GEOS F436/636 Beyond the Mouse

Christine (Chris) Waigl

University of Alaska Fairbanks – Fall 2018

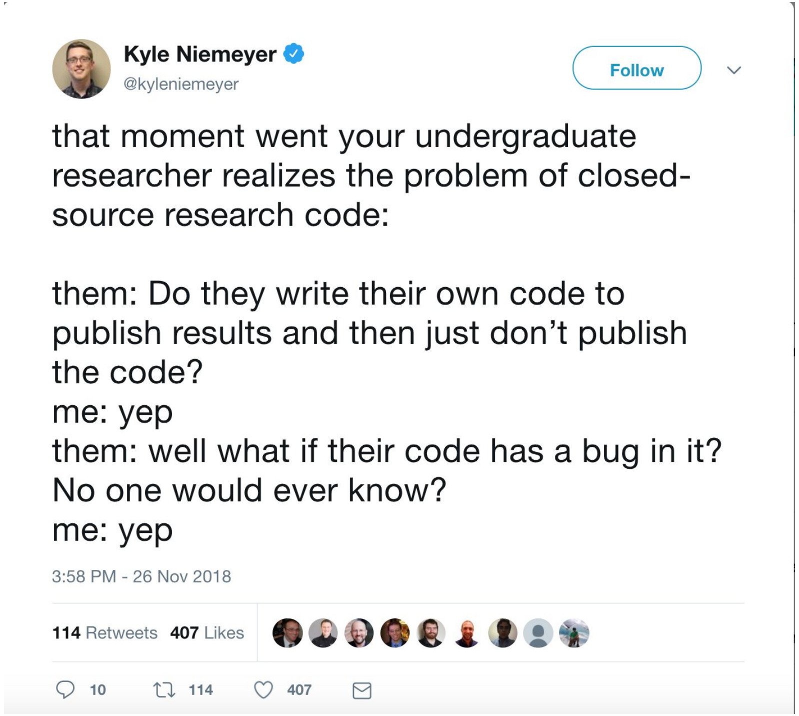
Week 14: More on tools for science

Topics for week 14

- Intro to reproducible science
- Some more on git for scientists

Reproducible science

Reproducibility is a cornerstone of scientific work



Kyle Niemeyer
@kyleniemeyer

that moment went your undergraduate researcher realizes the problem of closed-source research code:

them: Do they write their own code to publish results and then just don't publish the code?
me: yep
them: well what if their code has a bug in it? No one would ever know?
me: yep

3:58 PM - 26 Nov 2018

114 Retweets 407 Likes

- It means that other researchers are able to re-run experiments, analysis of observational data or theoretical proofs, and re-use scientific results.
- Several fields of science have experienced a "reproducibility crisis"
- Peer review doesn't ensure reproducibility.
- In computational fields, most scientists' workflow does not optimize for reproducible science.

Selfish reasons for reproducible science

- Reproducibility helps to avoid errors.
- Reproducibility makes it easier to write papers.
- Reproducibility helps convince reviewers and other scientists.
- Reproducibility enables continuity of your work.
- Reproducibility helps to build your reputation.

Reproducibility is enabled by practices

Share, publish and archive all research products (including code & data).

- Share: make available to others.
- Publish: make citeable, discoverable, findable.
- Archive: enable long-term preservation.

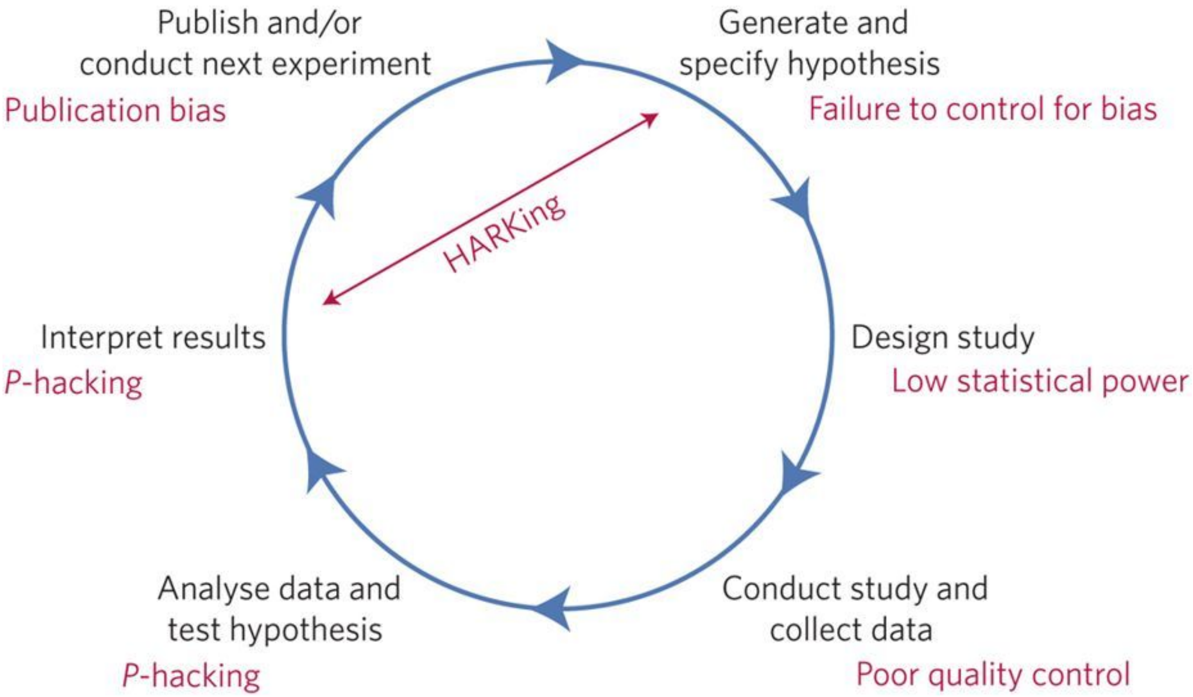
Supported by:

- Software tools (R, Python have communities dedicated to reproducibility)
- Online repositories (Digital Object Identifiers (DOIs), persistent tags, metadata, catalogs...)
- Community practices

Open research practices aren't always easy...

HARK = Hypothesizing after results are known.

Munafò, Marcus R., Brian A. Nosek, Dorothy V. M. Bishop, Katherine S. Button, Christopher D. Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J. Ware, and John P. A. Ioannidis. “A Manifesto for Reproducible Science.” Nature Human Behaviour 1, no. 1 (January 2017): 0021. https://doi.org/10.1038/s41562-016-0021 .


Generate and specify hypothesis
Failure to control for bias
Design study
Low statistical power
Conduct study and collect data
Poor quality control
Analyse data and test hypothesis
P-hacking
Interpret results
P-hacking
Publish and/or conduct next experiment
Publication bias
HARKing

Cf. Victoria Stodden et al. (2014)

Computational reproducibility: when detailed information is provided about code, software, hardware and implementation details.

Empirical reproducibility: when detailed information is provided about non-computational empirical scientific experiments and observations. In practise this is enabled by making data freely available, as well as details of how the data was collected.

Statistical reproducibility: when detailed information is provided about the choice of statistical tests, model parameters, threshold values, etc. This mostly relates to pre-registration of study design to prevent p-value hacking and other manipulations.

Computational reproducibility exists on a scale:

Reviewable Research. The descriptions of the research methods can be independently assessed and the results judged credible.

Replicable Research. Tools are made available that would allow one to duplicate the results of the research, for example by running the authors' code to produce the plots shown in the publication.

Confirmable Research. The main conclusions of the research can be attained independently without the use of software provided by the author.

Auditable Research. Sufficient records (including data and software) have been archived so that the research can be defended later if necessary or differences between independent confirmations resolved. The archive might be private, as with traditional laboratory notebooks.

Open or Reproducible Research. Auditable research made openly available. This comprised well-documented and fully open code and data that are publicly available that would allow one to (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

Open science practices are increasingly required

... by funding agencies:

- NSF: https://www.nsf.gov/bfa/dias/policy/dmp.jsp
- US Office of Management and Budget (governs NASA) https://project-open-data.cio.gov/policy-memo/

... by journals:

- Nature: https://www.nature.com/authors/policies/availability.html
- PLOS One: https://journals.plos.org/plosone/s/editorial-and-publishing-policies

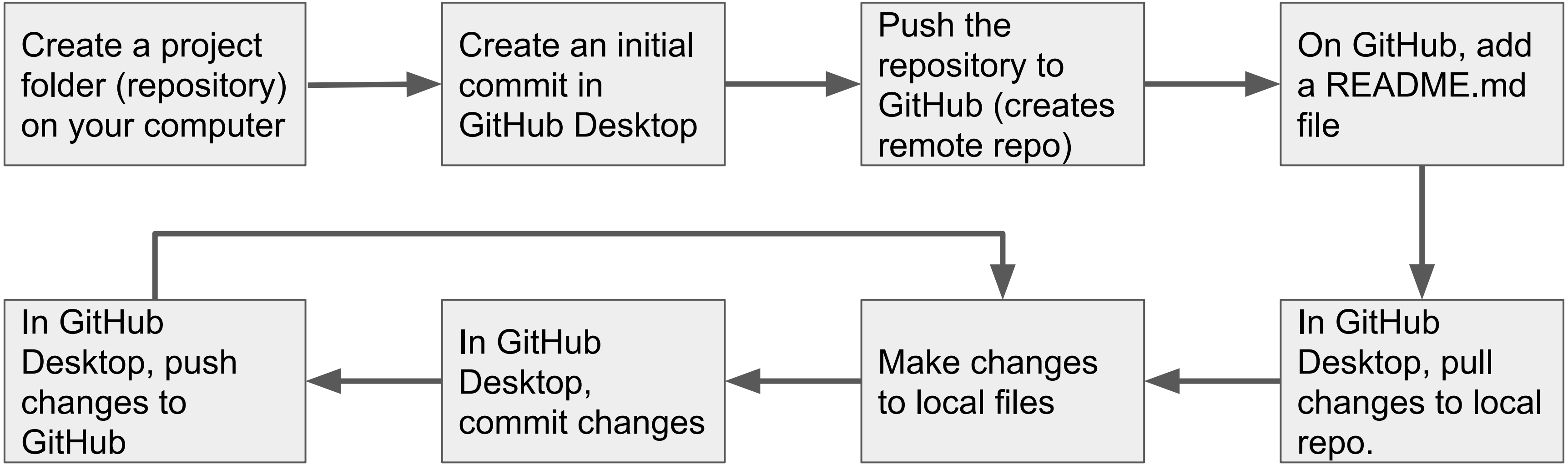
Further reading

- R Open Science Reproducibility Guide: http://ropensci.github.io/reproducibility-guide/sections/introduction/
- Konkol, Markus et al.. “Computational Reproducibility in Geoscientific Papers: Insights from a Series of Studies with Geoscientists and a Reproduction Study.” International Journal of Geographical Information Science 33, no. 2 (February 1, 2019): 408–29. https://doi.org/10.1080/13658816.2018.1508687 .
- Munafò, Marcus et al.. “A Manifesto for Reproducible Science.” Nature Human Behaviour 1, no. 1 (January 2017): 0021. https://doi.org/10.1038/s41562-016-0021 .
- Peng, Roger: Reproducibility for *Biostatistics* https://academic.oup.com/biostatistics/article/10/3/405/293660
- Report by Victoria Stodden: http://stodden.net/icerm_report.pdf

More on git/GitHub

We've only scratched the surface

We have established a basic workflow for git and GitHub using GitHub Desktop. It goes like this:



Create a project folder (repository) on your computer

Create an initial commit in GitHub Desktop

Push the repository to GitHub (creates remote repo)

On GitHub, add a README.md file

In GitHub Desktop, pull changes to local repo.

Make changes to local files

In GitHub Desktop, commit changes

In GitHub Desktop, push changes to GitHub

Git has a lot more to offer!

- You can use the git command from the command line instead of GitHub Desktop to carry out the git workflow. The commands you need will be:
    - `git status, git add, git commit (-m "commit message"), git push, git pull`
- The next step is to add more git functionality to your workflow. Specifically, branching and merging. Each time you work on a new feature of your code, your development should go in a new branch. This leaves the old, functioning code alone in your "master" branch. Once you're done with developing your new code, you can merge the changes into the master branch.
- A great place to learn more is the tutorial published by Atlassian/Bitbucket (a commercial competitor of GitHub): https://www.atlassian.com/git/tutorials/setting-up-a-repository . Read the sections "Getting Started" and "Collaborating".