REMODEL automated report
A Community Poll on Reproducibility in Computational Geosciences

Robert Reinecke

May 23, 2021

Report last updated: 19th May, 2021
Compilation date: May 23, 2021

This document summerizes a full analysis for detailed results. **It is not a full publication but rather an automated representation of the data present in this repository**
It summarizes the extensive results and builds the foundation for the publication of a journal paper. It will also be distributed along with the data in the course of a journal and data publication.

1 Aim of this survey

Software development has become an integral part of the geosciences[1] as models and data processing get more sophisticated. Paradoxically, it poses a threat to scientific progress as the pillar of science, reproducibility, is seldomly reached[2]. Software code tends to be either poorly written and documented or not shared at all; proper software licenses are rarely attributed. This is especially worrisome as scientific results have potential controversial implications for stakeholders and policymakers and may influence the public opinion for a long time[3].
In recent years, progress towards open science has led to more publishers demanding access to data and source code alongside peer-reviewed manuscripts4,5. Still, recent studies find that results can rarely be reproduced [6,7].
In this project, we conduct a poll among the geoscience community which is advertised via scientific blogs (AGU, EGU), research networks (researchgate.net and mailing lists), and social media. Therein, we strive to investigate the causes for that lack of reproducibility. We take a peek behind the curtain and unveil how the community develops and maintains complex code and what that entails for reproducibility[8] . Our survey includes background knowledge, community opinion, and behaviour practices regarding reproducible software development. We postulate that this lack of reproducibility[9] might be rooted in insufficient reward within the scientific community, insecurity regarding proper licencing of

software and other parts of the research compendium as well as scientists' unawareness about how to make software available in a way that allows for proper attribution of their work. We question putative causes such as unclear guidelines of research institutions or that software has been developed over decades[10], by researchers' cohorts without a proper software engineering process[1] and transparent licensing.
To this end, we also summarize solutions like the adaption of modern project management methods from the computer engineering community[11] that will eventually reduce costs while increasing the reproducibility of scientific research[8].

1 A comment to "Most Computational Hydrology is not Reproducible, so is it Really Science?" R.W. Hut, N.C. van de Giesen, N. Drost, Water Resources Research, 2017
2 Hutton, C., Wagener, T., Freer, J., Han, D., Du_y, C., and Arheimer, B., Most computational hydrology is not reproducible, so is it really science? Water Resources Research, 2016
3 Munafò, M., Nosek, B., Bishop, D. et al., A manifesto for reproducible science. Nat Hum Behav, 2017
4 Executive editors, G. Editorial: The publication of geoscientifc model developments v1.2. Geoscientifc Model Development, 2019
5 Katz, D. S., Niemeyer, K. E., and Smith, A. M., Publish your software: Introducing the journal of open source software (joss), Computing in Science Engineering, 2018
6 Stagge, J. H., Rosenberg, D. E., Abdallah, A. M., Akbar, H., Attallah, N. A., and James, R., Assessing data availability and research reproducibility in hydrology and water resources. Scientific data, 2019
7 Añel, J. A., García-Rodríguez, M., and Rodeiro, J.: Current status on the need for improved accessibility to climate models code, Geosci. Model Dev., 2021
8 Stodden, V., The reproducible research standard: Reducing legal barriers to scientific knowledge and innovation. IEEE Computing in Science & Engineering, 2009
9 https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970
10 Muller, C., Schaphoff, S., von Bloh, W., Thonicke, K., and Gerten, D., Going open-source with a model dinosaur and establishing model evaluation standards. EGU, 2018
11 https://software.rajivprab.com/2019/11/25/the-birth-of-legacy-software-how-change-aversion-feeds-on-itself

Our larger questions:

- Is eproducibility is an issue in the geosciences? Are bad code and documentation the root cause of that issue?
- Is model software too complex? Does that hinder reproducibility?

- Are researchers missing the tools and know-how (methods, licenses etc.) to build good model code?
- Is missing funding and missing time preventing researchers from making their models more accessible?

We define reproducibility as:
"Reproducibility in the context of modeling in the geosciences means that results obtained by a modeling experiment should be achieved again with a high degree of agreement when the study is replicated with the same model design, inputs, and general methodology by different researchers.
We explicitly exclude the retracing of results by means of using a different modeling environment (including variations in model concept, algorithms, input data or methodology)."

2 Data processing

We designed the survey according to standards from psychology research. We apply descriptive statistics to analyse demographic background and basic analysis. Further, we apply inferential statistical methods to test the unerlying hypotheses.
The raw data of the survey is stored in the folder **LiveData**. **The raw data has not been modified or cleaned in any way.** To run some basic cleanup run the follwoing script:

```python
import process_data.py as p
p.process()
```

3 Results

All data processing and plotting (including building this document) can be executed by running `python run.py`. Plotting details and addtional processing can be found in th script `plot_all.py`.
Our main hypothesis for this analysis where the following:

- **H1** Young scientists develop software more actively than established researchers.
- **H2** Young scientists are more familiar with software licenses than established researchers.
- **H3** Young scientists are more familiar with modern development methods than established researchers.
- **H4** Senior researcher percieve reproducability as a lesser problem than early career researchers.
- **H5** Software complexity is the main reason for a lack of reproducability.

- **H6** Researchers code frequently but without knowledge about proper engineering methods, licences and tools.
- **H7** The most frequently used language is still C/Fortran. Younger scientists tend to use Python and R; this is consistent throughout fields
- **H8** Most researchers are autodidacts when it comes to coding.
- **H9** Most researchers have never reproduced code with the original model. Only with their own model. This differs between fields.
- **H10** Practitioners and researchers perceive the issue of reproducibility differently. Scientists are more aware (?).
- **H11** There are more researchers that apply software than they are ones that develop it. This differs between activities and fields.
- **H12** Most researcher do not know if their software belongs to them.
- **H13** Models that are available are hard to use. Causes?: Bad code, no documentation, no input data
- **H14** Senior researchers are convinced their work is reproducible. (much more at least than young scientists)
- **H15** The smaller the scale the more reproducible and accessible.
- **H16** Researchers think that their software is bug free and always correspond to their intended implementation.
- **H17** Most senior scientists think investment in FOSS doesn't pay off.
- **H18** Senior researchers think that their code/project is easier to understand but that conflicts with reality. And younger researchers have the opposite understanding.
- **H19** We need more funding to enable reproducible computational science.
- **H20** New research software tends to be FOSS, big players are often legacy software grown over decades which is hard to reimplement as FOSS.

3.1 Sample Characteristics (demographics)

Who were our participants? Here we present characteristics of the participants in our poll, i.e. their current career stage, their years of research experience, their geo-scientific field and scale as well as their current focus of work. This is purely descriptive statistics. As we welcomed everyone to our poll, we did not form any assumptions regarding sample characteristics. Also, we tried, but might not have reached a representative sample of the population of geo-scientists.
Here, we report basic sample characteristics. Also, we can check for and report any salient sample properties.
Corresponding survey questions:

- DM01 - What career stage are you in?
- DM02 - For how long have you been working in your research field?
- DM06 - To which field within the geosciences does your research mainly belong?
- DM05 - What geographic scale are you working on?
- DM07 - What is the focus of your work?

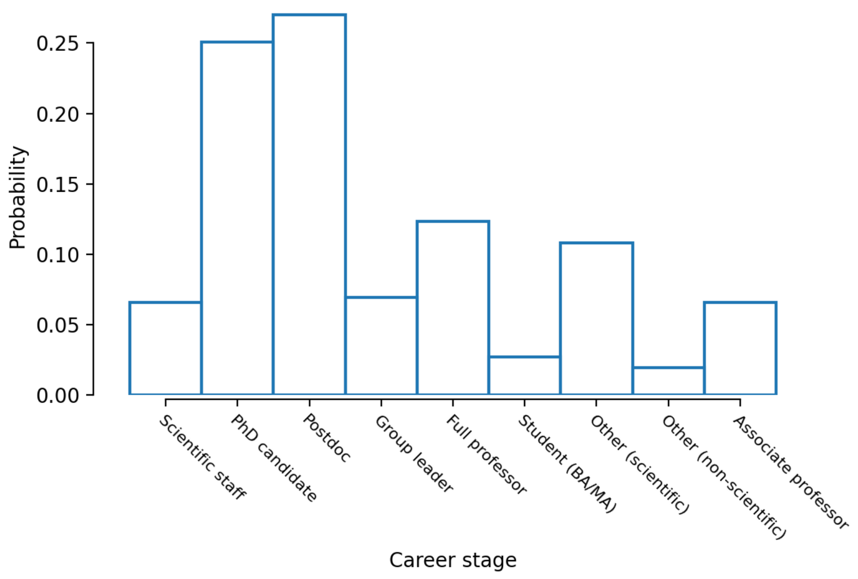

Figure 1: DM01 - Career stage of participants

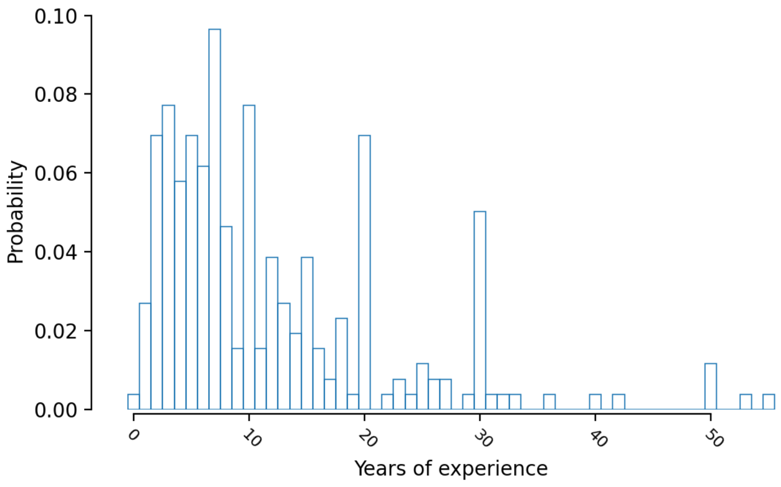

Figure 2: DM02 - For how long have you been working in your field?

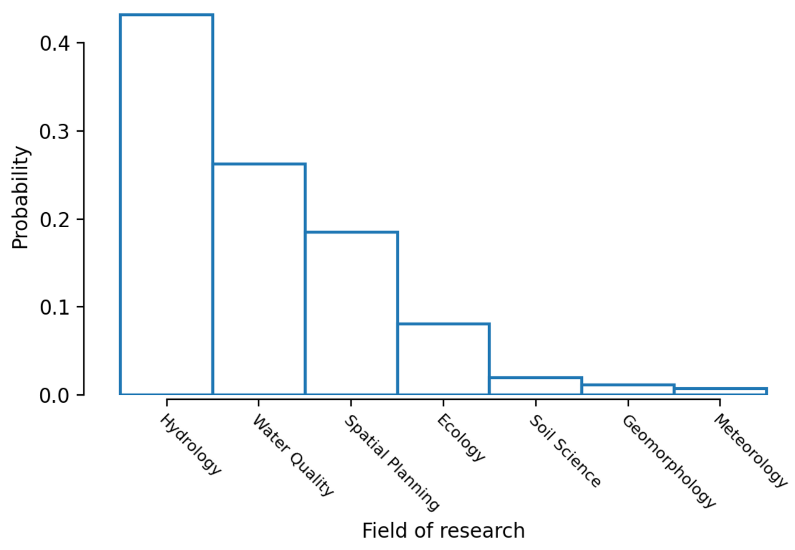

Figure 3: DM06 - Which field do you belong to?

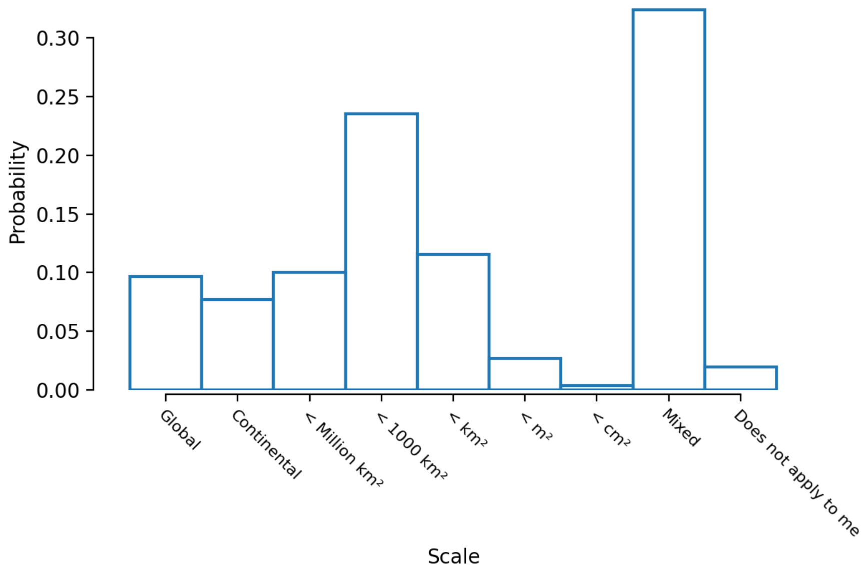

Figure 4: DM05 - What scale are you working on?

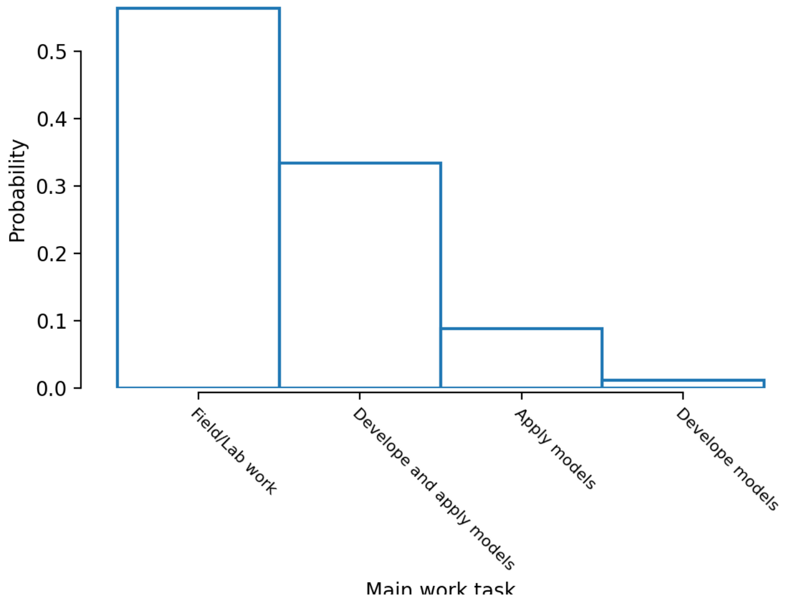

Figure 5: DM07 What is the focus of your work?

3.2 Community Opinion on Reproducibility

This part covers people's view and opinion. We assume that our defintion of reproducibility was used (annotated at several points in the poll).
Analysis Steps:

- plot corresponding questions (across full sample)
- perform statistical testing of our above stated hypotheses (across subgroups, e.g. early career vs senior researcher)
- perform further data exploration (NOT hypothesis testing) - this might inspire future research, e.g. correlation analysis

Corresponding survey questions:

- O101 - How strongly do you agree with the following statements?
- O103 - What are the reasons for a lack of reproducibility?
- S113 - How long do you think does it take for an average PhD student to efficiently work with your research software?

Corresponding hypotheses

- H4 Senior researcher percieve reproducability as a lesser problem than early career researchers.
- H5 Software complexity is the main reason for a lack of reproducability.
- H10 Practitioners and researchers perceive the issue of reproducibility differently. Scientists are more aware (?).
- H13 Models that are available are hard to use. Causes?: Bad code, no documentation, no input data.
- H14 Senior researchers are convinced their work is reproducible. (much more at least than young scientists).
- H16 Researchers think that their software is bug free and always correspond to their intended implementation.

O101: Opinion on Reproducibility in Geo-Sciences: How strongly did participants generally agree with statements? Do they consider it a problem at all? Do they think that their work is reproducible?

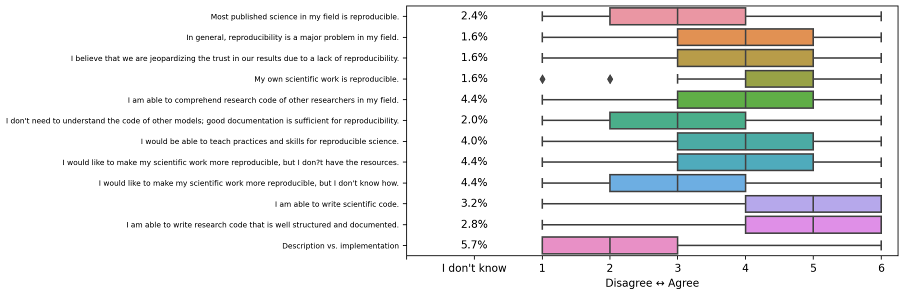

Figure 6: O101 How strongly do you agree with the following questions?

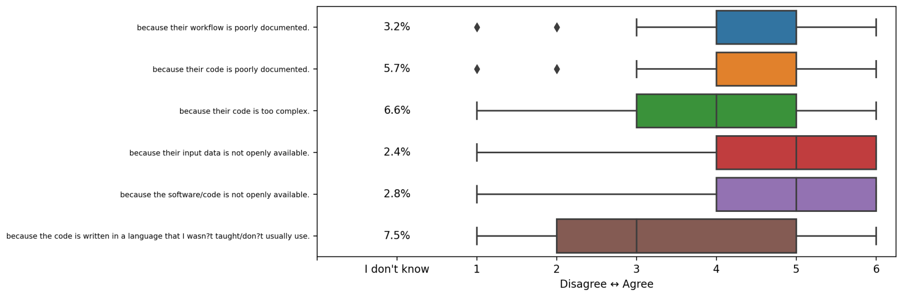

Figure 7: O103 What are the reasons for poor reproducibility?

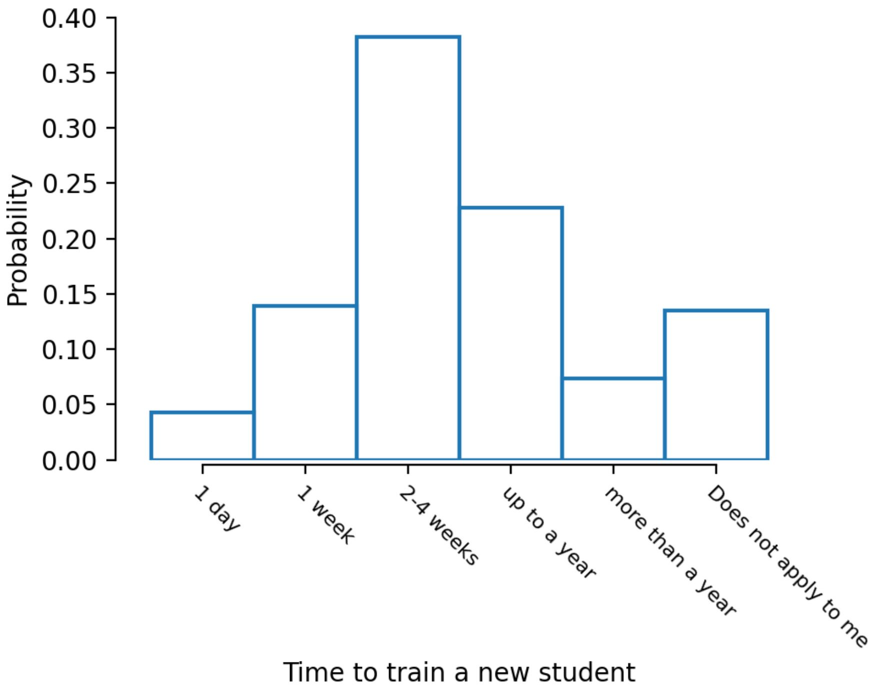

Figure 8: S113 How long does it take to train a new student in your software?

3.2.1 H4 Differences in Opinion on Reproducibility

Number of established (full Prof.) researchers: 32
Number of young (Students, PhD candidates, Postdocs, Junior Prof., Associate Prof.) researchers: 177
Mean agreement to reproducibility: 3.118577
Mean agreement among young sc.: 3.023121
Mean agreement among established sc.: 3.612903
Median agreement to reproducibility: 3.000000
Median agreement among young sc.: 3.000000
Median agreement among established sc.: 4.000000 Career has no normal distribution
No statistical test possible, samples of to low to do a proper Wilcoxon or even t-test

3.2.2 H5 Reasons for lack of reproducibility

All answers (without don't know) - Mean workflow: 4.390438, code documentation : 4.440816, code complexity: 3.884774, input data not available: 4.501976, code availability: 4.500000, written language: 3.473029
Thus, the main reason for a lack of reproducibility is: Input data, then 2: code availability, 3: documentation, 4: workflow, 5: code complexity, 6: language
Mean agreement code complexity 3.884774
Mean agreement among young sc.: 3.784431
Mean agreement among established sc.: 4.033333
Median agreement: 4.000000
Median agreement among young sc.: 4.000000
Median agreement established sc: 4.000000

3.2.3 H10 Practitioners are less awarte of the issue

No Data to investiagte this question. Only scientists answered the poll.

3.2.4 H13 Reasons for reproducibility

See H5. No addtional data to explore this further.

3.2.5 H14 Senior researchers are convinced their work is reproducible. (much more at least than young scientists).

Mean agreement on: my own research is reproducible: 4.431373
Mean agreement among young sc.: 4.417143
Mean agreement among estbalished sc: 4.562500
Median agreement: 5.000000
Median agreement among young sc: 5.000000
Median agreement among established sc.: 5.000000

3.2.6 H16 Researchers think that their software is bug free and always correspond to their intended implementation.

We did not ask a question that perfectly relates. Parts are answered in Fig. 6. Main relation to question: Implementing an algorithm based on a description from a publication yourself is the same as using the exact software package/original code that was used in that very publication.
Mean agreement - description is same as implementation: 2.379592

3.3 Reproducibility Practices and Skills

This part covers "actual" behaviour. (still only self-report assessment, but we can't change that)
Start here with summary of our hypotheses
We expected to see that... (formulate in a neutral tone)
Analysis Steps: plot corresponding questions (across full sample) perform statistical testing of our above stated hypotheses (across subgroups, e.g. early career vs senior researcher) perform further data exploration (NOT hypothesis testing) - this might inspire future research, e.g. correlation analysis
Corresponding hypotheses

- H6 Researchers code frequently but without knowledge about engeneering methods licences and tools.
- H9 Most researchers have never reproduced code with the original model. Only with their own model. This differs between fields.
- H12 Most researchers don't know if their software belongs to them.
- H2 Young scientists are more familiar with licencing issues.

Corresponding survey questions:

- O102 Did you actively reproduce scientific results in the past?
- S103 How often do you use research software?
- S110 How often do you develope research software?
- S202 Do you own your software?
- S112 Which licences are familiar?
- S101 What kind of programming languages are used?
- S111 Which licences do you use?
- S104 Do you practice any of the following methods?
- S105 Do you these tools?
- S106 Gow did you learn to program?

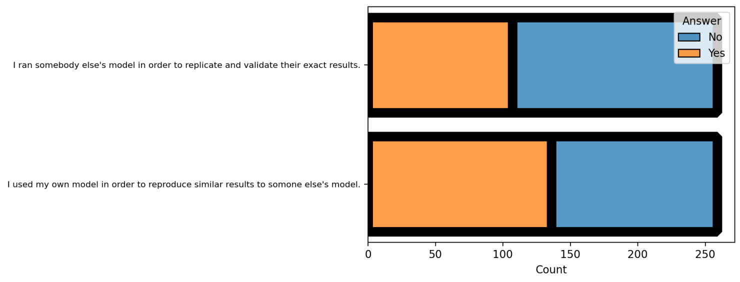

Figure 9: O102 Did you reproduce results?

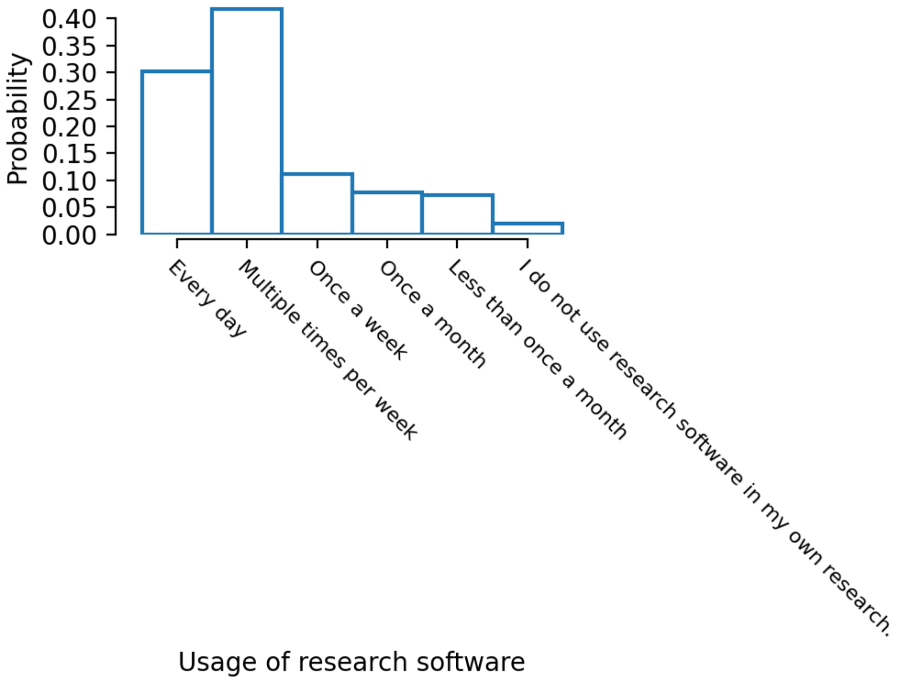

Figure 10: S103 Frequence of using software.

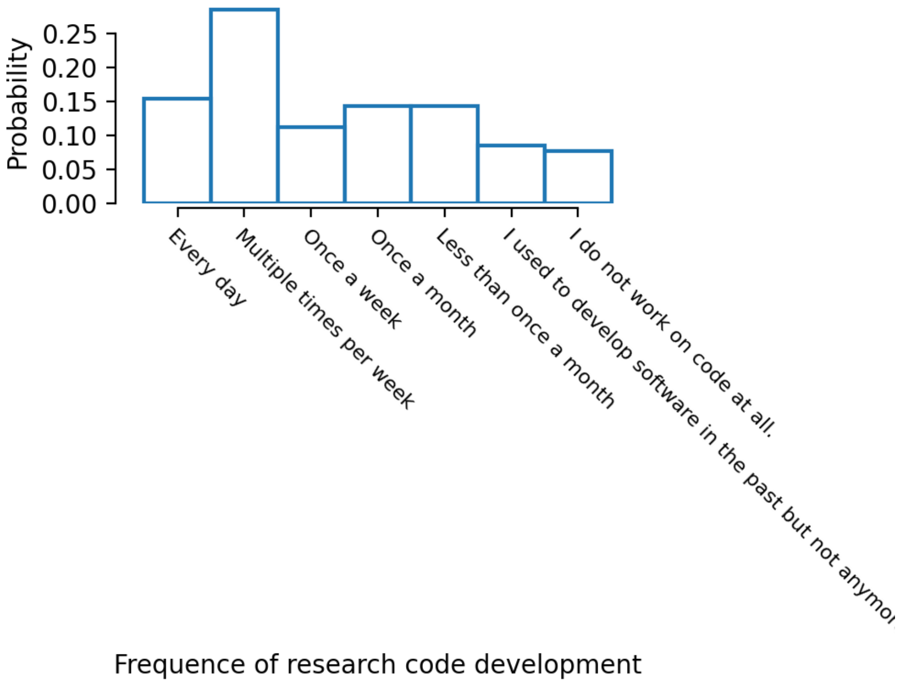

Figure 11: S110 Frequence of developing software

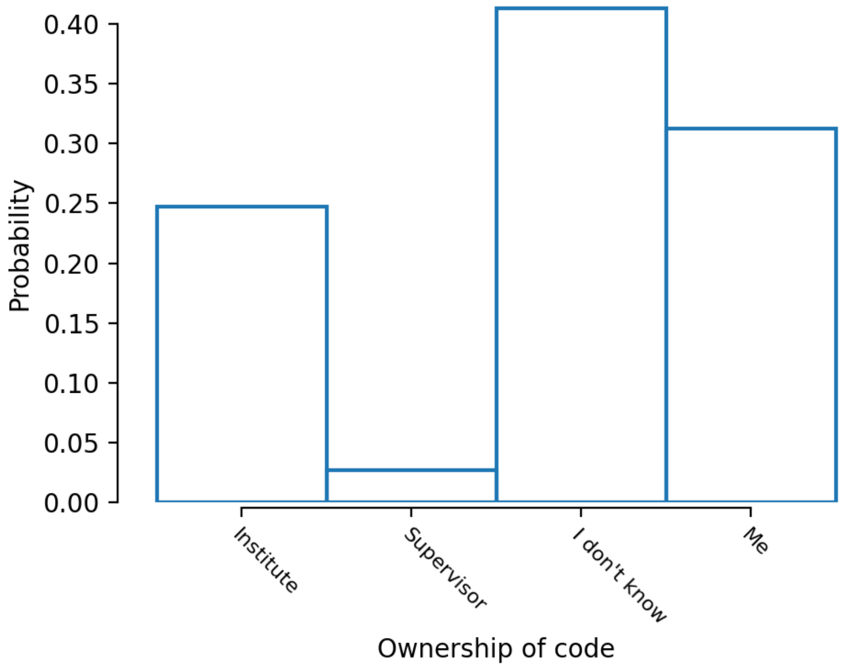

Figure 12: S202 Do you own your software?

3.3.1 H6 Knowledge

S111
What software licences do you use? Results: [None 'none' 'hfvjhgjh' 'GPL' 'MIT' 'Creative Commons' 'CECILL-C Licence (a French equivalent to the L-GPL licence)' 'Creative Commons licenses' 'None' 'MIT, LGPL' 'MIT / GPL3' 'GPLv3' 'Creative Common' 'GNU' 'Apache License v2' 'GPL MIT' 'GNU GPL v3' 'MIT license' 'CC BY-SA' 'Apache License' 'BSD' 'mit' 'GPL, PD, WTFPL' 'MATLAB' 'GNU General Public License (GPL)' 'Gpl v3' 'GPL, MIT' 'CC' 'mit, ah hoc' '???' 'GPL, MIT, Apache' 'GNU General Public License (GPL), BSD 3-Clause' 'Apache 2.0' 'US Government work - Public Domain' 'NONE' 'BSD 3-Clause' 'GPL3' 'Apache' 'FreeBSD' 'SWMM' 'BSD, GNU' 'GPL or CC-4.0 non-commercial' 'BSD 3-Clause, MIT' 'GNU 3, MIT' 'MIT, GPL' 'GNU GPL' 'educational' 'MIT, GNUv3, Creative Commons' 'GNU General Public License' 'MIT, GPL, or commercial (depending on use)' 'MatLab' 'BSD3, GPL' 'GPL v3' 'NOne' 'BSD-2-Clause License' 'GNU LGPL-2.1 License' 'MIT, BSD']
S112
Which of the licences are you familar with?
Other licences mentioned: [None 'CC0, Unlicense, WTFPL' 'LGPL' 'PD, WTFPL' 'None' 'Creative Commons' 'ArcGIS, Django' "ESRI's ArcGIS" 'MatLab' 'MIT']
Alternative answers. I dont know them at all: 49, I have heard of them but have no idea what they stand for: 69 Number of times licences were recongnized:
Apache License 33
BSD 3-Clause 24
FreeBSD 17
GNU General Public License (GPL) 119
MIT license 72
Mozilla Public License 23
Common Development and Distribution License 34
Other 9
S101
What kind of programming languages do you mainly use?
Other languages mentioned: ['Matlab', 'MATLAB', 'NCL', 'GAMS', 'matlab', 'ECMAScript', 'bash', 'Java', 'MatLab', 'Linux bash', 'shell', 'MPI', 'Stella', 'OpenFOAM', 'Delphi', 'PERL, MatLab, Pascal', 'Bash', 'Legacy (meta) languages in MS Office and ESRI: VBcode, AML (1981-1999), model builder. TBX ', 'Matlab and if possible I try to provided also compiled versione for complete reproducibility', 'whatever the students want to learn and /or makes sense for a particular project', 'Basic', 'Matlab, Julia', 'Octave/Matlab', 'Julia', 'Matlab, IDL']
Number of times languages were selected:
Fortran 51
C/C++ 64
Python 146
R 118
Others 72

None of the above 13
S104
What methods are you applying?
Others mentioned: [None 'Procedural programming' 'UML' 'Continuous Integration' 'Validation by comparison of numercial results with analytical solution or code-to-code intercomparison' 'None' 'Yelling at machine' 'git reviews with other researchers on the same project' 'continuous-integration' 'Maybe?' 'team code review' 'Open-source development (issues / pull-requests on GitHub, GitLab, etc.)']
Number of times methods were selected:
Pair programming 14
Scrum 20
Test-driven development 43
Object-oriented programming 108
Functional programming 67
Software Design Patterns (such as Abstract factory or Decorator or similar) 19
Automated Testing 47
Other 12
None of the above: 104
S105
What tools are you using?
Others mentioned: [None 'Pycharm' 'Automatic formatting; issue tracker' 'ReadTheDocs/TravisCi/PyCharm/GitHub/' 'Doxygen ' 'Code quality Checkers' 'Markdown' 'Continuous Integration tools: TravisCI, CircleCI' 'Cloud drive storage such as Box to maintain an archive of previous versions' 'No' 'jupyter hubs to make sure all members of the team work on the same distribution' 'Autoformatters, Pre-commit hooks, Test Coverage, ' 'Jupyter notebook' 'auto-formatters, linters, test frameworks, continuous integration tools' 'Continuous integration']
Number of times tools were selected:
Version control such as Git, SVC or similar 156
IDEs such as Eclipse or similar 85
Automated documentation such as Docstrings or similar 52
Other tools 14
None of the above 84
S203
What keeps you from publishing as Open Source?
Other reasons mentioned: [None "I don't write code" "I publish my code in out institutions library network, which can be freely accessed, but I'm not sure if this qualifies per se open source." 'The model I am developing is not mine and the owners do not want to make it open-source. ' 'Code is unique for a certain data set. ' 'code not ready - still in development phase' 'I try to publish most, but often time is lacking to bring it into a shape that it can be used by anyone (all bugs checked, properly documented)' 'if possible in terms of time, codes have been published' 'Too shy, worried about code quality' "Haven't published any yet" 'I don?t know if it would be useful' "Haven't come to completion of PhD projects yet, but I plant to make everything open source" 'not good

enough to publish' 'Supervisor opposition ' 'The exceutable files are available on the web as a package that is regularly updated' 'most of software has been developped during non-paid time and/or for personal consulting purposes. Some concern on giving away "for free" all of this immense invested non-paid time. Further, my code is relatively simple.' 'LPJ-GUESS model that we frequenctly use is not openly licensed by the main development group in Lund.' 'sensitive data/privacy and legal issues' 'I have not spent a lot of time thinking about the subject, but I do publish my code on github' 'Priority. I publish open data... I publish maps... I include scripts/code in some datasets... For others, it is so legacy format, that I archive it privately. ' 'not sure how and what are the standards' 'No specific reason']
Hurdles that were selected:
Licence (too complex to understand which to pick or restricted by university/institution) 42
Complexity (code too complex, not enough documentation) 40
Competition (fear to lose lead on other groups) 24
Technical ressources (e.g. storage) 21
Time 30
Other 38
Funding 84
Staff resources 21
I publish all my code as open source: 74
I don't want to: 5
Does not apply to me: 41
S106
Where did you learn to programm?
Trainings that were selected:
Computer Science degree 7
Courses during undergraduate studies (e.g., BA / Bsc) 102
Courses during postgraduate studies (e.g., MA / MSc / PhD) 95
Workshops 59
Online-courses 62
Self-taught /autodidact 212
I'm not able to write my own code: 9

3.3.2 H9 Activity of reproduction

XX

3.4 Hurdles against and Solutions towards Reproducibility

This part covers "actual" behaviour. (still only self-report assessment, but we can't change that)
Start here with summary of our hypotheses
We stated that... (formulate in a neutral tone)

Analysis Steps: plot corresponding questions (across full sample) perform statistical testing of our above stated hypotheses (across subgroups, e.g. early career vs senior researcher) perform further data exploration (NOT hypothesis testing) - this might inspire future research, e.g. correlation analysis
corresponding hypotheses (see Robert's evaluation plan): H7 H13 H3
Corresponding survey questions: S203 - What keeps you from publishing as open source? S201 - What would help to increase reproducibility? S204 (open text)

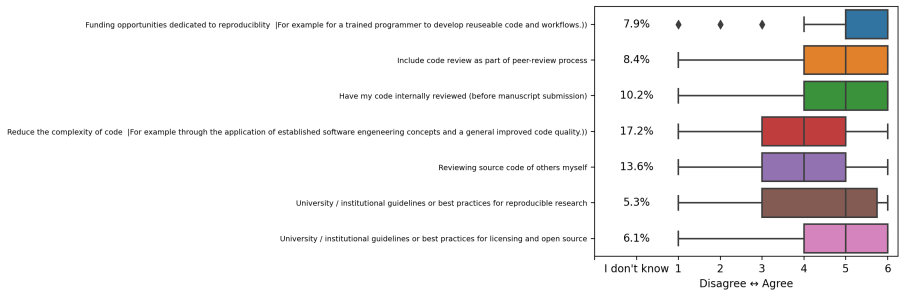

Figure 13: S201

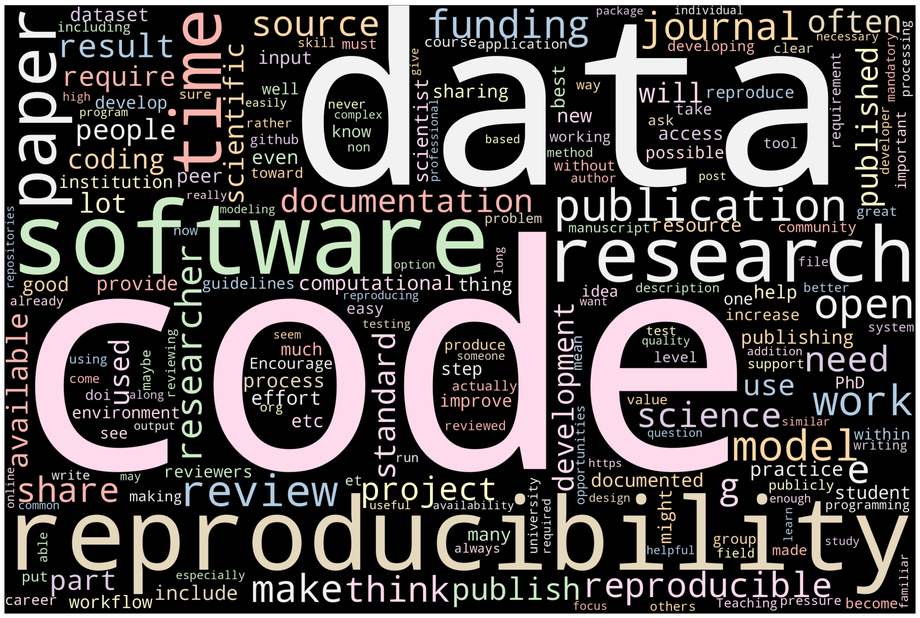

Figure 14: Full text answers as word cloud