NATURE | TOOLBOX: Q&A

'Boot camps' teach scientists computing skills

The director of the Mozilla Science Lab discusses its course on scientific computing together with researchers who have taken the training.

Richard Van Noorden

03 September 2014

Since June 2013, the Mozilla Science Lab — an initiative of Mozilla, an international non-profit organization best known for creating the Firefox web browser — has enlisted hundreds of researchers as volunteers to run 'boot camps' in scientific computing. The project is called 'Software Carpentry'. *Nature* spoke to Kaitlin Thaney, director of the Mozilla Science Lab, and to two attendees of the workshops: Rebecca Perry, who studies applied physics at Harvard University in Cambridge, Massachusetts, and Brian Sadacca, a neuroscientist at the US National Institute on Drug Abuse in Baltimore, Maryland.

What is Software Carpentry?
Kaitlin Thaney: It's a two-day training 'boot camp' that teaches researchers some basic technical skills to deal with data. We introduce 'version control', so that researchers can better organize and track the progress of their work; how to work in the command line and use bash; data analysis and data-cleaning tricks using [the programming languages] Python or R; and other concepts like testing code and working with databases like SQL. We aren't teaching researchers to be coders or programmers — but the efficiencies we are talking about can shave off a day from every week of data-processing work for the rest of a researcher's career.

Visit the Toolbox hub for more articles

Software Carpentry actually started back in 1998, when my colleague Greg Wilson [then at Los Alamos National Laboratory in New Mexico] began a course to help researchers make their workflows more efficient. But that has evolved significantly into an open-source, community-based effort — in the past 14 months we have reached more than 4,000 researchers, including scientists, engineers and librarians; we have run 115 boot camps and have certified almost 200 volunteer training instructors. We work all over the world (but mainly in North America and the United Kingdom), and any individual or institution can request that we host a boot camp in their research centre.

Don't many researchers already have these skills?
Rebecca Perry: There is a huge variety of expertise. Some scientists still shy away from computing, and others may be familiar with programs like MATLAB, yet not some of the concepts good for efficiency, like using the command line and shell scripts. Like many researchers, I had been learning programming in fragments as needed for different research projects — so I signed up to a Software Carpentry boot camp for women in science and engineering last year to be confident I had a base level of knowledge that might be assumed for collaborative projects.

Brian Sadacca: I write my own software to examine how neurons encode information, but I had not been focused on [best practices in computing]. It's common among self-taught programmers to spend so much time and effort on getting the analysis going that there isn't a lot of forethought given to documenting and testing the things they build. Before the boot camp, I wasn't aware of what I didn't know. At my course (at the National Socio-Environmental Synthesis Center in Annapolis, Maryland) I was sitting next to someone that looks at carbon sequestration, and someone else dealing with forestry data — very different kinds of data from what I work with, and yet we had similar challenges.

What do scientists find most useful about the course?
Sadacca: I didn't have an appreciation of how professional version-control software allows every single change you make to your project to be saved forever. Git [software] is a very powerful tool — the programmer's equivalent of a lab notebook. It allows proper documentation and collaboration without confusion about who is working on what version.

Thaney: Yes, we normally have a big 'ah-hah' moment when it comes to learning version control and Git software. It's heavily used in software engineering, but the principle behind it applies to anyone working with shared content. GitHub [built on Git] is a repository, but allows individual groups to work almost seamlessly on the same project without having to, for example, send around 30 different copies of a document with initials appended to different titles. This skill sticks the most.

Perry: I now use Git for my personal projects, and share some of my side-projects on GitHub. That said, I don't share my research data online. I'm definitely willing to, but my raw data is quite sizeable: microscope images and movies. And there's no clear database where if I put in the effort to share it, I'd have confidence that the community would use it.

Sadacca: I use local Git repositories for my work here at the US National Institutes of Health, and for personal projects I use BitBucket, which is similar to GitHub, but importantly, hosts projects privately. GitHub is all public (in its free version) and this makes me a little anxious — I don't like putting out anything less than my best work.

Is two days of instruction enough to learn these skills?

Thaney: Not enough to get proficient. People use this course as a springboard, and also to learn about the community support that can foster further learning about best tools and practices. Our materials are already online, and we are working on building additional resources and cultivating reference materials.

What else is Software Carpentry planning?

Thaney: We are hitting challenges incentivizing volunteer instructors to continue teaching: despite how valuable teaching is to the community, it isn't recognized when it comes to promotion and tenure. So we are thinking about better certification models. And we are hoping to foster communities worldwide, along with other groups that help deliver and coordinate software training, such as the Software Sustainability Institute [headquartered in Edinburgh, UK].

And while Software Carpentry reaches practising scientists who don't yet identify with 'open research', we want to build bridges with other courses to further the adoption of open research, [which is an overarching goal for Mozilla Science Lab]. They include, for example, the Open Science Training Initiative, run by Sophie Kay at the University of Oxford, UK, or resources around entry-level data materials from the Open Knowledge Foundation [in Cambridge, UK]. If you are not already aware of open science or what it all means, some of the technical computing issues can provide a very big barrier to entry.

Nature | doi:10.1038/nature.2014.15799