Why does reproducibility matter?

Childhood Cancer Data Lab

Have you ever had problems reproducing…

- Someone else's research?
- Your research?
- Both?

What are your barriers to reproducible science?

Some barriers others have noticed…

- Technical factors
  - Natural variation
  - Batch effects and sensitivity to equipment or experimental conditions
- Study design and statistics
  - Selective reporting and "P-hacking"
- Rewards and incentives
  - Hypercompetition and "publish or perish" culture
  - Paywalls
- Human factors
  - Confirmation bias
  - Poor record keeping and/or sharing

Adapted from https://www.repro4everyone.org/resources/intro-reproducibility

Reproducibility vs replication

- Reproducibility
  - Authors provide all the necessary data and computer code to re-run the analysis and re-create the results
  - The exact same data/code are used to re-derive the exact same results
- Replication
  - A separate study arrives at the same scientific findings as another study
  - New data/code and analyses are performed that identify consistent results with previous work

	Same data	**Different data**
Same methods	Reproducibility	Replicability
Different methods	Robustness	Generalizability

Adapted from https://www.repro4everyone.org/resources/intro-reproducibility

Reproducibility

Reproducibility = "*The ability of independent analysts to recreate the results claimed by the original authors using the original data and analysis techniques.*"

Goals of reproducible research:

- Provide a record of the processes used to analyze data underlying a work, which can allow for understanding beyond what's in a methods section
- Ensure integrity and transparency into an analysis
- *Bonus:* Allows for work to be built on by others

Quote and goals from Peng and Hicks. 2021.

Why do we like reproducible science?

- Reproducibility supports…
  - You!
  - Your collaborators and team!
  - Your community!
  - The scientific endeavor!

- Reproducibility makes your funders and journals happy

Why does working reproducibly support you?

Access to your code on multiple computers and an ability to roll back experimental changes to code

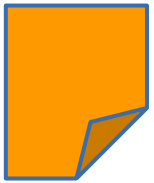
A consistent way to manage your software environment between revisions that are months apart

A record of data downloaded from public resources

A record of what you did and why when it comes time to write your methods section

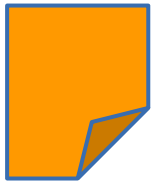
Scenario 1: The email attachment

A postdoc emails a script `code-final.R` to their lab and a collaborator



`code-final.R`

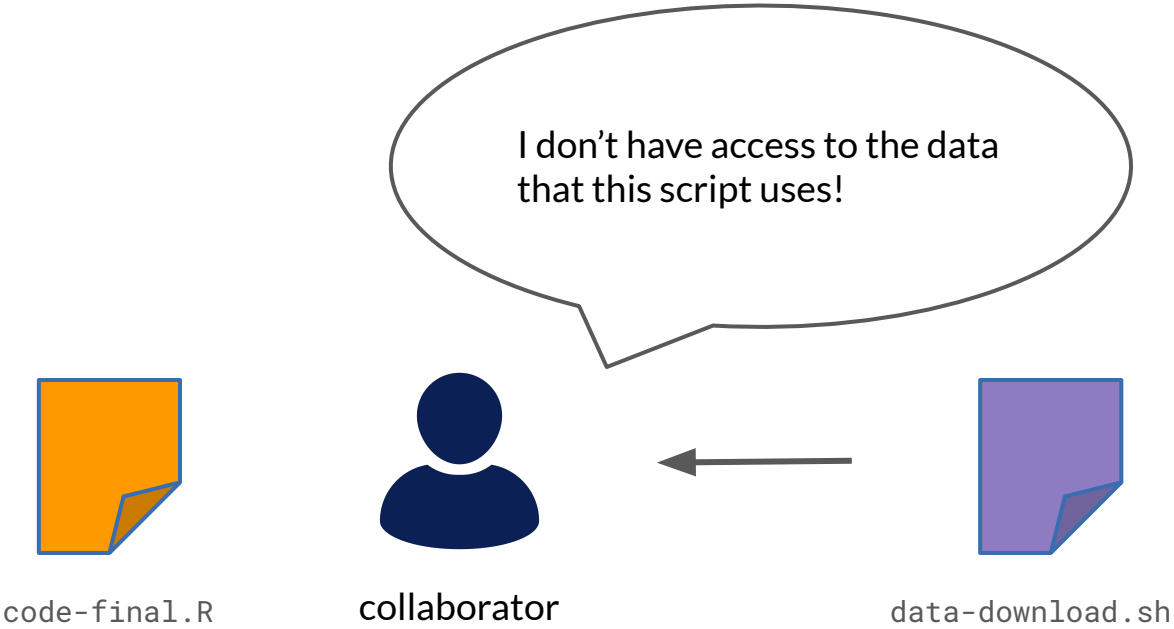
labmate



`code-final.R`

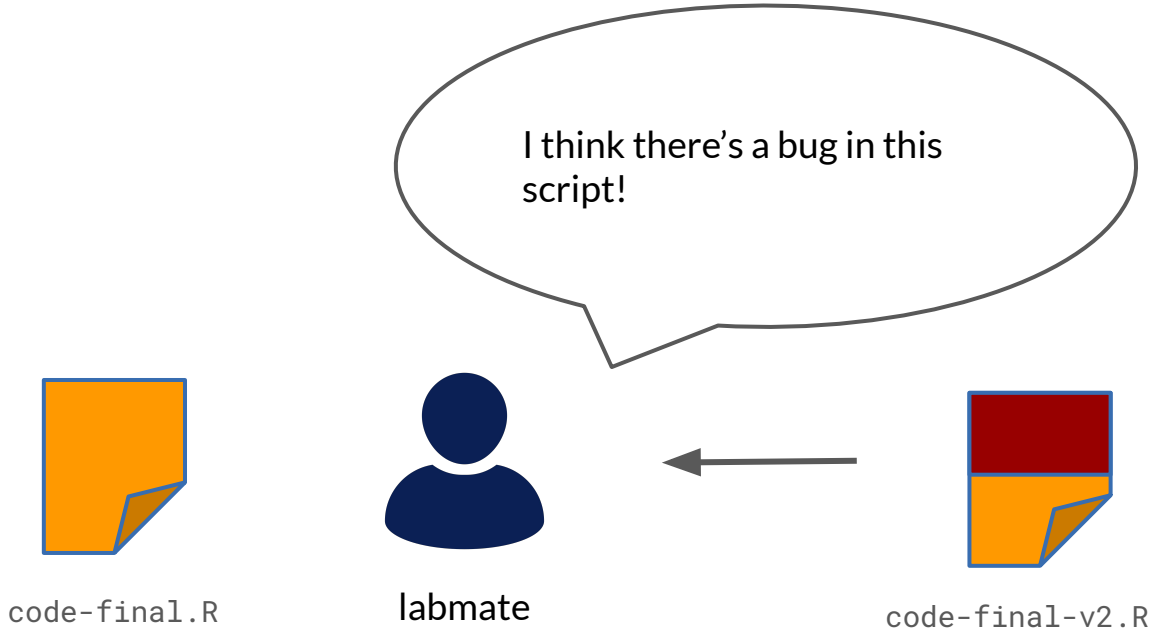
collaborator

Scenario 1: The email attachment



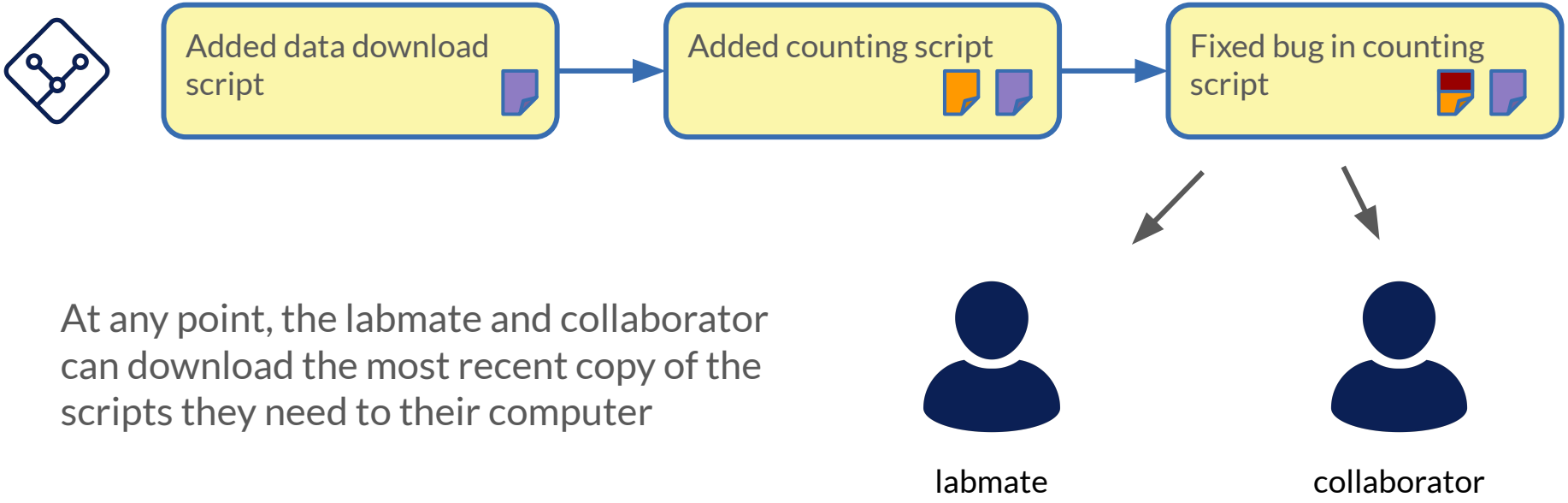
The data download script gets sent as another email attachment

Scenario 1: The email attachment



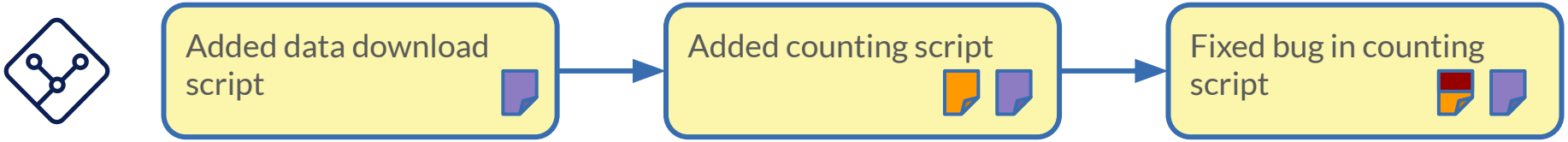
A new copy of the script gets circulated, but the collaborator isn't cc'd and continues to use the script with a bug

Scenario 2: Version control

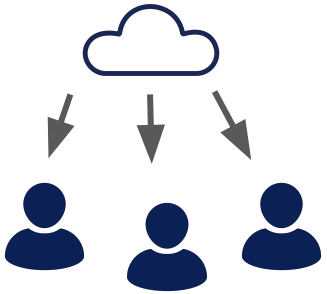


At any point, the labmate and collaborator can download the most recent copy of the scripts they need to their computer

Version control for reproducible research

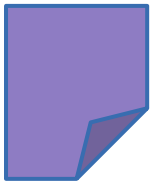


To meet the definition of reproducibility, we need a shared understanding of “original data and analysis techniques”



Peng and Hicks. 2021.

Scripting can help us have a shared understanding of “original data”



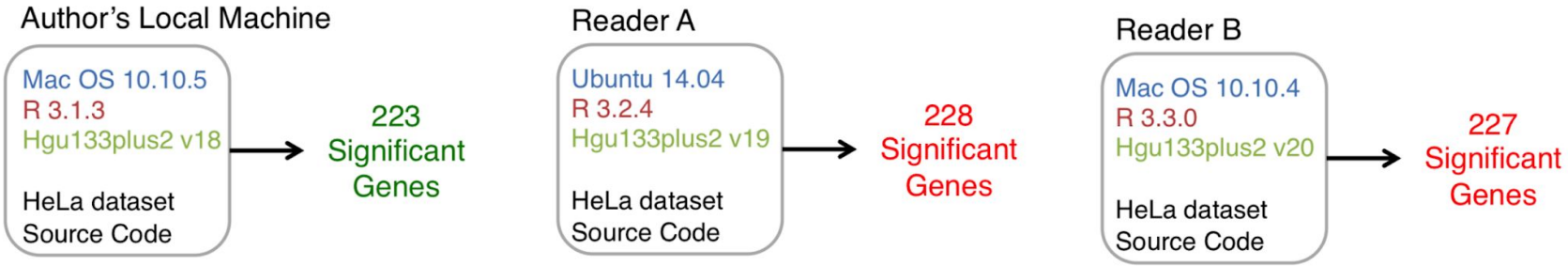
`data-download.sh`

The “original data” might mean a specific release of something underpinning your results. For example:

- A genome build
- A freeze associated with a point in time during a longitudinal study

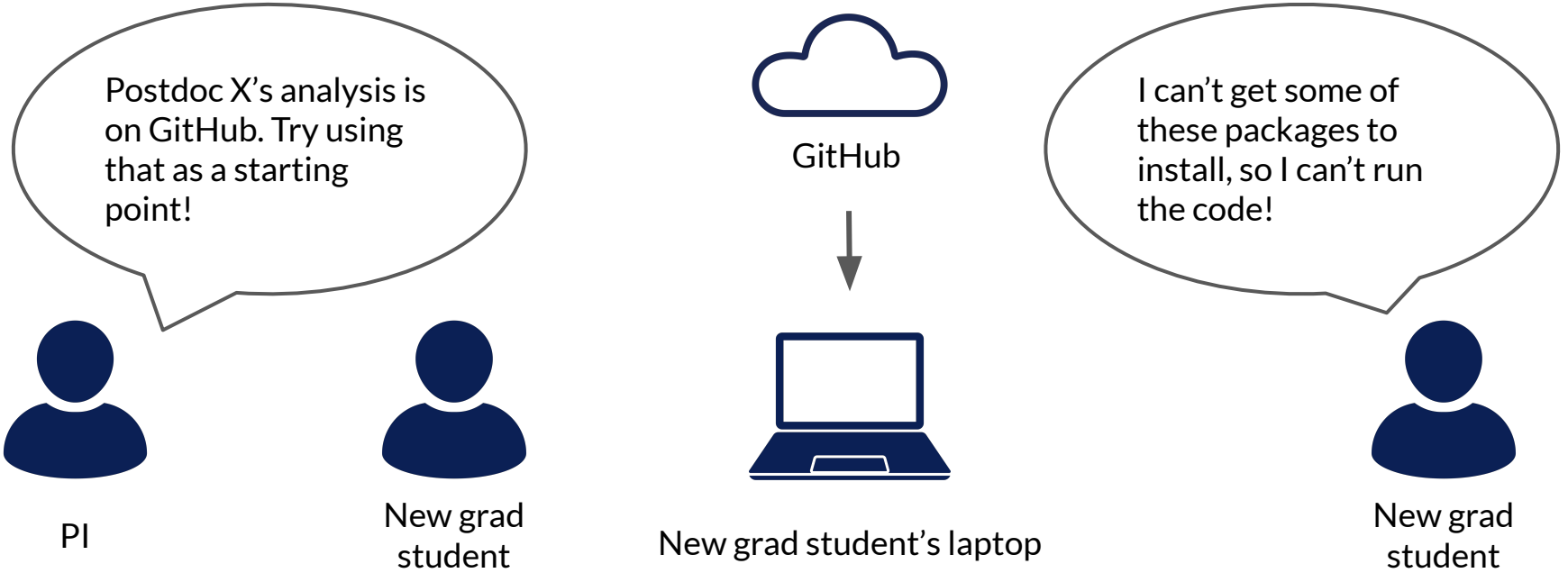
Standardizing what data someone downloads in a script means they get whatever the right data is for the project (including you on your laptop vs. the server!).

If we don't control for software versions, we might get different results (or use different analysis techniques if the defaults change)

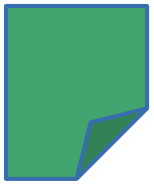


Adapted from [Beaulieu-Jones and Greene. 2018.]

But also, installing software is sometimes a headache!



Package and environment management systems can simplify installation

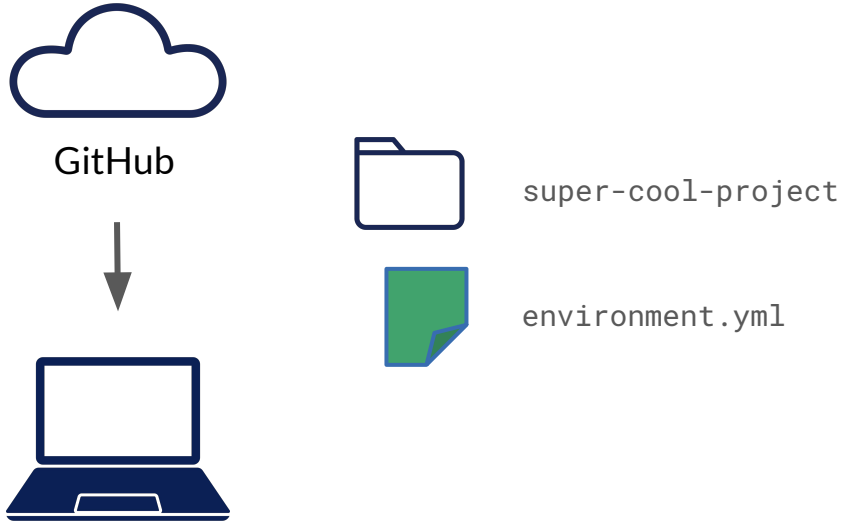


`environment.yml`

Package managers create “recipes” for building environments that can include all the packages required for the project. The exact files will vary depending on the system you use.

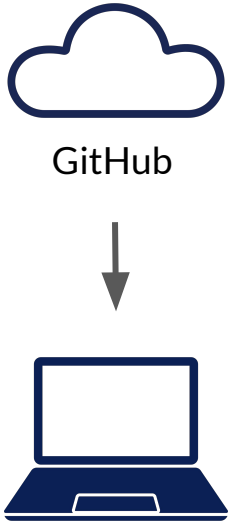
The “recipes” can live under version control in the Git repository.

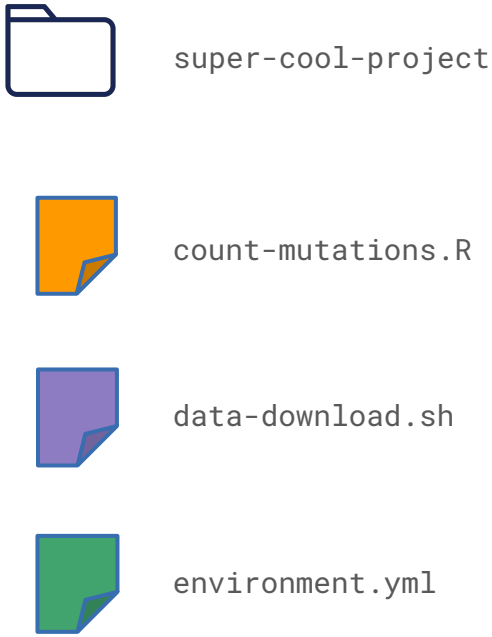
Package and environment management systems can simplify installation



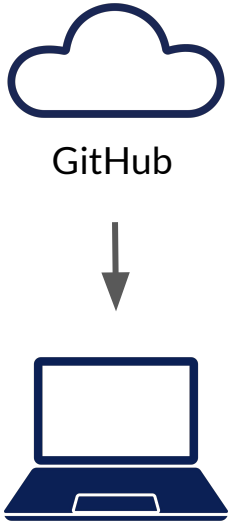
With the installation of our package manager of choice and a few short commands, we can have a working environment.

Scripts and environment files aren't always enough





Documentation provides important context



The README might tell you:

- What the goal of the project is
- How to setup your environment
- How to run the scripts and in what order

Writing it down: An important part of science

Always 👏 Be 👏
Documenting

Science at the computer is just like science at the bench!

We need to record our steps, even if they feel obvious at the time.

Future You is unimpressed with your code

We tend to feel code we are writing is clear and obvious *in the moment we are writing it*.

Future You does not agree! They have no idea what you were thinking and why you were thinking it.

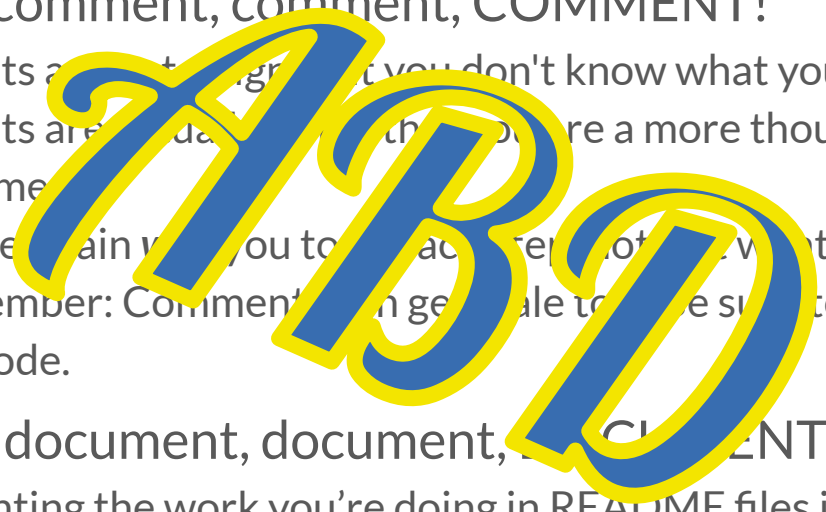
And if Future You doesn't understand what's going on, odds are others won't either.

How can you appease Future You?

- Comment, comment, comment, COMMENT!
  - Comments are not a sign that you don't know what you are doing
  - Comments are actually a sign that you are a more thoughtful (and advanced?) programmer
  - Strive to explain ***why*** you took each step, not just what the step was
  - But remember: Comments can get stale too! Be sure to update them when you update code.
- Document, document, document, DOCUMENT!
  - Documenting the work you're doing in README files is (part of) your lab notebook
  - Documenting your code can also help you gain insight into your code and find ways to improve it

How can you appease Future You?

- Comment, comment, comment, COMMENT!
  - Comments a[illegible] you don't know what you are doing
  - Comments ar[illegible]re a more thoughtful (and advanced?) programme[illegible]
  - Strive to e[illegible]ain w[illegible]ou to[illegible]ac[illegible]ep[illegible]ot[illegible]t the step was
  - But remember: Comment[illegible]n ge[illegible]ale to[illegible]e su[illegible]to update them when you update code.
- Document, document, document, [illegible]CU[illegible]ENT!
  - Documenting the work you're doing in README files is (part of) your lab notebook
  - Documenting your code can also help you gain insight into your code and find ways to improve it



The practices we teach in this workshop can help!

Access to your code on multiple computers and an ability to roll back experimental changes to code

A consistent way to manage your software environment between revisions that are months apart

A record of data downloaded from public resources

A record of what you did and why when it comes time to write your methods section

Every little bit helps

- Everyone has to start somewhere!
- Nobody's code is perfect!

WORLD VIEW *A personal take on events*

Publish your computer code: it is good enough

Freely provided working code — whatever its quality — improves programming and enables others to engage with your research, says **Nick Barnes**.

https://doi.org/10.1038/467753a

Reproducibility standards for machine learning in the life sciences

To make machine-learning analyses in the life sciences more computationally reproducible, we propose standards based on data, model and code publication, programming best practices and workflow automation. By meeting these standards, the community of researchers applying machine-learning methods in the life sciences can ensure that their analyses are worthy of trust.

Benjamin J. Heil, Michael M. Hoffman, Florian Markowetz, Su-In Lee,
Casey S. Greene and Stephanie C. Hicks

Table 1 | Proposed reproducibility standards

	Bronze	Silver	Gold
Data published and downloadable	x	x	x
Models published and downloadable	x	x	x
Source code published and downloadable	x	x	x
Dependencies set up in a single command		x	x
Key analysis details recorded		x	x
Analysis components set to deterministic		x	x
Entire analysis reproducible with a single command			x

https://doi.org/10.1038/s41592-021-01256-7

ONE CUP AT A TIME CLUB

But remember - It's an honor just to compete!