Just tyPyt: a pain-free recipe for reproducible reports and publications

Mathieu Daëron (*daeron@lsce.ipsl.fr*)
Laboratoire des Sciences du Climat et de l'Environnement

Reproducible, transparent research: pros and cons

Good:

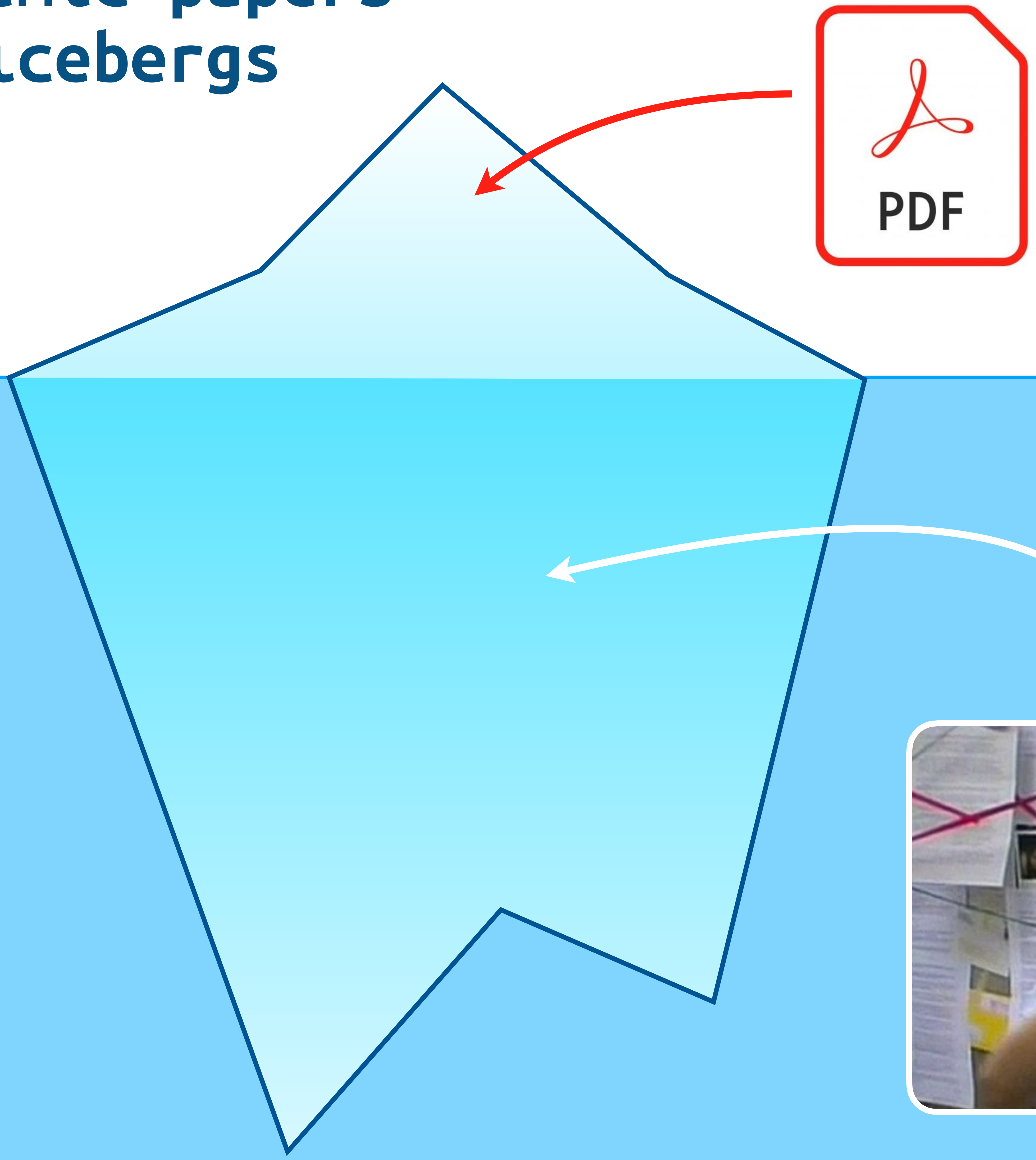
- You avoid mistakes
- You can reuse old work
- You get better reviews
- Everyone (not just your pals) can fork your ideas

Bad:

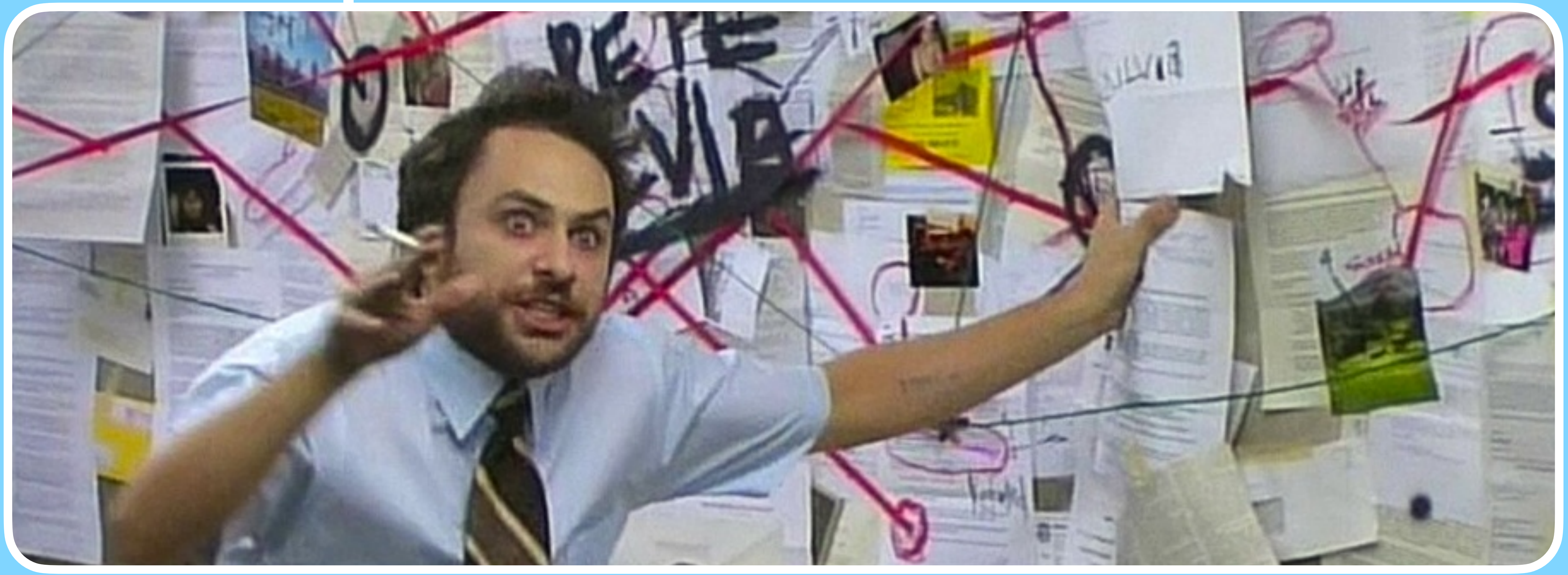
- It's more work for you
- ~~Everyone (not just your pals) can fork your ideas~~

Academic papers are icebergs

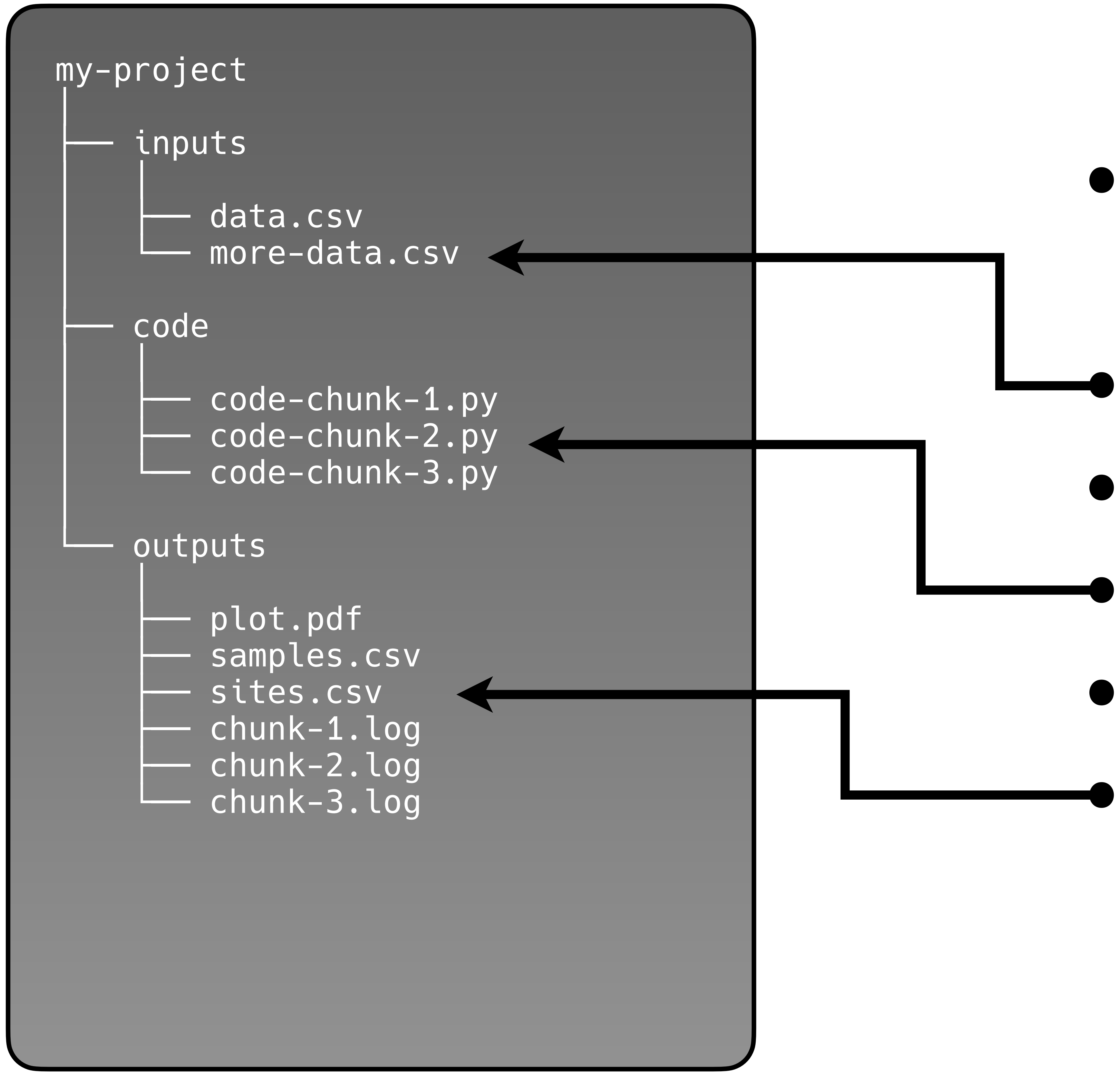
Academic papers are icebergs



1 document
7 pages
1.4 MB



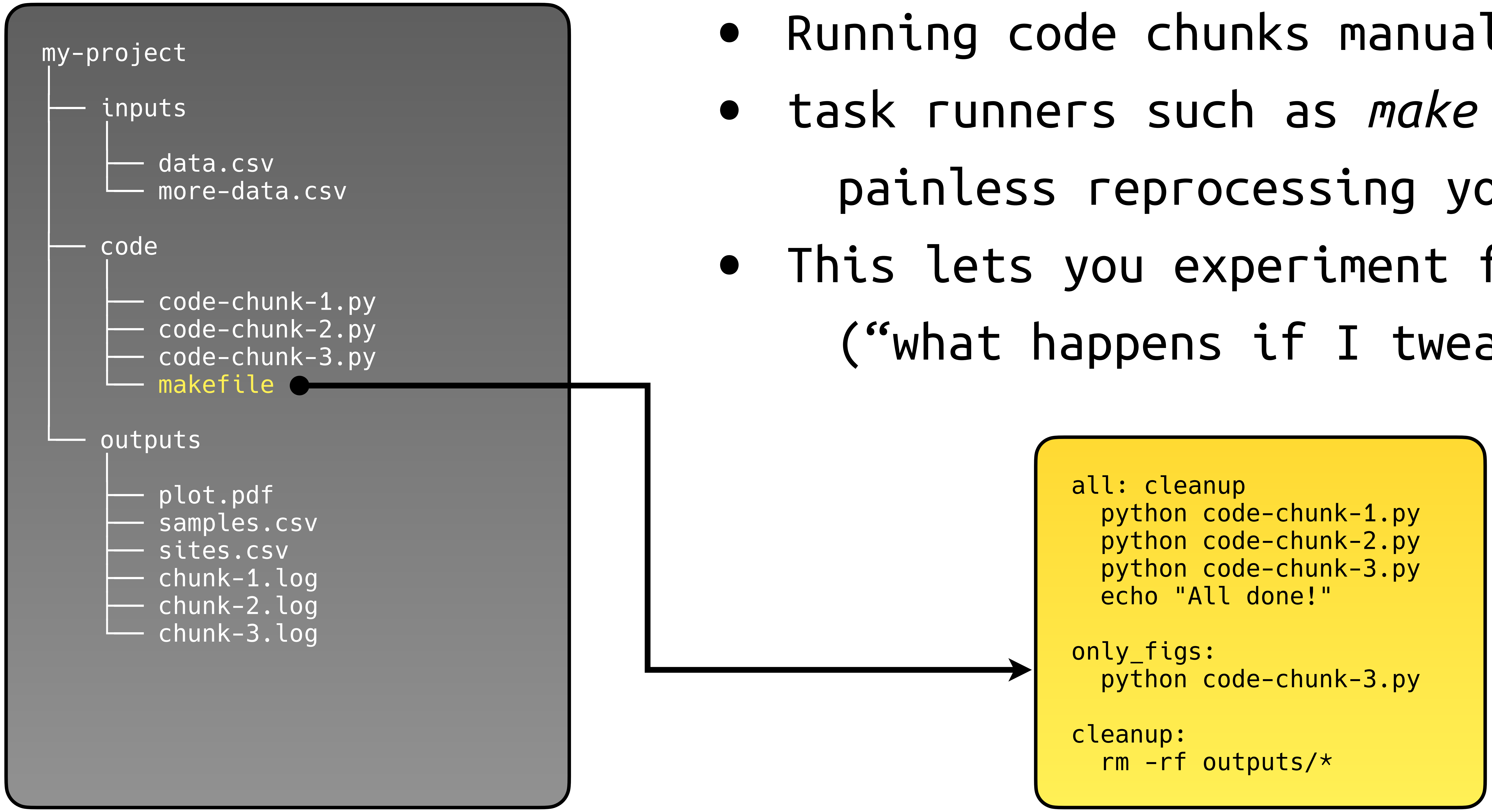
From raw data to something meaningful:
Use a high-level coding language (Python, R, Julia...)



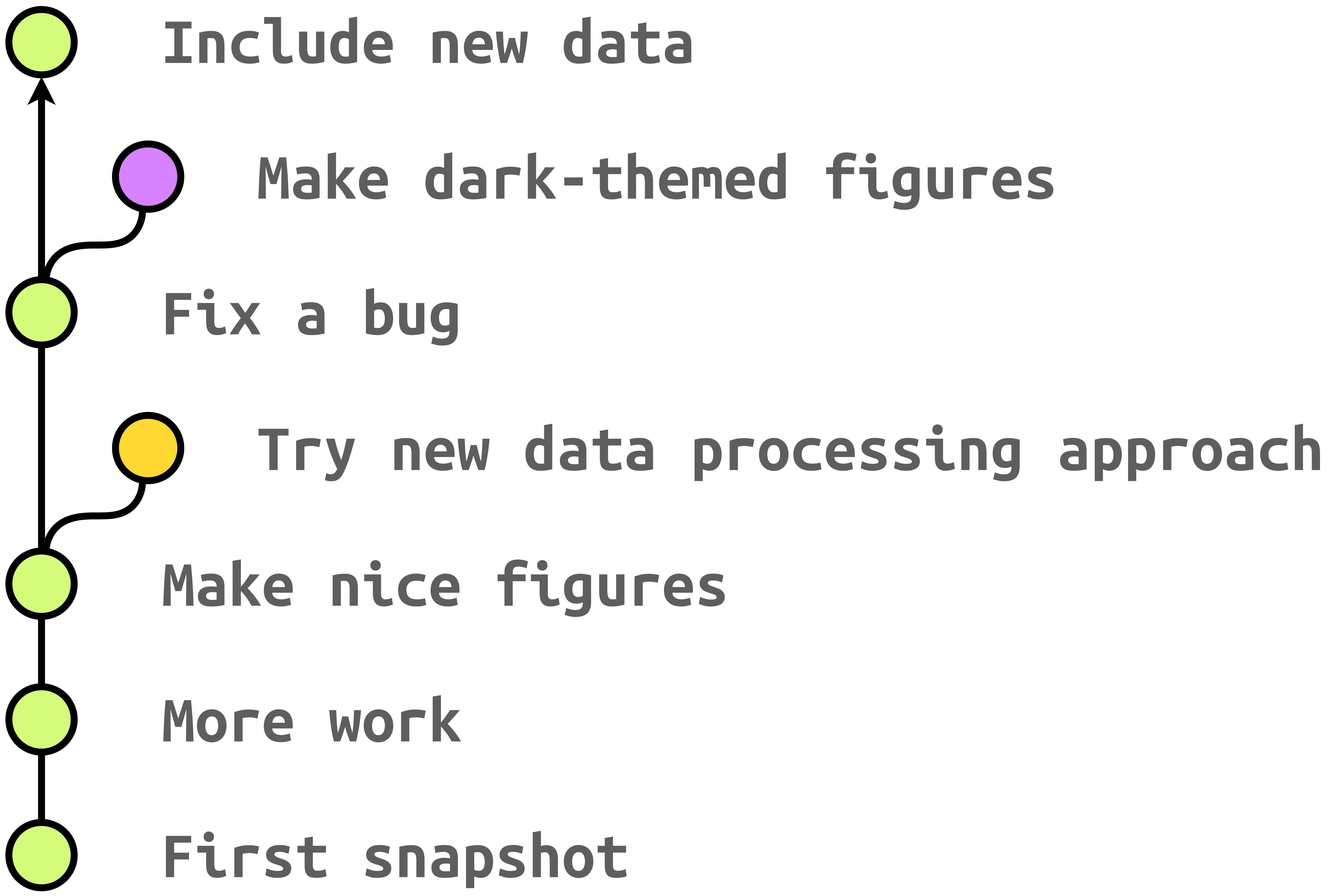
- High-level means easily understandable (for others and for later you)
- Leave raw inputs absolutely unchanged
- Everything after that should be in code
- Separate code into smaller chunks
- Comments are your friend
- Generate lots of outputs (tables, plots, logs)

Automating the whole pipeline (*make*, *just*, *poe*...)

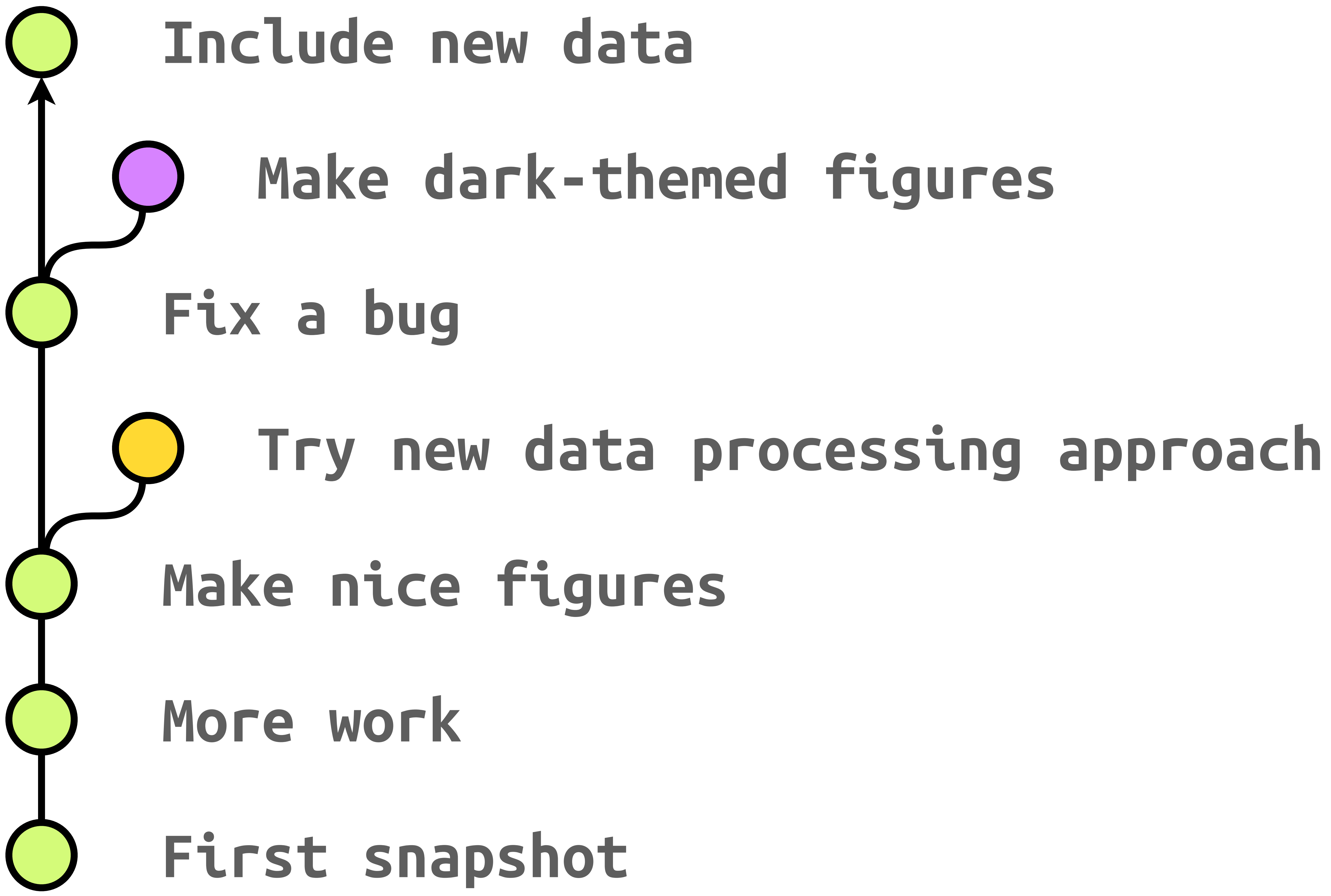
- Running code chunks manually is no fun
- task runners such as *make* allow painless reprocessing your full project
- This lets you experiment freely (“what happens if I tweak this parameter?”)



Enter the multiverse:
Embrace version control (Git, Jujutsu...)



Enter the multiverse:
Embrace version control (Git, Jujutsu...)



From: Jens Fiebig
Subject: Re: OGLS preprint
Date: 11 October 2023 at 22:22
To: Mathieu Daëron

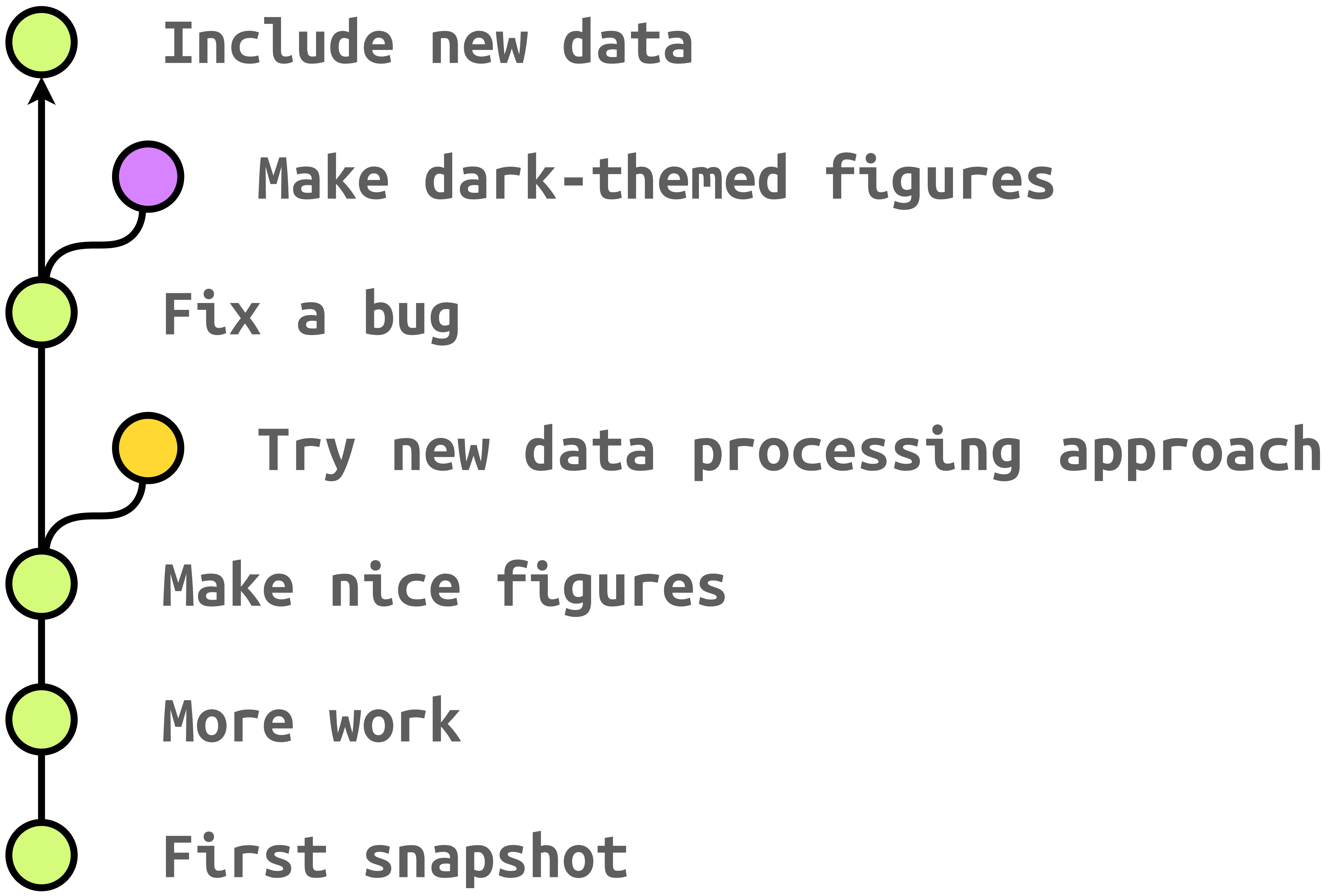
Hi Mathieu, [...]

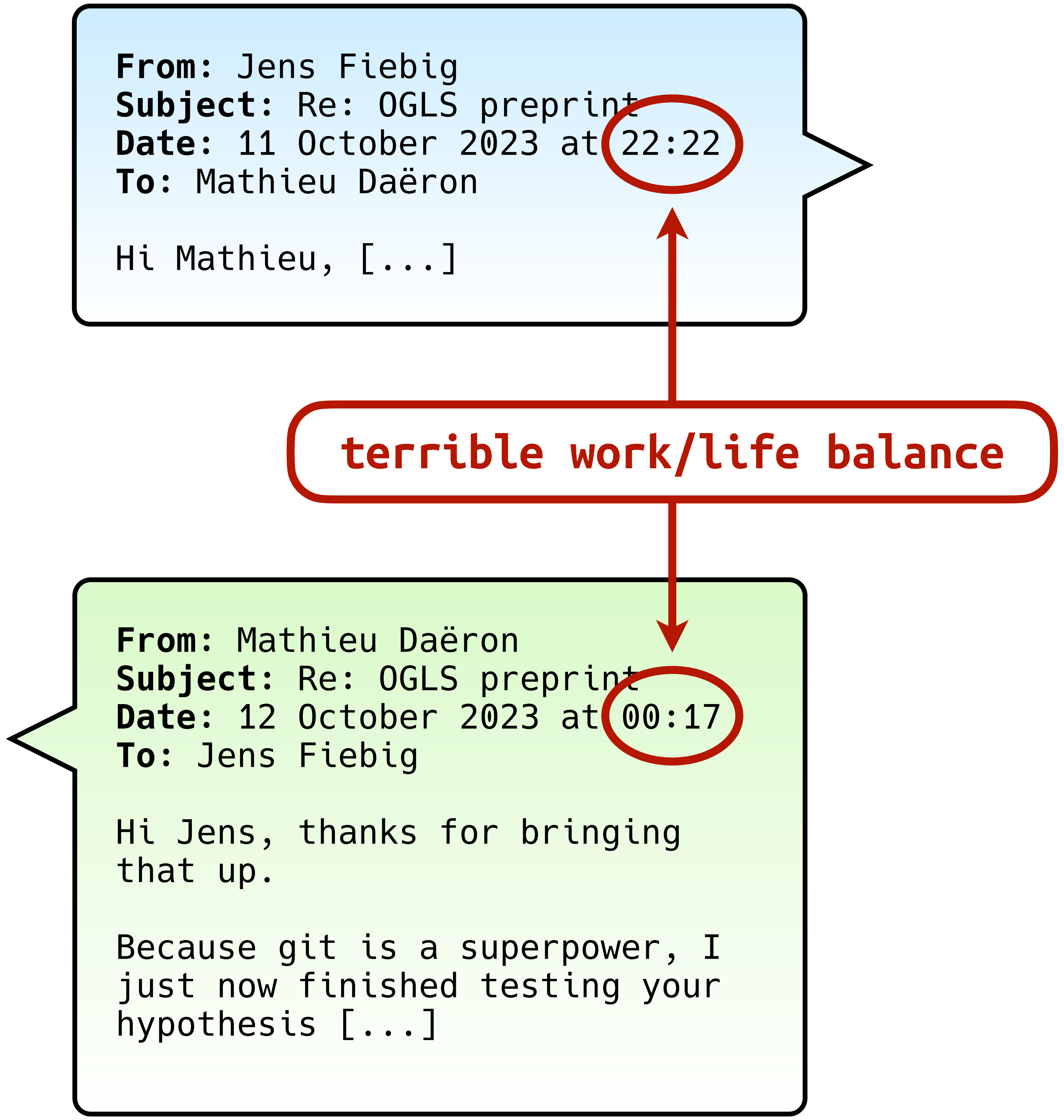
From: Mathieu Daëron
Subject: Re: OGLS preprint
Date: 12 October 2023 at 00:17
To: Jens Fiebig

Hi Jens, thanks for bringing that up.

Because git is a superpower, I just now finished testing your hypothesis [...]

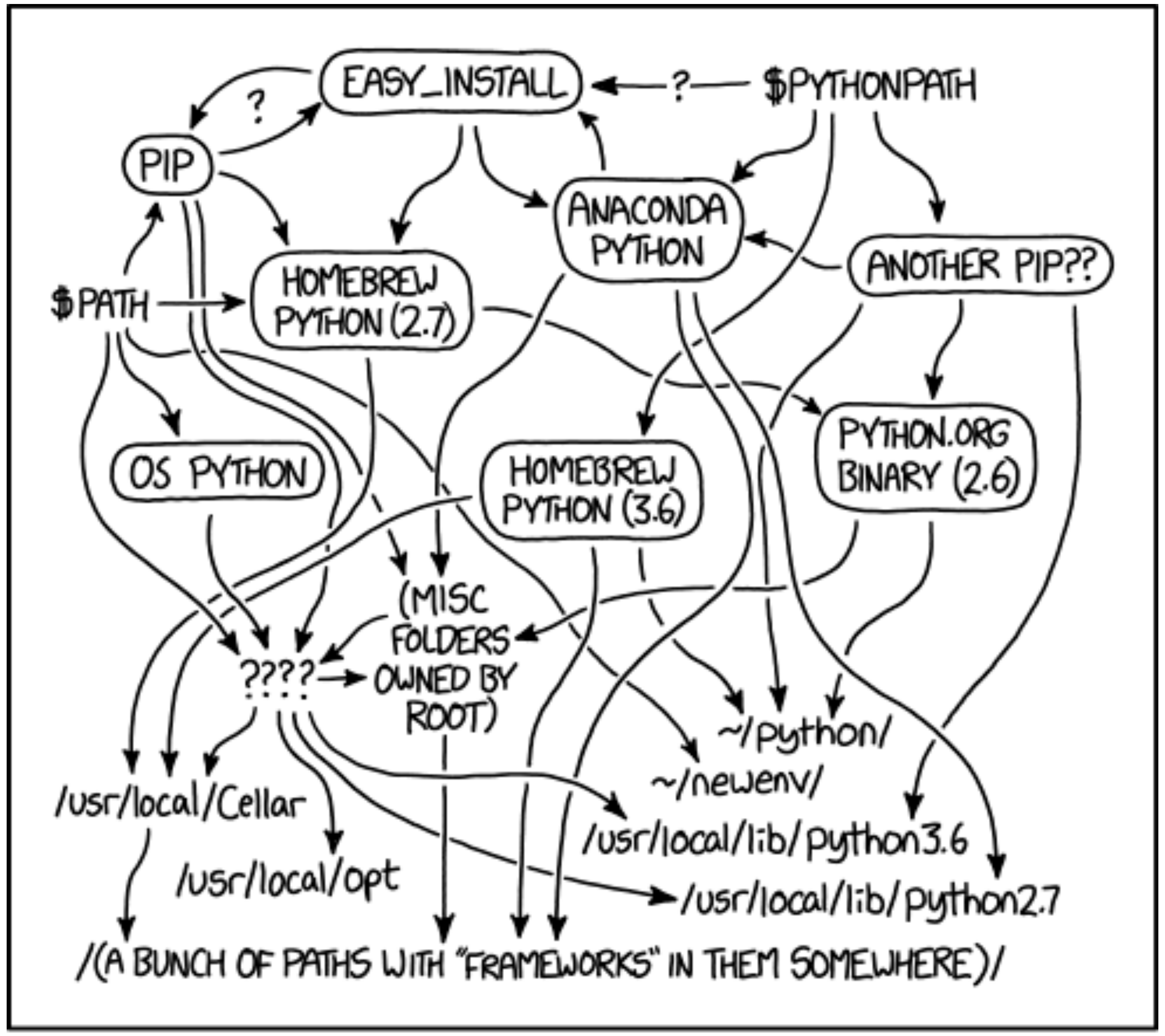
Enter the multiverse:
Embrace version control (Git, Jujutsu...)





Escape dependency hell (*uv*, *pixi*...)

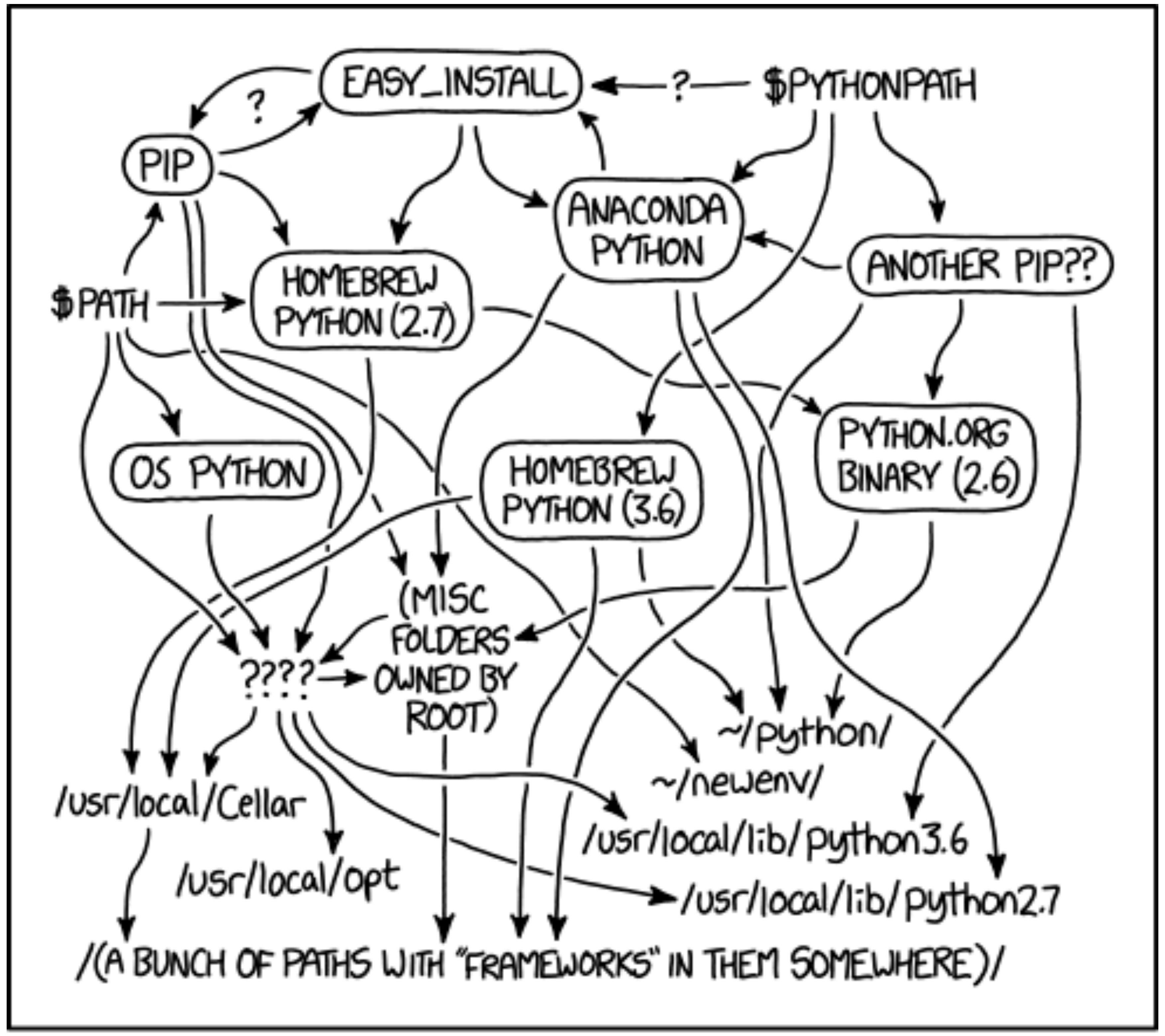
- Each ***user*** may have code packages with different, conflicting versions.
- Each ***project*** may require different, incompatible package versions.
- Python used to be particularly bad.



https://xkcd.com/1987

Escape dependency hell (*uv*, *pixi*...)

- Each ***user*** may have code packages with different, conflicting versions.
- Each ***project*** may require different, incompatible package versions.
- Python used to be particularly bad.



https://xkcd.com/1987

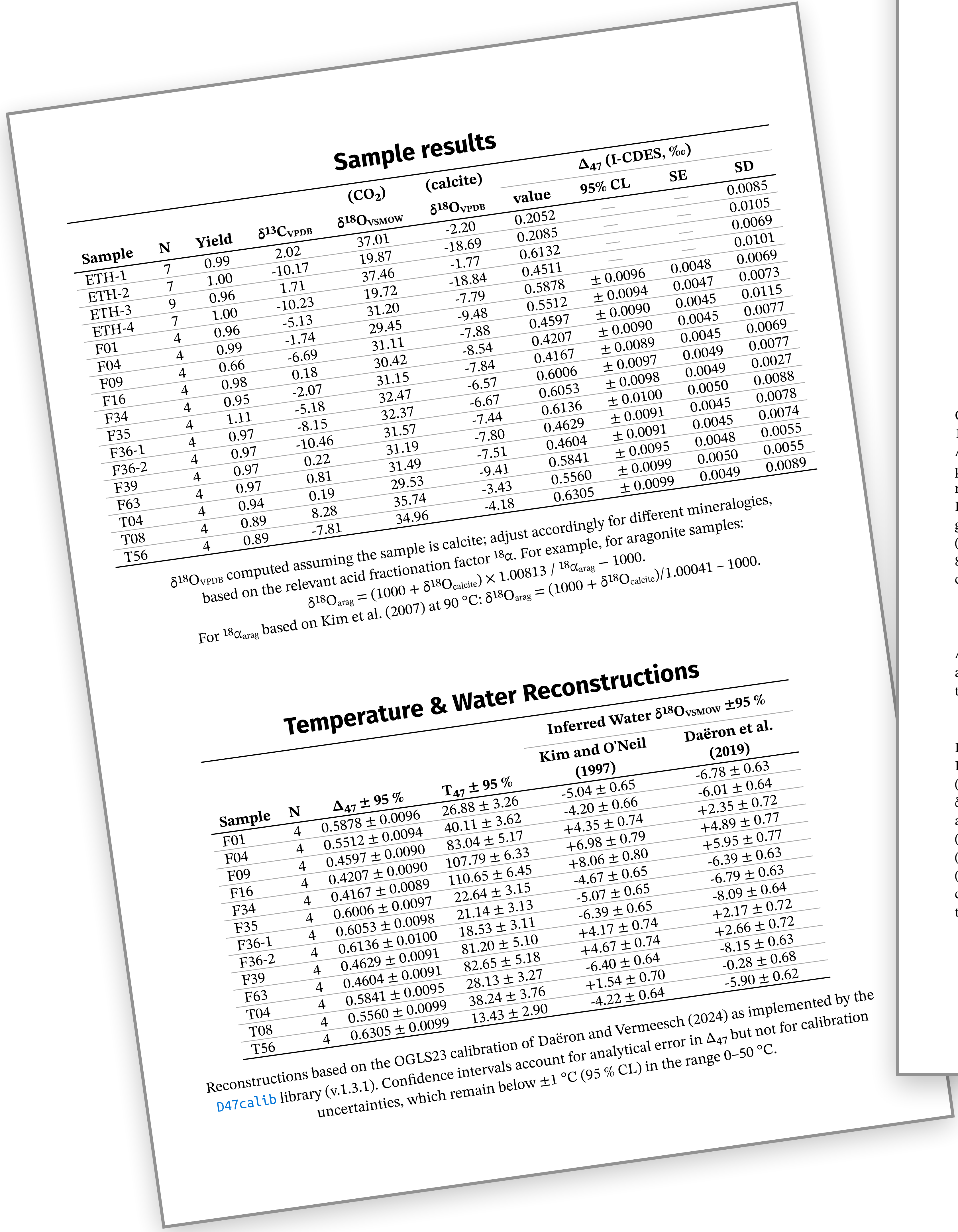
- Solution: modern package managers such as **uv** (*https://docs.astral.sh/uv*) create a complete virtual Python installation within each project.
- These tools are designed for speed and cross-platform reproducibility.
- Largest change to my coding quality-of-life in years

From data to written ideas (Typst)

```
my-project
├── inputs
├── code
├── outputs
├── tech-report
│   ├── data.csv
│   ├── metadata.toml
│   └── report.typ
└── manuscript
    ├── my-paper.typ
    └── my-refs.bib
```

- Hot take: Typst is considerably less insane that either Word or LaTeX
- Bonus perk: painless dynamic loading of previously output data (*.csv, *.toml...)
- Usual bells and whistles included (citations, figure/table refs, links...)

Automatic yet good-looking reports



Sample results

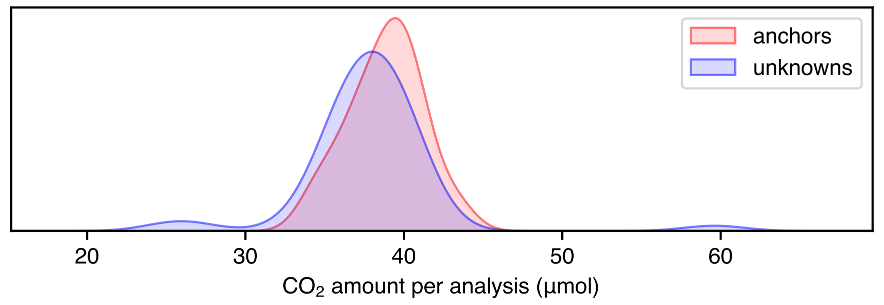
Sample	N	Yield	$\delta^{13}C_{VPDB}$	$\delta^{18}O_{VSMOW}$ (CO₂)	$\delta^{18}O_{VPDB}$ (calcite)	Δ_{47} (I-CDES, ‰) value	95% CL	SE	SD
ETH-1	7	0.99	2.02	37.01	-2.20	0.2052	—	—	0.0085
ETH-2	7	1.00	-10.17	19.87	-18.69	0.2085	—	—	0.0105
ETH-3	9	0.96	1.71	37.46	-1.77	0.6132	—	—	0.0069
ETH-4	7	1.00	-10.23	19.72	-18.84	0.4511	—	—	0.0101
F01	4	0.96	-5.13	31.20	-7.79	0.5878	± 0.0096	0.0048	0.0069
F04	4	0.99	-1.74	29.45	-9.48	0.5512	± 0.0094	0.0047	0.0073
F09	4	0.66	-6.69	31.11	-7.88	0.4597	± 0.0090	0.0045	0.0115
F16	4	0.98	0.18	30.42	-8.54	0.4207	± 0.0090	0.0045	0.0077
F34	4	0.95	-2.07	31.15	-7.84	0.4167	± 0.0089	0.0045	0.0069
F35	4	1.11	-5.18	32.47	-6.57	0.6006	± 0.0097	0.0049	0.0077
F36-1	4	0.97	-8.15	32.37	-6.67	0.6053	± 0.0098	0.0049	0.0027
F36-2	4	0.97	-10.46	31.57	-7.44	0.6136	± 0.0100	0.0050	0.0088
F39	4	0.97	0.22	31.19	-7.80	0.4629	± 0.0091	0.0045	0.0078
F63	4	0.97	0.81	31.49	-7.51	0.4604	± 0.0091	0.0045	0.0074
T04	4	0.94	0.19	29.53	-9.41	0.5841	± 0.0095	0.0048	0.0055
T08	4	0.89	8.28	35.74	-3.43	0.5560	± 0.0099	0.0050	0.0055
T56	4	0.89	-7.81	34.96	-4.18	0.6305	± 0.0099	0.0049	0.0089

$\delta^{18}O_{VPDB}$ computed assuming the sample is calcite; adjust accordingly for different mineralogies, based on the relevant acid fractionation factor $^{18}\alpha$. For example, for aragonite samples: $\delta^{18}O_{arag} = (1000 + \delta^{18}O_{calcite}) \times 1.00813\ /\ ^{18}\alpha_{arag} - 1000$.
For $^{18}\alpha_{arag}$ based on Kim et al. (2007) at 90 °C: $\delta^{18}O_{arag} = (1000 + \delta^{18}O_{calcite})/1.00041 - 1000$.

Temperature & Water Reconstructions

Sample	N	Δ_{47} ± 95 %	T_{47} ± 95 %	Inferred Water $\delta^{18}O_{VSMOW}$ ±95 %: Kim and O'Neil (1997)	Daëron et al. (2019)
F01	4	0.5878 ± 0.0096	26.88 ± 3.26	-5.04 ± 0.65	-6.78 ± 0.63
F04	4	0.5512 ± 0.0094	40.11 ± 3.62	-4.20 ± 0.66	-6.01 ± 0.64
F09	4	0.4597 ± 0.0090	83.04 ± 5.17	+4.35 ± 0.74	+2.35 ± 0.72
F16	4	0.4207 ± 0.0090	107.79 ± 6.33	+6.98 ± 0.79	+4.89 ± 0.77
F34	4	0.4167 ± 0.0089	110.65 ± 6.45	+8.06 ± 0.80	+5.95 ± 0.77
F35	4	0.6006 ± 0.0097	22.64 ± 3.15	-4.67 ± 0.65	-6.39 ± 0.63
F36-1	4	0.6053 ± 0.0098	21.14 ± 3.13	-5.07 ± 0.65	-6.79 ± 0.63
F36-2	4	0.6136 ± 0.0100	18.53 ± 3.11	-6.39 ± 0.65	-8.09 ± 0.64
F39	4	0.4629 ± 0.0091	81.20 ± 5.10	+4.17 ± 0.74	+2.17 ± 0.72
F63	4	0.4604 ± 0.0091	82.65 ± 5.18	+4.67 ± 0.74	+2.66 ± 0.72
T04	4	0.5841 ± 0.0095	28.13 ± 3.27	-6.40 ± 0.64	-8.15 ± 0.63
T08	4	0.5560 ± 0.0099	38.24 ± 3.76	+1.54 ± 0.70	-0.28 ± 0.68
T56	4	0.6305 ± 0.0099	13.43 ± 2.90	-4.22 ± 0.64	-5.90 ± 0.62

Reconstructions based on the OGLS23 calibration of Daëron and Vermeesch (2024) as implemented by the `D47calib` library (v.1.3.1). Confidence intervals account for analytical error in Δ_{47} but not for calibration uncertainties, which remain below ±1 °C (95 % CL) in the range 0–50 °C.



Methods

Sample preparation and analysis

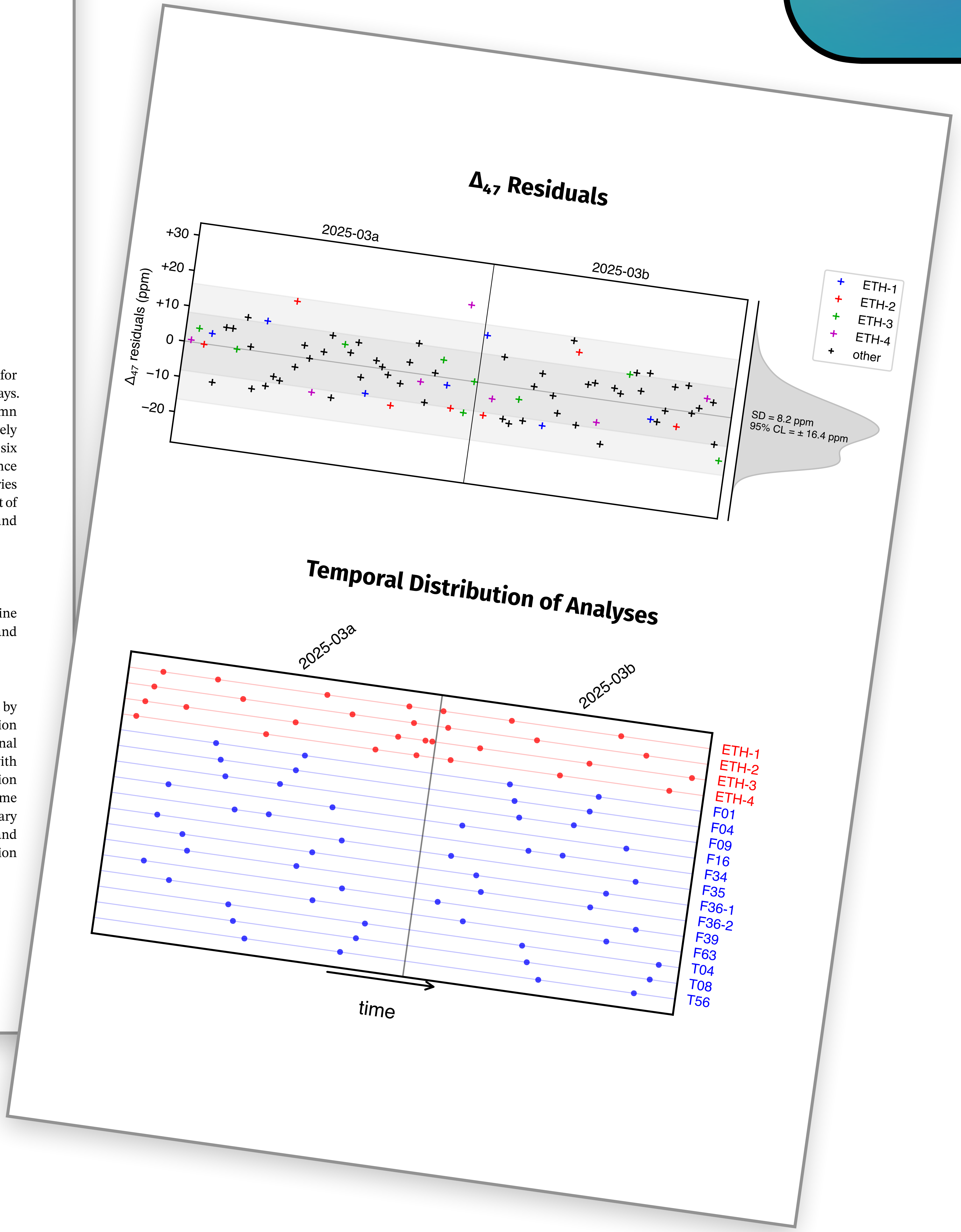
Carbonate samples were converted to CO_2 by phosphoric acid reaction at 90 °C in a common, stirred acid bath for 15 minutes. Initial phosphoric acid concentration was 103 % (1.91 g/cm³) and each batch of acid was used for 7 days. After cryogenic removal of water, the evolved CO_2 was helium-flushed at 20–25 mL/mn through a purification column packed with Porapak Q (50/80 mesh, 1 m length, 2.1 mm internal diameter) and held at −20 °C, then quantitatively recollected by cryogenic trapping and transferred into an Isoprime 100 dual-inlet mass spectrometer equipped with six Faraday collectors (m/z 44–49). Each analysis took about 2.5 hours, during which analyte gas and working reference gas were allowed to flow from matching, 10 mL reservoirs into the source through deactivated fused silica capillaries (65 cm length, 110 µm internal diameter). Every 20 minutes, gas pressures were adjusted to achieve m/z = 44 current of 80 nA, with differences between analyte gas and working gas generally below 0.1 nA. Pressure-dependent background current corrections were measured 12 times for each analysis.

IRMS data processing

All background measurements from a given session within ±6 hours of any given analysis were used to determine a mass-specific relationship for that analysis, linking background intensity (Z_m), total m/z,=,44 intensity (I_{44}), and time (t), with P being a polynomial of degree 2 to 4:

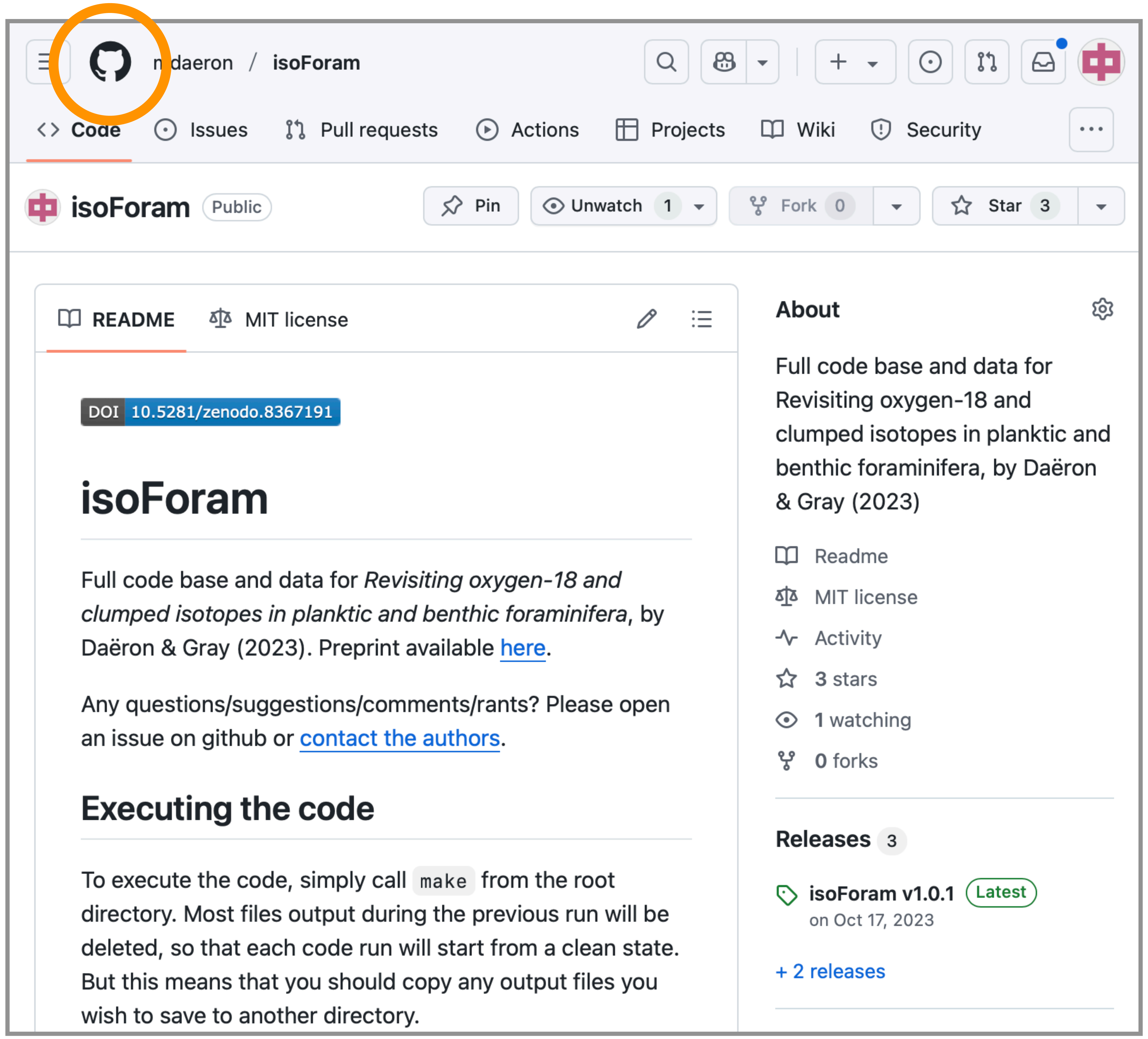
$$Z_m = aI_{44} + P(t)$$

Background-corrected ion current ratios (δ_{45} to δ_{49}) were converted to $\delta^{13}C$, $\delta^{18}O$, and "raw" Δ_{47} values as described by Daëron et al. (2016) using the IUPAC oxygen-17 correction parameters (Brand et al., 2010). The isotopic composition ($\delta^{13}C$, $\delta^{18}O$) of unknown samples was standardized using an affine ("two-point") correction based on the nominal $\delta^{13}C_{VPDP}$ and $\delta^{18}O_{VPDB}$ values of the ETH carbonate standards (Bernasconi et al., 2018). The same standards, along with an oxygen-18 acid fractionation factor of 1.00813 (Kim et al., 2007), were used to compute the isotopic composition ($\delta^{13}C$, $\delta^{18}O$) of our working reference gas. Raw Δ_{47} values were then converted to the I-CDES reference frame (Bernasconi et al., 2021) using a pooled regression approach (Daëron, 2021) as implemented by the `D47crunch` library (v.2.4.2). Full analytical errors are derived from the external reproducibility of unknowns and standards ($N_f = 62$) and conservatively account for the uncertainties in raw Δ_{47} measurements as well as those associated with the conversion to the I-CDES reference frame (Daëron, 2021).



Sharing and referencing your work (Zenodo, GitHub...)

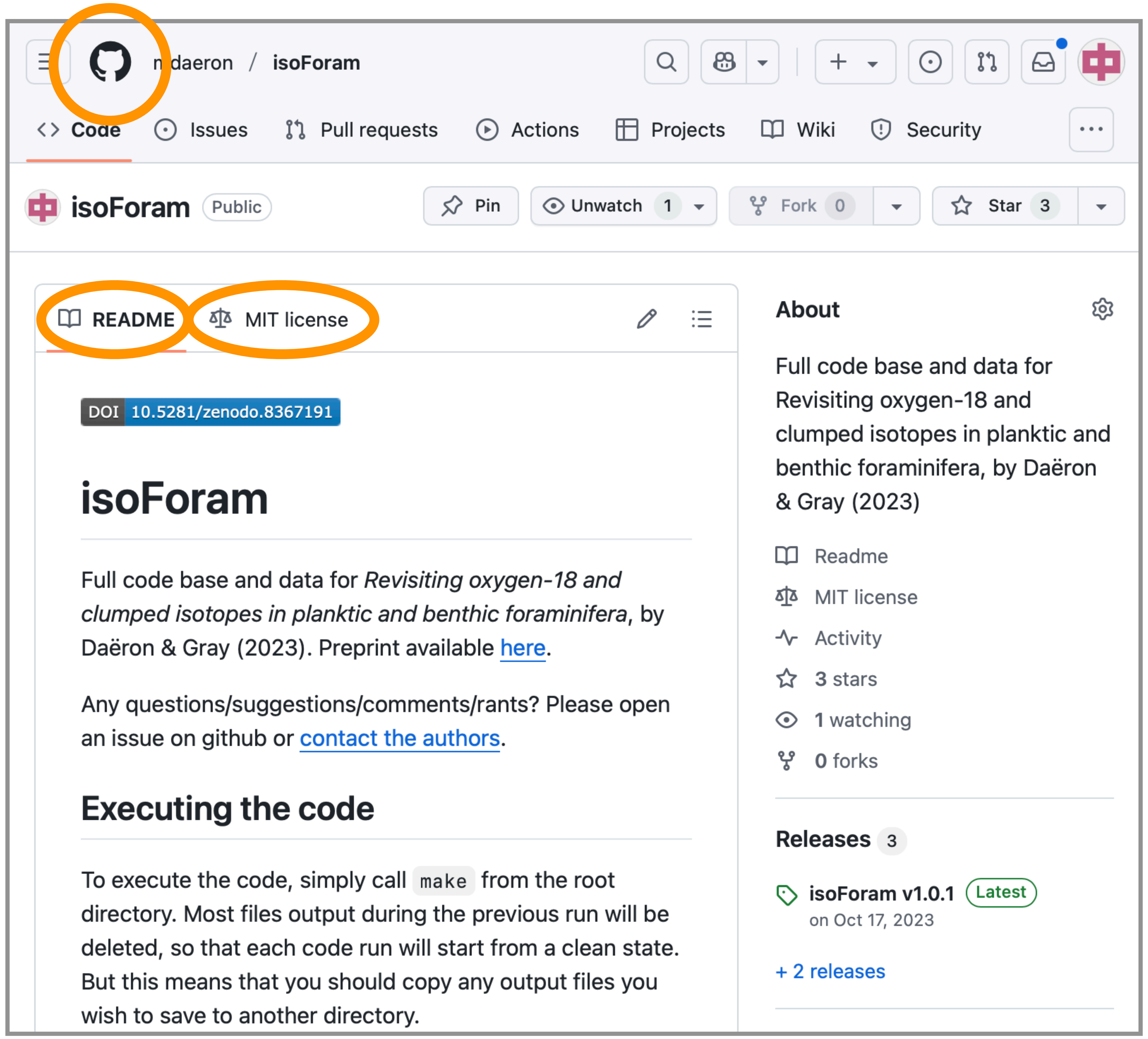
Backup your project to an online code repository



Sharing and referencing your work (Zenodo, GitHub...)

Add a readme and a license (I recommend MIT)

Backup your project to an online code repository

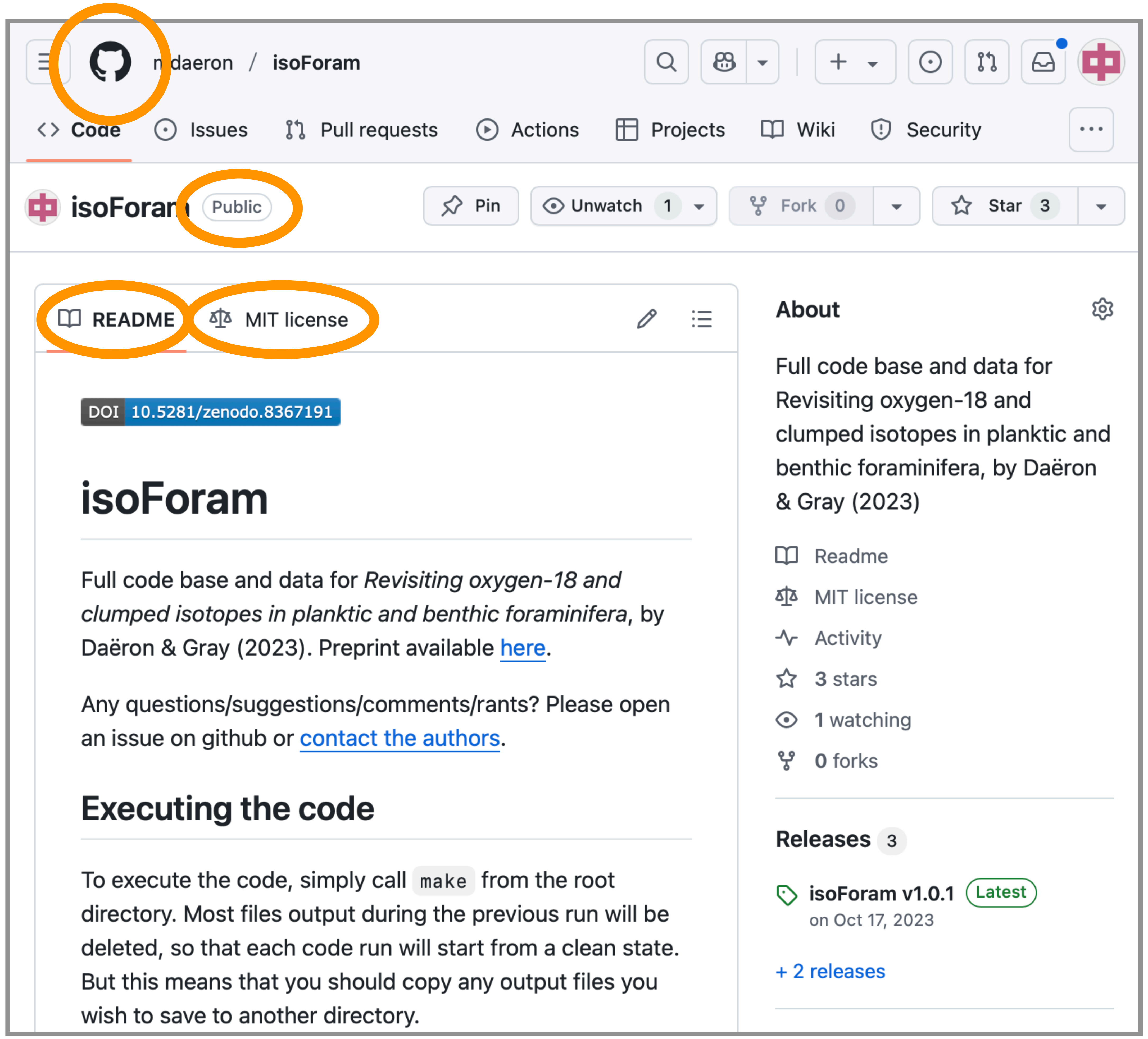


Sharing and referencing your work (Zenodo, GitHub...)

Add a readme and a license (I recommend MIT)

Backup your project to an online code repository

Make it public when you're ready



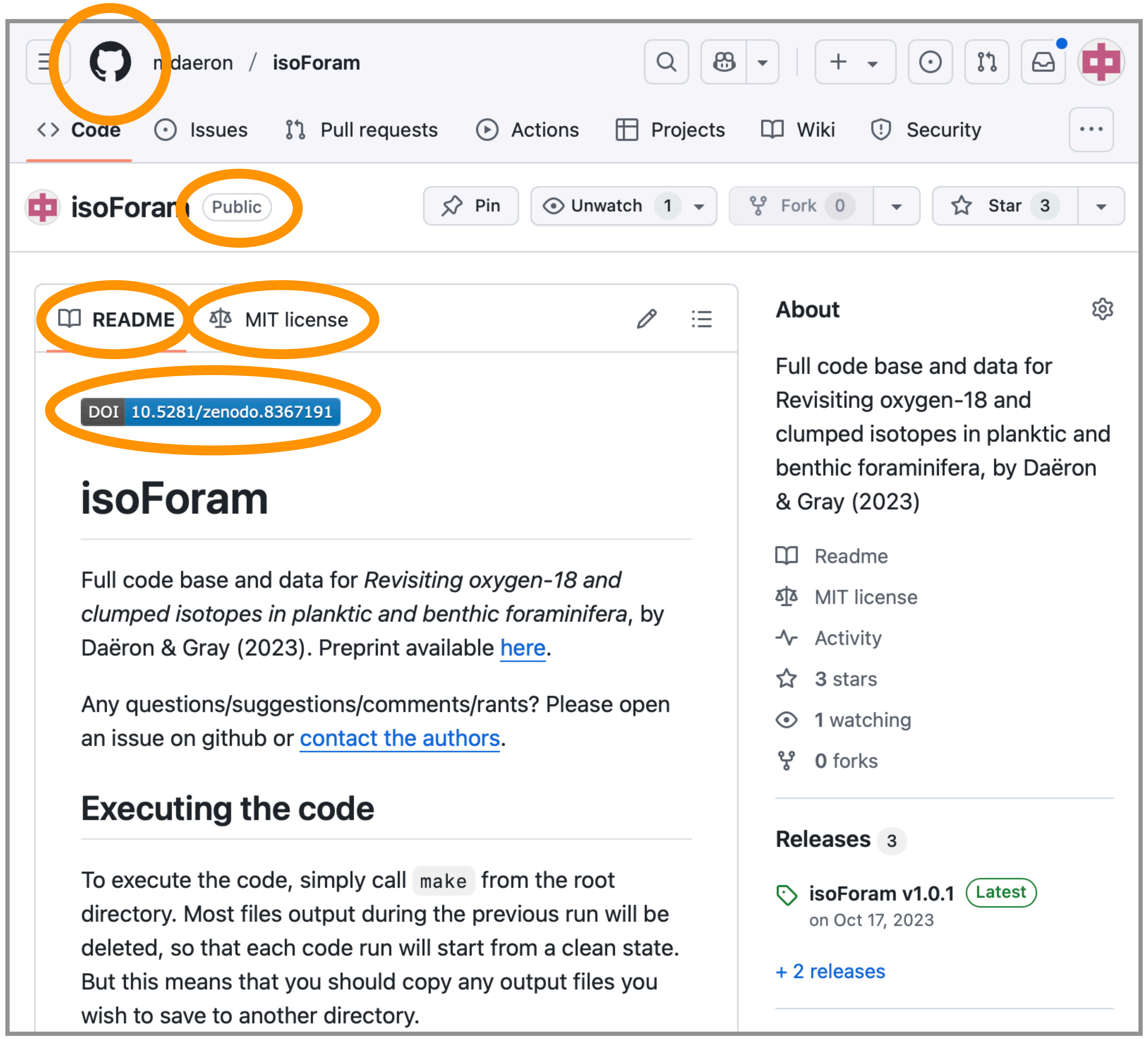
Sharing and referencing your work (Zenodo, GitHub...)

Add a readme and a license (I recommend MIT)

Backup your project to an online code repository

Make it public when you're ready

Publish versions of record with a permanent DOI (I recommend *zenodo.org*)



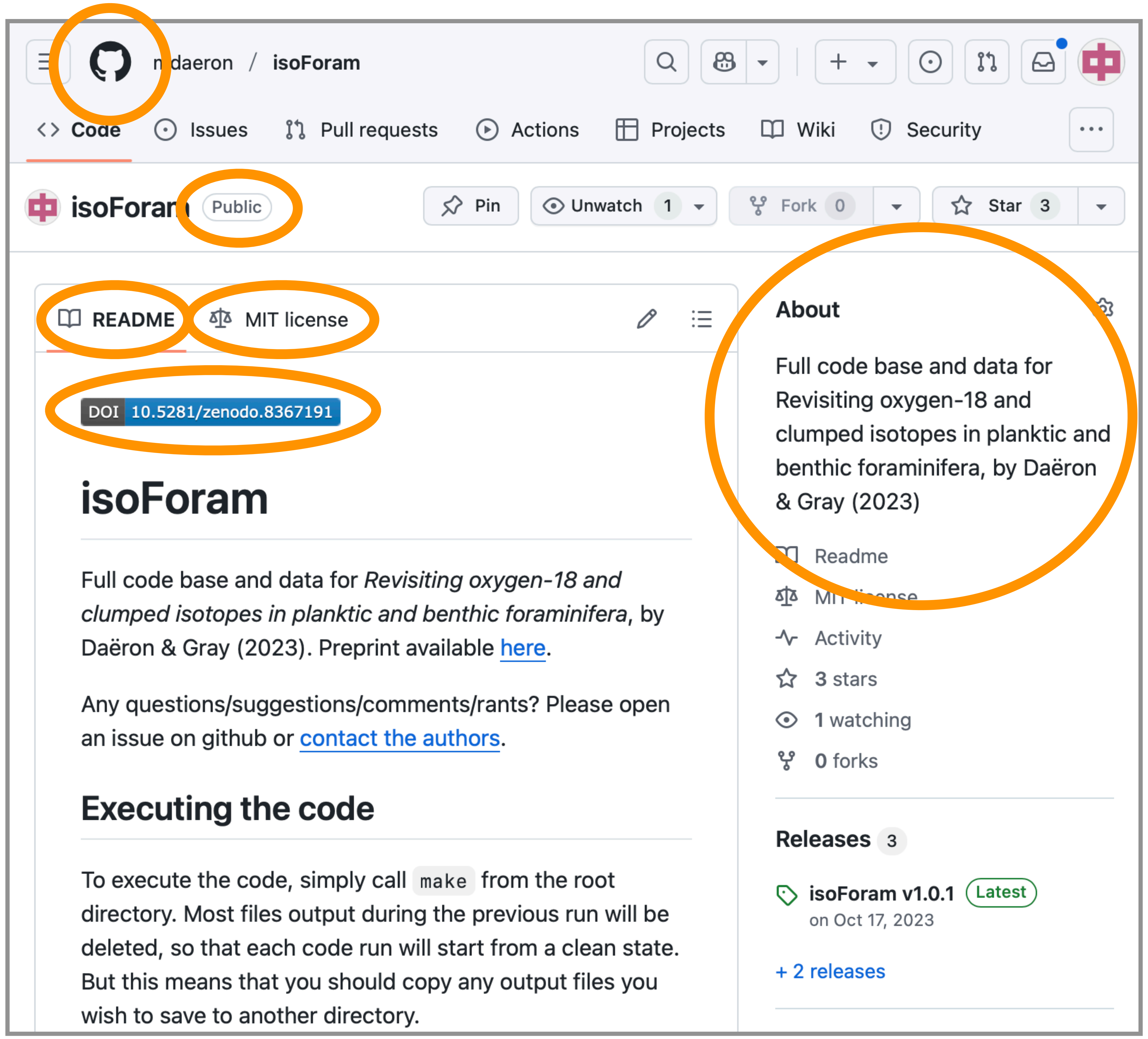
Sharing and referencing your work (Zenodo, GitHub...)

Add a readme and a license (I recommend MIT)

Backup your project to an online code repository

Make it public when you're ready

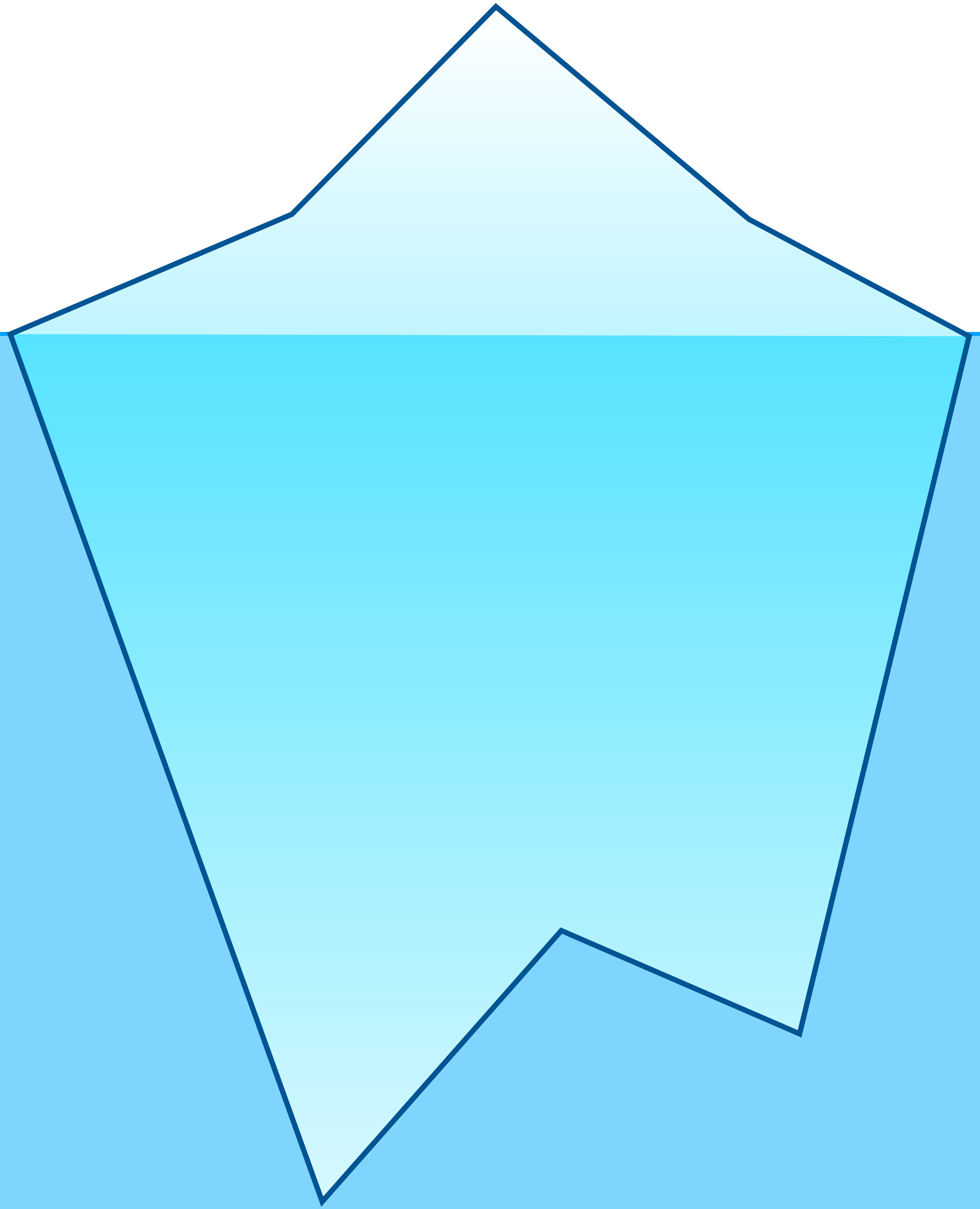
Publish versions of record with a permanent DOI (I recommend *zenodo.org*)



Refer to the peer-reviewed article

Yes you can!

- None of these tools are very difficult to learn.
- All of them are free and open-source.
  (so your work will be reusable decades from now)
- Very worthwhile investment, particularly
  (but not only) for students & early-career



Good researchers copy;
great researchers fork.

(paraphrased after W. Faulkner,
I. Stravinsky, P. Picasso, and S. Jobs)