@jhagenberg

@jonas-hag

MAX PLANCK INSTITUTE
OF PSYCHIATRY

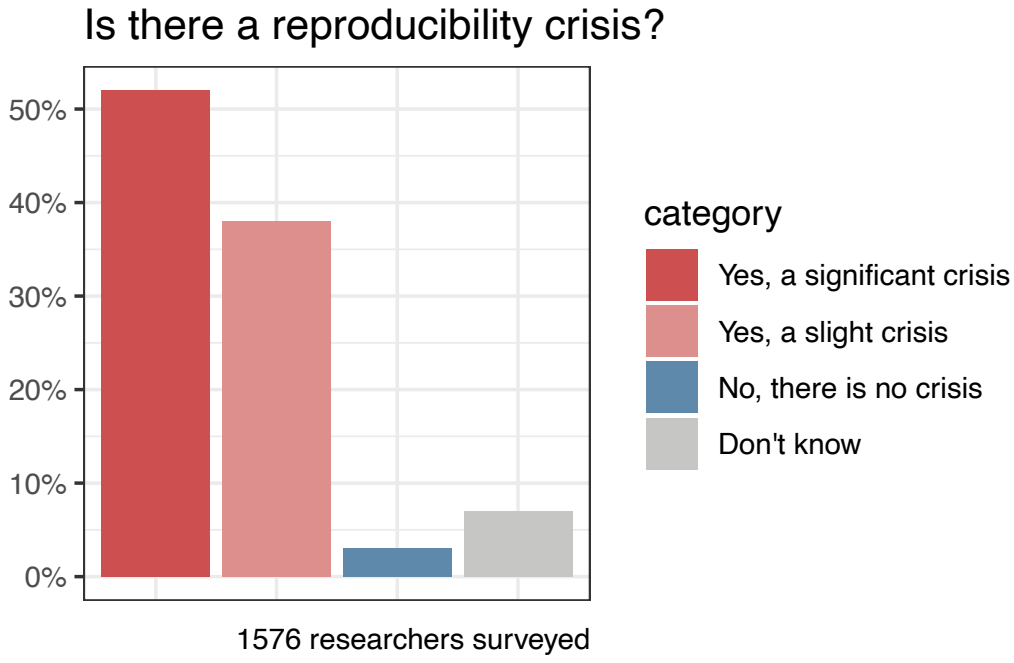
The CodeClub at the MPIP

Future Opportunities for Software in Research

12.05.2022

Jonas Hagenberg
&
Linda Dieckmann

Relevancy



Data from Baker (2016)

What factors contribute to irreproducible research?

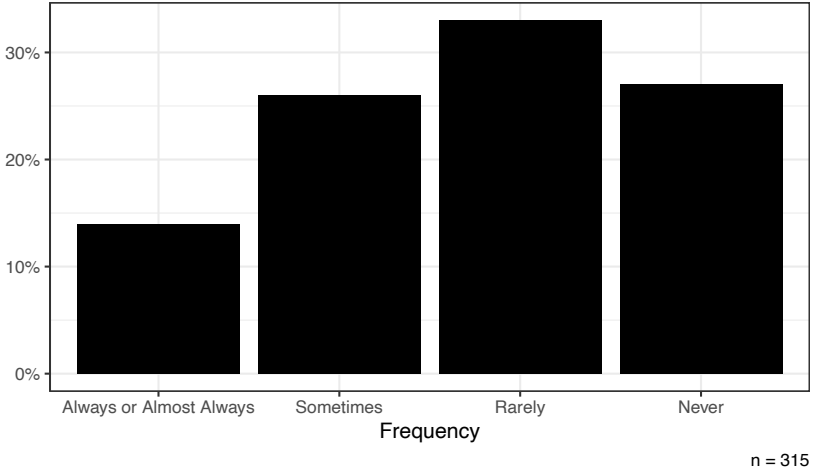
Insufficient oversight/mentoring: ca. **50%** say it always/often contributes

Methods, code unavailable: ca. **45%** say it always/often contributes

Relevancy

Code review in research?

How often is your analysis code for a paper reviewed by someone else before submission [...]?



Data from Vable et al. (2021) from an informal twitter poll

The CodeClub at the MPIP

Motivation:

- Minimise errors in the code
- More exchange between researchers
- Improve code quality

The CodeClub at the MPIP

Motivation:

- Minimise errors in the code
- More exchange between researchers
- Improve code quality

Situation at the MPIP:

- Interdisciplinary

Biostatistics

Psychology

Bioinformatics

Medicine

Neuroscience

The CodeClub at the MPIP

Motivation:

- Minimise errors in the code
- More exchange between researchers
- Improve code quality

Situation at the MPIP:

- Interdisciplinary
- Different programming languages

R Matlab

Python

The CodeClub at the MPIP

Motivation:

- Minimise errors in the code
- More exchange between researchers
- Improve code quality

Situation at the MPIP:

- Interdisciplinary
- Different programming languages
- Individual projects quite diverse

Rather data analysis than software development

Rarely working as a team together on *one* project

The CodeClub at the MPIP

Structure:

- Meetings: 1x per month, ca. 1h
- Format:
    - Short presentations (ca. 10 min) about a relevant topic
    - Short exercises in small groups
- Code Review:
    - Platform to find a partner
    - Help with how to perform a code review
    - Discuss problems and experiences in the group

Prior art

PLOS COMPUTATIONAL BIOLOGY

EDUCATION

Ten simple rules to increase computational skills among biologists with Code Clubs

Ada K. Hagan[1¤a], Nicholas A. Lesniak[1], Marcy J. Balunas[2], Lucas Bishop[1], William L. Close[1], Matthew D. Doherty[1], Amanda G. Elmore[1], Kaitlin J. Flynn[1¤b], Geoffrey D. Hannigan[1¤c], Charlie C. Koumpouras[1¤d], Matthew L. Jenior[1¤e], Ariangela J. Kozik[3], Kathryn McBride[1], Samara B. Rifkin[3], Joshua M. A. Stough[1], Kelly L. Sovacool[4], Marc A. Sze[1¤f], Sarah Tomkovich[1], Begum D. Topcuoglu[1¤c], Patrick D. Schloss[1]*

Implementation of the CodeClub

- Ask everyone to present – does not need to be polished
- Set realistic goals (only 1h)
- Use breakout rooms for exercises
- Mix experienced with less experienced participants
- Collect presentations at one place

Topics covered so far:

- How to do Code Review
- Introduction to R package version control with renv
- How to get a DOI for your code repository (Zenodo)
- Programming setup for remote servers with Sublime Text
- Introduction to singularity
- Introduction to snakemake
- Introduction to conda environments

Example tutorial

renv exercise

Jonas Hagenberg

17.11.2021

Part A

1. Create a new RStudio project (with github) and initialise `renv`
2. Copy `example_script.R` into your project and install the necessary packages
3. Execute the code in `example_script.R`
4. Update the `renv` lockfile
5. Push the changes to github
6. Share the repository with your partner

Part B

1. Clone the repository you got from your partner
2. If not done already, install `renv`
3. Restore the package library with `renv::restore()`
4. Execute the code in `example_script.R` and compare the output of `sessionInfo()`

https://github.molgen.mpg.de/jonashagenberg/renv_tutorial

Code review Guidelines

For code authors

Prepare your code for the review:

- check with the reviewer how they want the code
  - should you just send the code files per mail?
  - best case: create a repository in GitHub, invite the reviewer as collaborator and assign them as a reviewer to a pull request
- provide enough context to understand your code/the changes. You can either link to an issue you solve with your pull request or provide additional materials such as a paper draft
- prepare the code (e.g., try to follow the style guides)
- don’t submit too many lines of code for review; if it’s a larger script you may mark the parts for the first code review

Code review Guidelines

For reviewers

Generals:

- be nice!
- communicate which ideas you feel strong about and those you don't
- ask open ended questions and ask for clarification
- offer and explain alternatives and workarounds
- stay empathetic and positive
- accept that many programming decisions are opinions
- ask questions, don't make demands
- talk in person if there are many things to clarify
- don't give strong, opinionated statements
- don't criticize the author but the code

Areas to check

Code style / understandability

- [ ] Does the code follow the style guide?
- [ ] Is the overall readability ok?
- [ ] If multiple scripts are given, is their naming meaningful and indicative of their order?
- [ ] Are unnecessary duplications of code snippets avoided?
- [ ] Is the script organized, so it is easy to follow what the code does?
- [ ] Is not all code in one big method?
- [ ] Do the comments also explain why you do something, not only what?
- [ ] Is it clear when and by whom the code was written?

Functionality

- [ ] Does the code do what the documentation says it should?
- [ ] Are there any bugs?

Maintainability

- [ ] Does the code require the least amount of effort to support it in the future?
- [ ] Are up-to-date and reliable libraries and frameworks used?

Reproducibility

- [ ] Are the results easy to reproduce (e.g. by using a reproducible environment?
- [ ] Are seeds used when necessary?

Data protection (especially important for phase 2 before publishing)

- [ ] Do the scripts not contain any data that shouldn't be published?
  (e.g. as outputs in markdown; comments or even files stored in the directory)

Theory and practice…

- How to get PhD students do code review?
- Which kind of code review suits the situation?

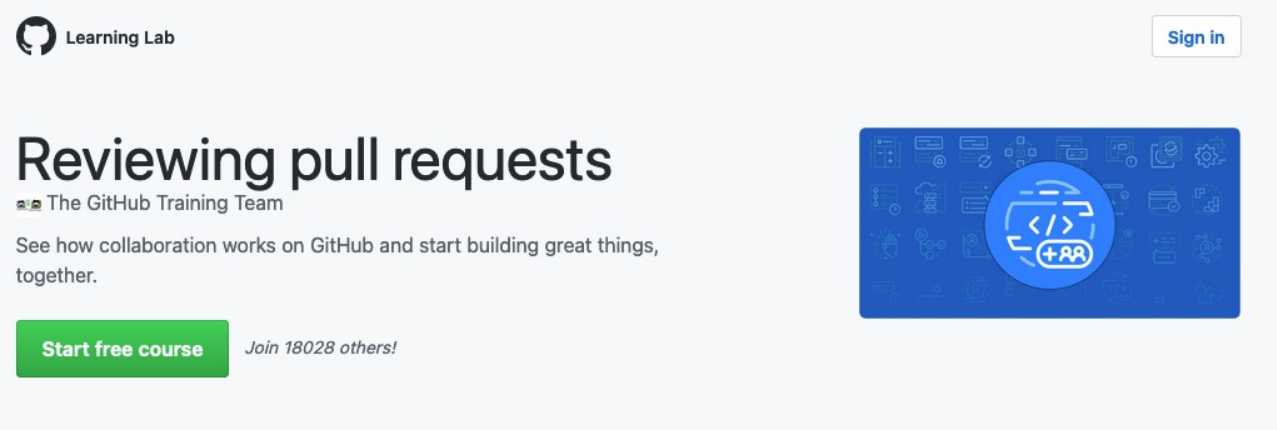
Code review - problems

- Many don’t know how to use GitHub and pull requests
- Often data analysis scripts instead of software development
- Code review is time-consuming
- Code review is not considered an important contribution

Code review - solutions

Many don’t know how to use GitHub and pull requests

Training



https://lab.github.com/githubtraining/reviewing-pull-requests

Code review - solutions

Often data analysis scripts instead of software development

Pragmatic approach: use comments in the code

```
#J maybe add some comments what you exclude here (you won't know this just
#J from the column index alone)
r_cvs_rcor_refrpc <- rcor_reffree_rpc_cvs_itu$r
r_cvs_rcor_refrpc <- r_cvs_rcor_refrpc[-c(1:5), -c(6:11)]
r_cvs_rcor_refrpc <- round(r_cvs_rcor_refrpc, 1)
```

Code review - solutions

Code review is time-consuming

Short weekly meetings instead of one long review

Code review - solutions

Code review is not considered an important contribution

Increase internal visibility

Available for Code Review

Name	Language	Mail	last Review	last reviewed
Jonas	R	jonas_hagenberg@psych.mpg.de	08/2021	-
Linda	R, (Python)	linda_dieckmann@psych.mpg.de	-	08/2021

CodeClub outcomes

- More exchange
- Identification of common problems
- Spreading of knowledge and ‘good practises’

Thank you

Linda Dieckmann

Darina Czamara

Janine Knauer-Arloth

All participants of the CodeClub

campusSource

References

- Anusha M Vable, Scott F Diehl, M Maria Glymour, Code Review as a Simple Trick to Enhance Reproducibility, Accelerate Learning, and Improve the Quality of Your Team's Research, *American Journal of Epidemiology*, Volume 190, Issue 10, October 2021, Pages 2172–2177, https://doi.org/10.1093/aje/kwab092
- Baker M. 1,500 scientists lift the lid on reproducibility. *Nature* 533, 452–454 (2016). https://doi.org/10.1038/533452a
- Hagan AK, Lesniak NA, Balunas MJ, Bishop L, Close WL, Doherty MD, Elmore AG, Flynn KJ, Hannigan GD, Koumpouras CC, Jenior ML, Kozik AJ, McBride K, Rifkin SB, Stough JMA, Sovacool KL, Sze MA, Tomkovich S, Topcuoglu BD, Schloss PD. Ten simple rules to increase computational skills among biologists with Code Clubs. PLoS Comput Biol. 2020 Aug 27;16(8):e1008119. doi: 10.1371/journal.pcbi.1008119. PMID: 32853198; PMCID: PMC7451508.
- Supporting computational reproducibility through code review. Nat Hum Behav. 2021 Aug;5(8):965-966. doi: 10.1038/s41562-021-01190-w. PMID: 34408298.