.js, Ruby on Rails

Apache-2.0
Kubernetes, TensorFlow, Spark

GPL
Linux, Git, bash

AGPL
Mastodon, Nextcloud

BSD-3-Clause
NumPy, pandas, Jupyter

Simplified after choosealicense.com/appendix — the licence text is what counts.

The MIT License, paragraph by paragraph

MIT License

Copyright (c) 2026 Your Name

WHO OWNS IT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

WHAT YOU MAY DO

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE ONE CONDITION

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

NO WARRANTY

THE CONDITION IN ACTION

These slides carry licence notices.

bento embeds reveal.js and other MIT-licensed code. So every saved deck includes their copyright and permission notices — exactly what paragraph three demands.

```
<!-- NOTICE — bento/slides
  MIT License …
```

Choosing a licence

I want it simple and permissive.	`MIT`
…and with an explicit patent grant.	`Apache-2.0`
Improvements must stay open.	`GPL-3.0-or-later`
…even when offered as a web service.	`AGPL-3.0-or-later`
It is data, text or images — not code.	`CC BY 4.0 · CC0`

ADDING ONE ON GITHUB

1. **Add file → Create new file**
2. Name it `LICENSE`
3. **Choose a license template**
4. Commit

FOR YOUR PROJECT

Code under **MIT**, your own data and texts under **CC BY 4.0** — and respect the licences of your sources.

Licence names as SPDX identifiers (spdx.org/licenses) — the standard way to name a licence in metadata.

Where open source lives



GitHub

github.com

The largest platform. Owned by Microsoft since 2018 — the platform itself is proprietary.



GitLab

gitlab.com

Open core: the Community Edition is open source and can be self-hosted.



Codeberg

codeberg.org

A non-profit from Berlin, running the open source forge Forgejo.



Bitbucket

bitbucket.org

Atlassian's platform, popular in companies that use Jira.

AND FOR THE LONG RUN

Software Heritage (softwareheritage.org) archives all publicly available source code — some 30 billion files from 445 million projects — so it survives when a platform closes.

What is GitHub?

A web platform for version control and collaboration, built on **Git**. It lets you:

Track changes

every version of your code — version control

Collaborate

with your team, from anywhere

Host repositories

store, organise and share code

Review changes

pull requests before merging

Manage work

issues for bugs, ideas and questions

Join the community

contribute to open source projects

IN THIS COURSE

Today's lab
your profile README, written in Markdown

The course repository
slides, labs and the schedule

Your project
one repository per team

Your questions
as issues — the answers help everybody

Git ≠ GitHub: Git is the tool on your machine (lecture 3), GitHub a service that hosts Git repositories.

Anatomy of a data project repository

`your-team / **your-project**`

`data/raw/`	original downloads — never edit them
`data/processed/`	cleaned data, created by your code
`notebooks/`	exploration and analysis
`src/`	reusable code: scraping, cleaning
`.gitignore`	what Git should not track
`LICENSE`	what others may do with your work
`README.md`	the front door of your project
`requirements.txt`	the packages and versions you use

A GOOD README ANSWERS

- What is this project about — and who is on the team?
- Which questions do you answer?
- Which data do you use — from where, under which licence?
- How do I reproduce your results?
- What did you find?
- Where did you use AI?

What is Markdown?

> **A lightweight markup language: plain text that is easy to read — and easy to turn into HTML.**

2004	created by John Gruber, with Aaron Swartz
Plain text	readable even without rendering
Everywhere	README files, issues, notebooks, docs, chats

YOU WILL WRITE MARKDOWN FOR

`README.md` `AI_USAGE.md` `GitHub issues` `pull requests` `Jupyter notebooks` `AGENTS.md`

WHY MARKDOWN — TODAY

“Mainstream LLMs, such as OpenAI’s GPT-4o, natively ‘speak’ Markdown, and often incorporate Markdown into their responses unprompted. [...] Markdown conventions are also highly token-efficient.”

README of microsoft/markitdown

Markdown in one slide

YOU WRITE

README.md

```
# Cycling in Cologne

**Question:** Do people cycle
more on *warm* days?

## Data

- Bike counts: [Offene Daten Köln](https://offenedaten-koeln.de)
- Weather: Deutscher Wetterdienst

Run `python src/clean.py` first.
```

YOU GET

Cycling in Cologne

Question: Do people cycle more on *warm* days?

Data

- Bike counts: Offene Daten Köln
- Weather: Deutscher Wetterdienst

Run `python src/clean.py` first.

`# Heading` `## Subheading` `**bold**` `*italic*` `[text](url)` `- list item` `` `code` `` `1. numbered` `| table |`

Reference: commonmark.org/help · GitHub Flavored Markdown adds tables, task lists and more: github.github.com/gfm

What is open data?

> **Open data and content can be freely used, modified, and shared by anyone for any purpose.**
>
> The Open Definition, Open Knowledge Foundation — opendefinition.org

1

Availability and access

Available as a whole, downloadable online, in a convenient and modifiable form.

2

Reuse and redistribution

Under terms that permit reuse, redistribution and combination with other data.

3

Universal participation

Everyone may use it — no restrictions on persons, groups or fields of use.

! **Public is not open.** A table in a PDF on a website is public — open data needs an open licence and a machine-readable format.

The 5-star open data scheme

Tim Berners-Lee's staircase for publishing data: every star makes the data easier to use — and to integrate.

Aim for **three stars** in your project: an open licence, structured, in an open format.

★☆☆☆☆

On the web, openly licensed

any format, e.g. a PDF

Look at it, print it, keep it.

★★☆☆☆

Structured data

e.g. Excel instead of a scanned table

Process it with software.

★★★☆☆

Open format

e.g. CSV instead of Excel

Use it without proprietary software.

★★★★☆

URIs for things

RDF: every thing gets an identifier

Others can link to your data.

★★★★★

Linked to other data

Linked Data

Discover more data — context for free.

Tim Berners-Lee (2006/2010), Linked Data design note, w3.org/DesignIssues/LinkedData · explained at 5stardata.info/en

Open data is everywhere

Government

Census, budgets, crime statistics

Destatis GENESIS-Online

Environment

Weather, climate, air quality

Deutscher Wetterdienst

Geospatial

Maps, addresses, satellite images

OpenStreetMap · Copernicus

Mobility

Timetables, traffic counts, sharing

Mobilithek

Health

Disease surveillance, health indicators

WHO Global Health Observatory

Research

Datasets behind publications

Zenodo · OpenAlex

Economy

Trade, prices, company registers

Eurostat · OpenCorporates

Knowledge

Facts about everything, linked

Wikidata

Examples of publishers — each has its own licence and terms. Check them before you use the data.

Open by default: open data and the law

EUROPEAN UNION

Open Data Directive (EU) 2019/1024

Public sector information should be open by default — reusable for any purpose, in machine-readable formats.

High-value datasets · since 9 June 2024

Six categories must be free of charge, machine-readable, via APIs and as bulk download — under CC BY 4.0 or CC0:

GERMANY

§ 12a E-Government Act (2017)

Federal authorities publish their unprocessed raw data as open data — free of charge, on GovData.

Data Use Act (DNG, 2021)

Implements the EU directive: rules for reusing public sector and publicly funded research data.

Geospatial · Earth observation and environment · Meteorological · Statistics · Companies and company ownership · Mobility

For you: public data in the EU is increasingly open, free and available via APIs.

Open source intelligence (OSINT)

The collection and analysis of data gathered from open sources — overt sources and publicly available information — to produce intelligence.

Careful: here, **open source** means open *sources of information*, not open source software.

SOURCES

Media	newspapers, radio, television
Internet	publications, forums, blogs, social media
Government	reports, budgets, public records
Commercial	industry reports, databases
Academia	journals, conference proceedings
Grey literature	preprints, patents, reports

METHODS

- Search engines and advanced operators (“Google dorking”)
- Web scraping and APIs
- Social media intelligence (SOCMINT)
- Geospatial intelligence (GEOINT): maps, satellite images
- Public records and data brokers

Definition after en.wikipedia.org/wiki/Open-source_intelligence · example: Bellingcat’s investigations, bellingcat.com

Open, but not for everything

As open as possible, as closed as necessary.

Principle of the EU research programmes Horizon 2020 and Horizon Europe

Personal data

Names, addresses, health records are protected by the GDPR — even when they are public somewhere.

Re-identification

“Anonymous” data can be de-anonymised: AOL search logs (2006) and the Netflix Prize ratings (2008) were traced back to people.

Other reasons

Security, trade secrets, locations of endangered species: some data should stay closed.

For your project: aggregate, don’t collect people — and never publish personal data you scraped.

Creative Commons licences

BY

Attribution

Credit the creator.

SA

Share alike

Share adaptations under the same licence.

NC

Non-commercial

No commercial use.

ND

No derivatives

Share only unchanged copies.

FROM MOST OPEN TO LEAST OPEN

CC0
No rights reserved
✓ open data

CC BY
Attribution
✓ open data

CC BY-SA
… and share alike
✓ open data

CC BY-NC
… non-commercial only
✗ not open

CC BY-NC-SA
… non-commercial, share alike
✗ not open

CC BY-ND
… no derivatives
✗ not open

CC BY-NC-ND
… non-commercial, no derivatives
✗ not open

> Use version 4.0 (2013) — it also covers the EU database right. NC and ND fail the Open Definition.

Licences made for data

OPEN DATA COMMONS · opendatacommons.org

Designed for databases

PDDL — Public domain — no conditions, like CC0.

ODC-By — Attribution — like CC BY.

ODbL — Attribution and share-alike for derived databases. Used by OpenStreetMap.

DATENLIZENZ DEUTSCHLAND 2.0

Used by German authorities

dl-de/zero-2-0

Zero — use without any conditions.

dl-de/by-2-0

Namensnennung — attribution: name the provider and link the licence.

Common on GovData and Offene Daten Köln — though GovData now recommends CC BY 4.0 or CC0.

Why special licences? In the EU, a database can be protected by a *database right* — even when its individual facts are not. Data licences cover that right.

Code, data, text: the right licence for each

What	Recommended	Why
Code	MIT · Apache-2.0 GPL-3.0	Deal with source code, binaries, patents
Data	CC0 · CC BY 4.0 ODbL	Cover database rights
Text, images, slides	CC BY 4.0 CC BY-SA 4.0	Made for creative works

> **Software licences like MIT or the GPL can be used for data — but they were not made for it.**

A REAL EXAMPLE: OUR WORLD IN DATA

"All visualizations, data, and code produced by *Our World in Data* are completely open access under the Creative Commons BY license. […] The data produced by third parties […] is subject to the license terms from the original third-party authors."

Two lessons: one licence for code *and* data works when it is stated clearly — and a dataset built from other datasets inherits their licences.

Quote: README of github.com/owid/co2-data — the dataset from the first slide

Give credit: title, author, source, licence

T Title
Data on CO_2 and Greenhouse Gas Emissions

A Author
Our World in Data

S Source
github.com/owid/co2-data

L Licence
CC BY 4.0

IN YOUR README OR ON A CHART

"Data on CO_2 and Greenhouse Gas Emissions" by Our World in Data, github.com/owid/co2-data, CC BY 4.0

COMBINING DATASETS

Credit every source

List each dataset with its licence in your README.

Share-alike travels

A dataset derived from BY-SA or ODbL data must use the same — or a compatible — licence.

NC and ND restrict you

No commercial use, no modified copies — and they limit what others may do with your results.

When in doubt, ask

Publishers often grant permission — or know a better source.

TASL: the attribution practice recommended by Creative Commons — wiki.creativecommons.org/wiki/Recommended_practices_for_attribution

Put the licence where machines can find it

terminal

```
$ curl -s https://api.github.com/repos/pandas-dev/pandas \
    | jq .license.spdx_id
"BSD-3-Clause"
$ curl -s https://api.github.com/repos/owid/co2-data \
    | jq .license.spdx_id
null
```

GitHub's API reports the licence it detects in a repository — as an SPDX identifier.

✓ **pandas-dev/pandas**

A `LICENSE` file in the repository: GitHub recognises it and reports **BSD-3-Clause**.

✗ **owid/co2-data**

The licence is stated in the README text only. People understand it — machines don't.

For your repository: a `LICENSE` file for your code, and the data licence in machine-readable metadata — FAIR principle R1.1.

The FAIR principles

F

Findable

Humans and machines can find the data and its metadata.

A

Accessible

Once found, it is clear how to get it — with authentication if needed.

I

Interoperable

It can be combined with other data and used by applications.

R

Reusable

It is described well enough to be reused — in other settings, by others.

FAIR ≠ open

FAIR health data can sit behind a login — access is regulated, but clear.

Open data can be un-FAIR: a CSV without metadata on a random web page is open, but hard to find and reuse.

Wilkinson et al. (2016), The FAIR Guiding Principles for scientific data management and stewardship, Scientific Data 3, 160018 · go-fair.org

To be findable and accessible

Findable

F1 (Meta)data are assigned a globally unique and persistent identifier

F2 Data are described with rich metadata (defined by R1 below)

F3 Metadata clearly and explicitly include the identifier of the data it describes

F4 (Meta)data are registered or indexed in a searchable resource

IN PRACTICE

A DOI from a repository like Zenodo · a README with metadata · listed in a portal

Accessible

A1 (Meta)data are retrievable by their identifier using a standardized communications protocol

A1.1 The protocol is open, free, and universally implementable

A1.2 The protocol allows for an authentication and authorization procedure, where necessary

A2 Metadata are accessible, even when the data are no longer available

IN PRACTICE

Download via HTTPS or an API · a landing page that stays, even if the file is gone

To be interoperable and reusable

Interoperable

I1	(Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation
I2	(Meta)data use vocabularies that follow FAIR principles
I3	(Meta)data include qualified references to other (meta)data

IN PRACTICE

Open formats (CSV, JSON) · standard codes: ISO 8601 dates, ISO 3166 country codes · links to the sources you used

Reusable

R1	(Meta)data are richly described with a plurality of accurate and relevant attributes
R1.1	(Meta)data are released with a clear and accessible data usage license
R1.2	(Meta)data are associated with detailed provenance
R1.3	(Meta)data meet domain-relevant community standards

IN PRACTICE

A data dictionary for every column · a licence · where each value comes from

Make your project dataset FAIR

1 Give it an identifier

F1 · A1

Publish it with a DOI — e.g. on Zenodo, which can archive GitHub releases automatically.

2 Describe it

F2 · R1

A README with title, authors, sources, dates — and a data dictionary for every column.

3 Use open formats

I1 · R1.3

CSV or JSON in UTF-8, ISO 8601 dates, ISO 3166 country codes.

4 License it

R1.1

CC BY 4.0 or CC0 for your data — compatible with your sources.

5 Document provenance

R1.2

Where each source comes from, when you collected it, which script built it.

6 Link your sources

I3

Cite every dataset you used, with its identifier or URL.

Due with milestone 2 on 18.12.26: your dataset, hosted online.

Where to find open data?

COLOGNE AND NRW

offenedaten-koeln.de
the city of Cologne's portal

open.nrw
the portal of North Rhine-Westphalia

GERMANY

govdata.de
the national data catalogue

genesis.destatis.de
official statistics

opendata.dwd.de
weather and climate

bund.dev
federal APIs, documented by volunteers

EUROPE AND THE WORLD

data.europa.eu
the EU's data catalogue

ec.europa.eu/eurostat
EU statistics

data.worldbank.org
development indicators

ourworldindata.org
global topics, CC BY

data.gov
the US federal portal

data.gov.uk
the UK's portal

RESEARCH AND SEARCH

zenodo.org
research data with DOIs

datasetsearch.research.google.com
a search engine for datasets

kaggle.com/datasets
community uploads — check the source

awesome-public-datasets
a curated list on GitHub

KNOWLEDGE AND MAPS

wikidata.org
linked facts, CC0

openstreetmap.org
the free map, ODbL

Portals have APIs, too

terminal

```
$ API=https://www.govdata.de/ckan/api/3/action
$ curl -s "$API/package_search?q=fahrrad&rows=1" > hits.json
$ jq .result.count hits.json
413
$ jq -r '.result.results[0].title' hits.json
Fahrrad-Dauerzählstellen Konstanz
$ jq -r '.result.results[0].resources[0].license' hits.json
http://dcat-ap.de/def/licenses/cc-by/4.0
```

A search request, answered as JSON — the FAIR principles at work: a standard protocol (A1), a licence in the metadata (R1.1).

413 datasets

match the search term *fahrrad* on GovData

Metadata for each

title, publisher — here the Mobilithek — formats, and a licence URI: CC BY 4.0

One API, many portals

GovData and Open.NRW both run CKAN: the same request works on both.

Next week you run commands like these yourself — and in lecture 5 we call APIs from Python.

Is this dataset any good?

01
Licence
Is it open? What do you have to credit?

02
Publisher
Who made it — and can you trust them?

03
Documentation
Metadata, a data dictionary, a method?

04
Format
How many stars? Can you read it with Python?

05
Coverage
Which years, places, granularity?

06
Freshness
When was it last updated — and how often?

07
Quality
Missing values, duplicates, obvious errors?

08
People
Does it contain personal data?

Write down the answers — they are the first section of your roadmap.

Project ideas: enrich an open dataset

Global development

BASE

World Bank Open Data

ENRICH WITH

Wikipedia lists, Freedom House scores

ANALYSE

Internet access vs. freedom scores

Cities and mobility

BASE

OpenStreetMap

ENRICH WITH

Wikipedia city pages, GTFS timetables

ANALYSE

Population density vs. road length per person

Companies and trade

BASE

OpenCorporates, WTO statistics

ENRICH WITH

Wikipedia company pages, SEC EDGAR filings

ANALYSE

Trade volume vs. number of subsidiaries

Health and science

BASE

ClinicalTrials.gov

ENRICH WITH

PubMed counts, Wikipedia page views

ANALYSE

Clinical trials vs. public attention

Conflict and media

BASE

GDELT event data

ENRICH WITH

Wikipedia page edits and views

ANALYSE

News coverage vs. public attention

Check the terms of every source: ACLED, last year's example for conflict data, is not open — registration, non-commercial use only, no redistribution.

The recipe: base + enrichment + analysis

Base dataset

official statistics, structured

World Bank · Destatis · OSM

\+

Enrichment

scraped or crowd-sourced signals

Wikipedia · news · prices

→

Analysis

find relationships

correlation · regression · trends

THE KEY STEP: JOIN ON A SHARED KEY — COUNTRY CODE, CITY, DATE

`base.csv`: `iso_code` `year` `gdp`

\+

`enrichment.csv`: `iso_code` `year` `wiki_edits`

→

`combined`: `iso_code` `year` `gdp` `wiki_edits`

```
base.merge(enrichment, on=["iso_code", "year"])
```

One line of pandas. The hard part is making the keys match — that is data cleaning.

> **The goal: relationships between official statistics and observed, crowd-sourced signals.**

Your next steps towards milestone 1

1

Form a team

Find your teammates — today's lab is a good place.

2

Create a repository

One per team, everyone a collaborator — and invite **breuert**.

3

Find a topic and data

A question you care about, and an open dataset to start from.

4

Ask questions

Two or three research questions your data can answer.

5

Write the roadmap

In `README.md`: topic, data and licences, questions, plan.

MILESTONE 1 · ROADMAP

Due 13.11.26

Everything you need is in today's lecture: an open dataset with a clear licence, a GitHub repository and Markdown.

Key takeaways

01 Open source software comes with the freedom to use, study, change and share it. The licence grants it — no licence, no permission.

02 **Permissive** licences like MIT ask for credit; **copyleft** licences like the GPL keep derived works open.

03 **Open data** can be freely used, modified and shared by anyone for any purpose. Check the licence and give credit: title, author, source, licence.

04 Make data **FAIR**: persistent identifiers, rich metadata, open formats, a clear licence and provenance.

Open tools and open data make your results transparent, reproducible and reusable — the two pillars of this course.

Lab: your GitHub profile in Markdown

YOUR TASK

1. Sign up at **github.com** — if you don't have an account yet.
2. Create a public repository named exactly like your **username**, with a `README.md`: your profile README.
3. Introduce yourself in Markdown — use headings, a list, a link, etc.
4. Then follow the instructions in the first lab assignment on GitHub: `https://github.com/breuert/data-integration`

Rather not show your full name in public? That is okay - simply include your name together with your GitHub handle and your student ID in the group e-mail.

Markdown to try

```
# Hi, I'm …

## What I study

- a list of interests

**bold** and *italic*

[my team's repo](https://…)

![a picture](me.png)
```

HELP

GitHub Docs: "Managing your profile README" · commonmark.org/help

Today’s learning outcomes

- ✓ **Explain how this course works**
  Lectures, labs, the group project, milestones and grading.
- ✓ **Explain open source software**
  The four freedoms, the Open Source Definition, why it matters.
- ✓ **Tell software licences apart**
  Permissive vs. copyleft — and why no licence means no permission.
- ✓ **Use GitHub and Markdown**
  Repositories, README files, your profile.
- ✓ **Explain and find open data**
  The Open Definition, five stars, the law, portals, quality checks.
- ✓ **License, credit and FAIRify data**
  CC, ODC and DL-DE licences, TASL, the FAIR principles.

Best practice: open tools, open data, open licences — and document everything.

Check your understanding

Show answers →

01

A public GitHub repository has no `LICENSE` file. May you copy its code into your project?

02

You change a GPL-licensed library and publish your program. What do you have to do?

03

Is a dataset under CC BY-NC 4.0 open data?

04

Your data is `data_final_v2.xlsx` on your laptop. Name three steps to make it FAIR.

Questions?

Next lecture: Command Line Interface on 30.10.26.
Milestone 1 — your roadmap — is due on 13.11.26.

RESOURCES FROM TODAY

- opensource.org/osd — the Open Source Definition
- gnu.org/philosophy/free-sw.html — the four freedoms
- choosealicense.com — pick a software licence
- creativecommons.org/cc-licenses — the CC licences
- opendefinition.org — what open means
- 5stardata.info/en — the five stars, explained
- go-fair.org/fair-principles — FAIR
- govdata.de · data.europa.eu — open data portals
- commonmark.org/help — Markdown in ten minutes