

extended average publication times by more than seven weeks, requiring (on average) two resubmissions of author materials and eight hours of employee time per manuscript “to replicate the analyses and curate the materials for public release”, according to a 2017 analysis published in *Inside Higher Ed* (see go.nature.com/2vhv3x). Thu-Mai Christian, a data archivist at the H. W. Odum Institute for Research in Social Science at the University of North Carolina at Chapel Hill, which performs the journal’s reproducibility assessments, says the biggest issue with submitted material is poor documentation.

All articles published in Nature Research journals, and in *Nature* itself, require a statement indicating where the data can be found, and those with custom code require a code-availability statement. Code is reviewed on a case-by-case basis. At the moment, reviewers must install the software themselves to test it, and hiccups are not infrequent, says *Nature Methods* chief editor Natalie de Souza. “It’s not 1–5%” of cases, she says. “It is more common.”

TOOLKIT TEST

Reproducibility advocates are converging around a tool set to minimize these problems. The list includes version control, scripting, computational notebooks and containerization — tools that allow researchers to document their data, the

steps they follow to manipulate it, and the computing environment in which they work (see ‘Getting reproducible’).

Without these pieces, says Ben Marwick, an archaeologist at the University of Wollongong in Australia, tantalizing connections lie abandoned because they’re impossible to explore. “It’s like we’re a roomful of hungry people handing around tins of canned food, and nobody has a can opener. And then we’re asking each other, why doesn’t anyone eat anything?”

Reproducibility can be a tough sell. Researchers aren’t software engineers, and learning to work that way is an intimidating prospect, says Lowndes. Reproducibility work can also be time-consuming, and there’s little incentive to pursue it, adds Rich FitzJohn, a research software engineer at Imperial College London. “If researchers are spending all of their time stressing about how long something is going to stay working for them, that’s just time they’re not spending doing the thing that they’re good at, which is solving science problems.”

Academia in many ways disincentivizes reproducibility, as most institutions reward high-profile publications, and making code and data freely available could expose researchers to criticism and the possibility of getting scooped, Stodden says. Speaking of how much academics can realistically achieve, she adds: “Whatever people feel like they can fit in their regular day-to-day job of

being a researcher, I think that’s enough for right now”. Even if researchers cannot build executable containers, for instance, making code available on GitHub or Zenodo repositories allows others to see what they did. (Stodden, Barba and Donoho are members of a US National Academies panel on reproducibility and replicability in science. The panel’s final report is anticipated by early 2019.)

Such approaches might be most useful for researchers themselves. Computational ecologist Christie Bahlai ran a course at Michigan State University in East Lansing that taught graduate students in ecology to work with data. One year, her students submitted a paper describing a model of firefly activity, including their code. A reviewer actually tested the code and pointed out that the team was basing its conclusions on an assumption; if they changed that assumption, they might draw a different conclusion.

“He was able to make this insightful comment because of the higher level of reproducibility of the paper,” says Bahlai, who is now at Kent State University in Ohio. The team expanded the article’s discussion to elaborate.

“This is how science is supposed to work, isn’t it?” she says. “Because of reproducibility, people were able to critique the work at a deeper level.” ■

Jeffrey M. Perkel is technology editor at Nature.

Q&A Harriet Alexander

Software training in Antarctica

Harriet Alexander is a postdoctoral fellow in oceanography bioinformatics at the University of California, Davis. In January, she travelled to McMurdo Station in Antarctica to take part in this year’s Antarctic Biology Training Program, a month-long course sponsored by the US National Science Foundation. When bad weather delayed her flight home, she was able to host a workshop for the non-profit training organization Software Carpentry, its first in Antarctica.

What is Software Carpentry?

Software Carpentry is one branch of The Carpentries, a project based in San Francisco, California, that trains researchers in data science and software. For US\$2,500, plus travel and accommodation, volunteer instructors present a 2-day workshop at your location, on topics such as the command line, programming, and how to work with data. Lessons can be tailored to specific fields, and are constantly being improved in line with best practices.



I have led two workshops, and I really love the format. That sort of intensive, full-day immersion in the technology, with lots of hands-on material, is a really great model.

How did the workshop go?

We had about 20 students. The first day, I taught the Unix command-line introduction and it went really well. The second day was Python, which we typically teach

by introducing some packages using the Conda packaging system. Anaconda is about 300 megabytes. That doesn’t sound like much — but when you’re in Antarctica, where the Internet is very slow, it’s huge. I ended up downloading Miniconda, a stripped-down version of 35 megabytes, and it took me probably 2.5 days. But I forgot that Miniconda has to download other stuff, so there was no way it was going to work.

Bottom line: if you’re going to be teaching somewhere remote, download what you need before you get there.

How was Antarctica?

It’s a completely different world — very stark, yet absolutely gorgeous. McMurdo is like a mining town plopped down in the middle of this barren ice. So it’s like moving to a small town for a month and living in a dorm. It was a wonderful experience. ■

INTERVIEW BY JEFFREY M. PERKEL

This interview has been edited for length and clarity.