

Library Carpentry: software skills training for library professionals

Jez Cope

The University Library, University of
Sheffield

James Baker

Sussex Humanities Lab, University of
Sussex

Introduction

Much time and energy is now being devoted to developing the skills of researchers in the related areas of data analysis and data management. However, less attention is currently paid to developing the data skills of librarians themselves: these skills are often brought in by recruitment in niche areas rather than considered as a wider development need for the library workforce, and are not widely recognised as important to the professional career development of librarians. We believe that building computational and data science capacity *within* academic libraries will have direct benefits for both librarians and the users we serve.

These technical skills can be of direct value across a wide range of common library activity. For example, many roles involve updating and processing metadata in various systems, which can be tedious and error-prone when done manually but lends itself to automation when the skills to do so are available. Reporting on complex, inconsistent and often messy datasets (such as usage statistics from different publishers) is another common task. Cleaning and normalising (“wrangling”) this data using manual techniques in spreadsheet software is time-consuming, but there are many tools commonly used by data scientists to make the process not only easier but more reproducible and hence more transparent.

Further, liaison and research support librarians are increasingly expected to facilitate researchers’ data needs. There is no expectation that we should become researchers ourselves (at least in the UK where librarians are not typically expected

Submitted 20th October 2016

Correspondence should be addressed to Jez Cope, The University Library, University of Sheffield, Western Bank, Sheffield S10 2TN. Email j.s.cope@sheffield.ac.uk

The 12th International Digital Curation Conference takes place on 20–23 February 2017 in Edinburgh. URL: <http://www.dcc.ac.uk/events/idcc17/>

Copyright rests with the authors. This work is released under a Creative Commons Attribution 4.0 International Licence. For details please see <http://creativecommons.org/licenses/by/4.0/>



to hold doctorates), but learning some of these skills and applying them in our own work is an excellent way to learn the language of researchers and build rapport.

What is Library Carpentry?

Library Carpentry¹ (Baker et al. 2016) is a global effort to provide training to librarians in technical areas that have traditionally been seen as the preserve of researchers, IT support and systems librarians. Established organisations such as Software Carpentry (Wilson 2006) and Data Carpentry (Teal et al. 2015) — below abbreviated SWC and DC respectively — offer introductory research software skills training with a focus on the needs and requirements of research scientists. Library Carpentry is a comparable introductory software skills training programme with a focus on the needs and requirements of library and information professionals.

Software Carpentry was developed in response to an overwhelming and unmet demand for training in good software engineering practice from physical scientists and engineers. It has gone through a number of iterations², but its current and most successful form is built on the following principles:

- “Curriculum is developed and improved in a public repository using methods borrowed from the open source software community.
- Instructors volunteer their time, while workshop host sites cover their travel and accommodation costs.
- Instructors use live coding instead of slides while learners following along on their own machines.”

Following the pilot course in London in November 2015, Baker et al. (2016) developed the following for Library Carpentry:

“Library Carpentry has four main guiding principles.

1. It is **COMMUNITY LED**. Library Carpentry has been community led and open access from the outset. The initial run at City University London established the tenor and content required to deliver software skills training to library and information professionals. Since then development and maintenance of modules has been driven by the community.
2. It is **RESPONSIVE**. Feedback mechanisms are embedded within lesson plans to ensure that lessons are responsive to community need. In turn, the technologies we teach and build our modules around were chosen to ensure as lesson materials are responsive.
3. It is **DISTRIBUTED**. As module development has become more distributed over time, a nominated individual has been assigned responsibility for maintaining each lesson. GitHub Issues and Gitter

¹ Library Carpentry homepage: <http://librarycarpentry.github.io>

² Software Carpentry history: <https://software-carpentry.org/scf/history/>

are used to log and coordinate issues and developments, further enhancing this transparent development model.

4. It seeks to create a SNOWBALL EFFECT. In order to expand the Library Carpentry community and embed the software skills within the library community, attendees are asked to pledge to pass on something they learn to someone else when 6 months. This transmission of skills through professional networks seeks to turn trainees into trainers, thereby also deepening the skill set of the community."

Skills identified as being valuable to librarians include:

Regular expressions Regular expressions (or regexes) are a powerful, concise and flexible system for searching text and extracting key parts. They are the Swiss-army knife of the software world, useful across a wide range of problems and integrated into many other tools: for example, in desktop applications such as Microsoft Word and programming languages such as Python.

Shell/command line A command-line is an alternative way of interacting with a computer by typing in commands to be run, either one at a time or in batches. Although appearing intimidating to beginners, the command-line is a powerful, flexible and efficient way to perform data processing tasks, especially when combining multiple commands into shell scripts or batch files.

Version control Tools such as `git`³ and GitHub⁴ enable the full version history of a set of files to be tracked and shared, including features to merge changes made by multiple collaborators. Version control is a foundational good practice for any kind of programming, however simple.

Databases Relational database management systems (RDBMS) lie at the heart of most computer systems in use in libraries. Usually these are accessed through a graphical user interface but understanding of the way such databases are structured and a little knowledge of the Structured Query Language (SQL) used to access them opens up many new possibilities.

Data-Cleaning OpenRefine is a powerful yet user-friendly data-cleaning tool. Features such as regular expressions and text clustering allow for flexible restructuring and error correction across hundreds of thousands of entries in minutes or even seconds.

Scripting languages Modern high-level languages such as Python, Ruby, R and JavaScript have greatly lowered the barrier to entry for new programmers. Not only are they much more forgiving of mistakes than their predecessors, they also come with huge toolboxes to help with almost any task, whether built-in or developed by members of the community.

These skills provide a set of tools of great value to the modern librarian. For example, they may help a librarian to:

- Automate common tasks, providing both a streamlined workflow and precise documentation of the process to communicate with colleagues

³ `git` version control system: <https://git-scm.org>

⁴ GitHub: <https://github.com>

- Quickly reformat, convert and fix hundreds of thousands of metadata records
- Unify inconsistent data for reporting, such as usage reports from multiple suppliers
- Create a dashboard visualising key aspects of that cleaned data that can be updated from live sources in seconds rather than hours of manual work

These benefits, while attractive, can seem daunting to the beginner. Library Carpentry therefore brings together groups of librarians with similar levels of experience who work together regularly to learn together and support each other in developing these new skills. Even if not every learner puts the skills into practice themselves a small amount of insight into how computers work and what might be possible can lead to improved communication with technical colleagues and a corresponding increase in the value placed on their expertise. In addition, the creation of communities of practice is an explicit aim of Library Carpentry training so that learners are empowered to continue exploring and learning with their colleagues. The skills described above, while being more than adequate for some, can provide a platform for others on which they can continue to build.

Development of Library Carpentry

In its initial exploratory run, Library Carpentry took the form of four three-hour sessions held at the City University London Centre for Information Science⁵ across four successive Monday evenings. These sessions – held in November 2015 and funded by the Software Sustainability Institute – attracted 59 participants from 14 institutions in London and its environs. Lessons were team-developed and team-taught, and were published under a CC-BY license⁶, building on earlier material developed for the British Library Digital Scholarship Training Programme, Software Carpentry, and the Programming Historian. They covered: regular expressions; the Unix shell; git; and OpenRefine.

The material for each of these original lessons was developed primarily by the lead instructor for that lesson, in most cases a librarian or information professional who had relatively recently learned the material themselves. The material was then reviewed and improved through collaboration with experienced users of the relevant tools. This approach led to the material being well-pitched for the audience, since the lead instructor was only a few steps ahead of the audience in their own learning and could readily identify potential pitfalls and common beginner mistakes that are often overlooked by more experienced teachers. The collaborative aspect ensured that, in spite of the relative inexperience of the instructors, the material taught was peer-reviewed and technically sound giving instructors and learners confidence.

In practical terms, the materials were developed using GitHub as an online collaborative platform. They were developed completely in the open from the start, an approach which has led to unsolicited but welcome contributions from a number

⁵ Library Carpentry workshop outline: <http://librarycarpentry.github.io/outline/>

⁶ Creative Commons Attribution license: <https://creativecommons.org/licenses/by/3.0/>

of directions. GitHub Pages⁷ provided seamless free web hosting on top of the GitHub repositories already in use for lesson development enabling us to deliver this content openly on the web at no cost.

Since then, 19 Library Carpentry workshops have been organised, training over 300 librarians in 8 countries across 4 continents:

- In Australia, starting with the University of Queensland in June 2016⁸
- In the USA, starting with a 5-day extensive course at UCSD in July 2016⁹ with an additional follow-up 2-day event at UC Berkeley in Oct 2016¹⁰
- In South Africa, starting with a workshop held at the Council for Scientific and Industrial Research, Pretoria Campus in August 2016¹¹ and was followed up with the first ever virtual LC workshop done in collaboration with OpenCon 2016 Cape Town¹²

During the Mozilla Science Lab Global Sprint (2-3 June 2016)¹³, an international team further developed module materials, added a new module on SQL (a relational database management language), assigned administrative roles required to support future development, and republished the materials using the Data Carpentry lesson template. Library Carpentry now has a distributed management and maintenance structure.

The Data Carpentry lesson template provided a consistent structure to the lessons and prompted the addition of a number of aspects which had previously been omitted due to time constraints, particularly a consistent set of learning objectives for each lesson.

The material continues to develop, with the growing community regularly discussing issues and questions in a dedicated open online discussion channel¹⁴.

In November 2016, James Baker and the international Library Carpentry team were awarded the 2016 British Library Labs Teaching/Learning Award¹⁵. As well as the recognition of a job well-done, this award also came with a small cash prize which is being used to facilitate further workshops.

⁷ GitHub Pages: <https://pages.github.com/>

⁸ University of Queensland workshop: <https://software-carpentry.org/blog/2016/06/LCworkshop.html>

⁹ UCSD: <https://ucsdlib.github.io/2016-07-18-UCSD/>

¹⁰ UC Berkley: <http://www.tim-dennis.com/2016-10-10-UCB/>

¹¹ Council for Scientific and Industrial Research, Pretoria Campus: <https://cmacdonell.github.io/2016-08-25-CSIR/>

¹² OpenCon 2016 Cape Town http://www.opencon2016.org/opencon_2016_cape_town

¹³ Mozilla Science Lab Global Sprint: <https://science.mozilla.org/programs/events/global-sprint-2016>

¹⁴ Library Carpentry gitter channel: <https://gitter.im/weaverbel/LibraryCarpentry>

¹⁵ British Library Labs Awards: <http://labs.bl.uk/British+Library+Labs+Awards>

Challenges

Scheduling

There are some interesting challenges involved in making this type of training work. First and foremost, it is a significant amount of time to spend out of the office for anyone, particularly at a time when budgets are being cut and staff are under pressure to get work done. It's well known from experience with Software and Data Carpentry that learners benefit most from such training when they are able to attend in a group of colleagues who work together regularly (see e.g. Wilson 2016). This works for two key reasons:

1. It enables learners to have relevant conversations about what they are learning in the context of issues they deal with in the course of their own work
2. It creates the seed of a community of practice within their own workplace, providing peer support to aid embedding of the new skills and further self-directed learning

Modelled after Software Carpentry, a typical Library Carpentry course requires the equivalent of two full days out of the office. This hits a sweet spot described by Wilson (2016):

“Once again, we discovered that five eight-hour days are more wearying than enlightening.”

This potentially requires a significant number of librarians from the same service to be away from their desks for a significant period. This is likely to be difficult to arrange, but we are developing several variations that help combat this.

The first workshops were run in the evenings after work, for two hours on four consecutive Mondays. This worked well in London, where the excellent public transport system allowed learners to travel easily to the venue after work and get home a few hours later. It also gave learners chance to digest and practice what they had learned each week before returning for the next lesson the following week.

For the planned workshops in Sheffield, this technique will not work so well, as many of our learners will have to travel a significant distance from other cities in the region. Instead we plan to run the course over two full, non-consecutive days. Using full days makes the most efficient use of learners' (and instructors') travel time; separating them by two weeks reduces the impact of staff being away from the office for the two full days.

Getting Buy-in

Many people, across all generations, have been unintentionally taught that computer skills are the preserve of a rare type of genius with only a chosen few able to

conquer the machine, and that those outside this lucky few attempts to meddle will cause more harm than good. In addition, it can be difficult for staff to justify to line managers the value of taking time out of day-to-day activities for training in these skills. This is likely to be exacerbated for librarians, since software skills are traditionally expected of systems librarians only, and not necessary for other colleagues.

Within the White Rose group of research libraries, we have spent significant time getting the buy-in of senior library management which we expect to smooth the way for individuals and teams making a case to attend. We are also taking the time to explain the value of this training from a variety of different perspectives, to potential learners and their immediate and senior managers. The Library Carpentry community is working hard to demystify computers and make these useful skills accessible to those who never thought they were within reach.

Finding instructors

Another challenge is that of finding instructors with the right combination of experience and expertise. For the reasons described above, we prefer to have sessions led by librarians and information professionals with the relevant skills whenever possible, with additional support provided by willing expert helpers. Both of these groups of people can be difficult to find.

This becomes easier with the realisation that neither instructors nor helpers need necessarily be far ahead of the learners in their knowledge. It is common within the existing SWC/DC community for learners on one course to progress to being helpers and then instructors on later courses, and structures and mentoring schemes are being introduced to support this progression. The core of this programme is the instructor training course, which is aimed at those with no formal training in teaching and based on a sound modern understanding of learning and pedagogy.

In many countries now, there is a small but growing base of SWC and DC instructors which can be called upon and we aim to develop a parallel community of Library Carpentry instructors. Indeed, an open instructor training course specifically aimed at librarians is being run in Portland, Oregon on May 4 & 5 as part of the csv,conf conference¹⁶. This course is a partnership between University of California Curation Centre and Software Carpentry. In the meantime, existing SWC and DC instructors often make willing and even enthusiastic Library Carpentry instructors.

Where Next?

This is only the beginning! Library Carpentry is in its infancy and only starting to grow through the hard work of a growing number of volunteers.

¹⁶ Instructor training for librarians: <https://datapub.cdlib.org/2016/12/08/announcing-instructor-training-for-librarians/>

We need:

- People willing to organise and run Library Carpentry courses
- Library Carpentry instructors, both experienced and fresh: Software or Data Carpentry instructor certification is helpful but not required at this stage
- Technical experts and librarians to contribute
- Use cases for the application software skills in libraries. Library Carpentry lessons and exercises reflect the library practice we know of. The more potential use cases you tell us about, the better Library Carpentry can reflect the diversity of modern librarianship.

If you would like to contribute, you can:

- Join in the conversation on GitHub¹⁷ or gitter¹⁸
- Organise your own Library Carpentry course using the existing material (don't forget to let us know: the community is there to support everyone)
- Contribute material by updating existing lessons or creating new ones

Bibliography

Baker, James, Caitlin Moore, Ernesto Priego, Raquel Alegre, Jez Cope, Ludi Price, Owen Stephens, Daniel van Strien, and Greg Wilson. 2016. "Library Carpentry: Software Skills Training for Library Professionals." *LIBER Quarterly* 26 (3). doi:[10.18352/lq.10176](https://doi.org/10.18352/lq.10176).

Teal, Tracy K., Karen A. Cranston, Hilmar Lapp, Ethan White, Greg Wilson, Karthik Ram, and Aleksandra Pawlik. 2015. "Data Carpentry: Workshops to Increase Data Literacy for Researchers." *International Journal of Digital Curation* 10 (1): 135–43. doi:[10.2218/ijdc.v10i1.351](https://doi.org/10.2218/ijdc.v10i1.351).

Wilson, Greg. 2006. "Software Carpentry: Getting Scientists to Write Better Code by Making Them More Productive." *Computing in Science & Engineering* 8 (6): 66–69. doi:[10.1109/MCSE.2006.122](https://doi.org/10.1109/MCSE.2006.122).

———. 2016. "Software Carpentry: Lessons Learned." *F1000Research* 3 (26). doi:[10.12688/f1000research.3-62.v2](https://doi.org/10.12688/f1000research.3-62.v2).

¹⁷ Library Carpentry issue discussion: <https://github.com/data-lessons/librarycarpentry/issues>

¹⁸ Library Carpentry gitter channel: <https://gitter.im/weaverbel/LibraryCarpentry>