18. Coding With Data in the Newsroom

Basile Simon

Abstract

Newsrooms present unique challenges to coders and technically minded journalists.

Keywords: computational journalism, programming, data cleaning, databases, data visualization

Inevitably, there is a point where data and code become companions. Perhaps when Google Sheets slows down because of the size of a data set; when Excel formulas become too arcane; or when it becomes impossible to make sense of data spanning hundreds of rows. Coding can make working with data simpler, more elegant, less repetitive and more repeatable. This does not mean that spreadsheets will be abandoned, but rather that they will become one of a number of different tools available. Data journalists often jump between techniques as they need: Scraping data with Python notebooks, throwing the result into a spreadsheet, copying it for cleaning in Refine before pasting it back again.

Different people learn different programming languages and techniques; different newsrooms produce their work in different languages, too. This partly comes from an organization's choice of "stack," the set of technologies used internally (for example, most of the data, visual and development work at *The Times* (of London) is done in R, JavaScript and React; across the pond *ProPublica* uses Ruby for many of their web apps). While it is often individuals who choose their tools, the practices and cultures of news organizations can heavily influence these choices. For example, the BBC is progressively moving its data visualization workflow to R (BBC Data Journalism team, n.d.); *The Economist* shifted their world-famous Big Mac Index from Excel-based calculations to R and a React/d3.js dashboard (González et al., 2018). There are many options and no single right answer.

The good news for those getting started is that many core concepts apply to all programming languages. Once you understand how to store data points in a list (as you would in a spreadsheet row or column) and how to do various operations in Python, doing the same thing in JavaScript, R or Ruby is a matter of learning the syntax.

For the purpose of this chapter, we can think of data journalism's coding as being subdivided into three core areas: Data work—including scraping, cleaning, statistics (work you could do in a spreadsheet); back-end work—the esoteric world of databases, servers and APIs; and front-end work—most of what happens in a web browser, including interactive data visualizations. This chapter explores how these different areas of work are shaped by several constraints that data journalists routinely face in working with code in newsrooms, including (a) time to learn, (b) working with deadlines and (c) reviewing code.

Time to Learn

One of the wonderful traits uniting the data journalism community is the appetite to learn. Whether you are a reporter keen on learning the ropes, a student looking to get a job in this field or an accomplished practitioner, there is plenty to learn. As technology evolves very quickly, and as some tools fall out of fashion while others are created by talented and generous people, there are always new things that can be done and learned. There are often successive iterations and versions of tools for a given task (e.g., libraries for obtaining data from Twitter's API). Tools often build and expand on previous ones (e.g., extensions and add-ons for the D3 data visualization library). Coding in data journalism is thus an ongoing learning process which takes time and energy, on top of an initial investment of time to learn.

One issue that comes with learning programming is the initial reduction of speed and efficiency that comes with grappling with unfamiliar concepts. Programming boot camps can get you up to speed in a matter of weeks, although they can be expensive. Workshops at conferences are shorter and cheaper, and for beginners as well as advanced users. Having time to learn on the clock, as part of your job, is a necessity. There you will face real, practical problems, and if you are lucky you will have colleagues to help you. There's a knack to finding solutions to your problems: Querying for issues over and over again and developing a certain “nose” for what is causing an issue.

This investment in time and resources can pay off: Coding opens many new possibilities and provides many rewards. One issue that remains at all stages of experience is that it is hard to estimate how long a task will take. This is challenging, because newsroom work is made of deadlines.

Working With Deadlines

Delivering on time is an essential part of the job in journalism. Coding, as reporting, can be unpredictable. Regardless of your level of experience, delays can—and invariably will—happen.

One challenge for beginners is slowdown caused by learning a new way to work. When setting off to do something new, particularly in the beginning of your learning, make sure you leave yourself enough time to be able to complete your task with a tool you know (e.g., spreadsheet). If you are just starting to learn and strapped for time, you may want to use a familiar tool and wait until you have more time to experiment.

When working on larger projects, tech companies use various methods to break projects down into tasks and sub-tasks (until the tasks are small and self-contained enough to estimate how long they will take) as well as to list and prioritize tasks by importance.

Data journalists can draw on such methods. For example, in one *The Sunday Times* project on the proportion of reported crimes that UK police forces are able to solve, we prioritized displaying numbers for the reader's local area. Once this was done and there was a bit of extra time, we did the next item on the list: A visualization comparing the reader's local area to other areas, and the national average. The project could have gone to publication at any point thanks to how we worked. This iterative workflow helps you focus and manage expectations at the same time.

Reviewing Code

Newsrooms often have systems in place to maintain standards for many of their products. A reporter doesn't simply file their story and it gets printed: It is scrutinized by both editors and sub-editors.

Software developers have their own systems to ensure quality and to avoid introducing bugs to collaborative projects. This includes “code reviews,” where one programmer submits their work and others test and review it, as well as automated code tests.

According to the 2017 Global Data Journalism Survey, 40% of responding data teams were three to five members and 30% of them counted only one or two members (Heravi, 2017). These small numbers pose a challenge to internal code reviewing practices. Data journalists thus often work on their own, either because they don't have colleagues, because there are no peer-review systems in place or because there is no one with the right skills to review their code.

Internal quality control mechanisms can therefore become a luxury that only a few data journalism teams can afford (there are no sub-editors for coding!). The cost of not having such control is potential bugs left unattended, sub-optimal performance or, worst of all, errors left unseen. These resource constraints are perhaps partly why it is important for many journalists to look for input on and collaboration around their work outside their organizations, for example from online coding communities.¹

Works Cited

- BBC Data Journalism team. (n.d.). What software do the BBC use? [Interview]. https://warwick.ac.uk/fac/cross_fac/cim/news/bbc-r-interview/
- González, M., Hensleigh, E., McLean, M., Segger, M., & Selby-Boothroyd, A. (2018, August 6). How we made the new Big Mac Index interactive. *Source*. <https://source.opennews.org/articles/how-we-made-new-big-mac-index-interactive/>
- Heravi, B. (2017, August 1). State of data journalism globally: First insights into the global data journalism survey. *Medium*. <https://medium.com/ucd-ischool/state-of-data-journalism-globally-cb2f4696ad3d>

About the Author

Basile Simon is graphics editor at Reuters Graphics and a Visiting Lecturer in Interactive Journalism at City, University of London.

¹ More on data journalism code transparency and reviewing practices can be found in chapters in this volume by Leon and Mazotte.