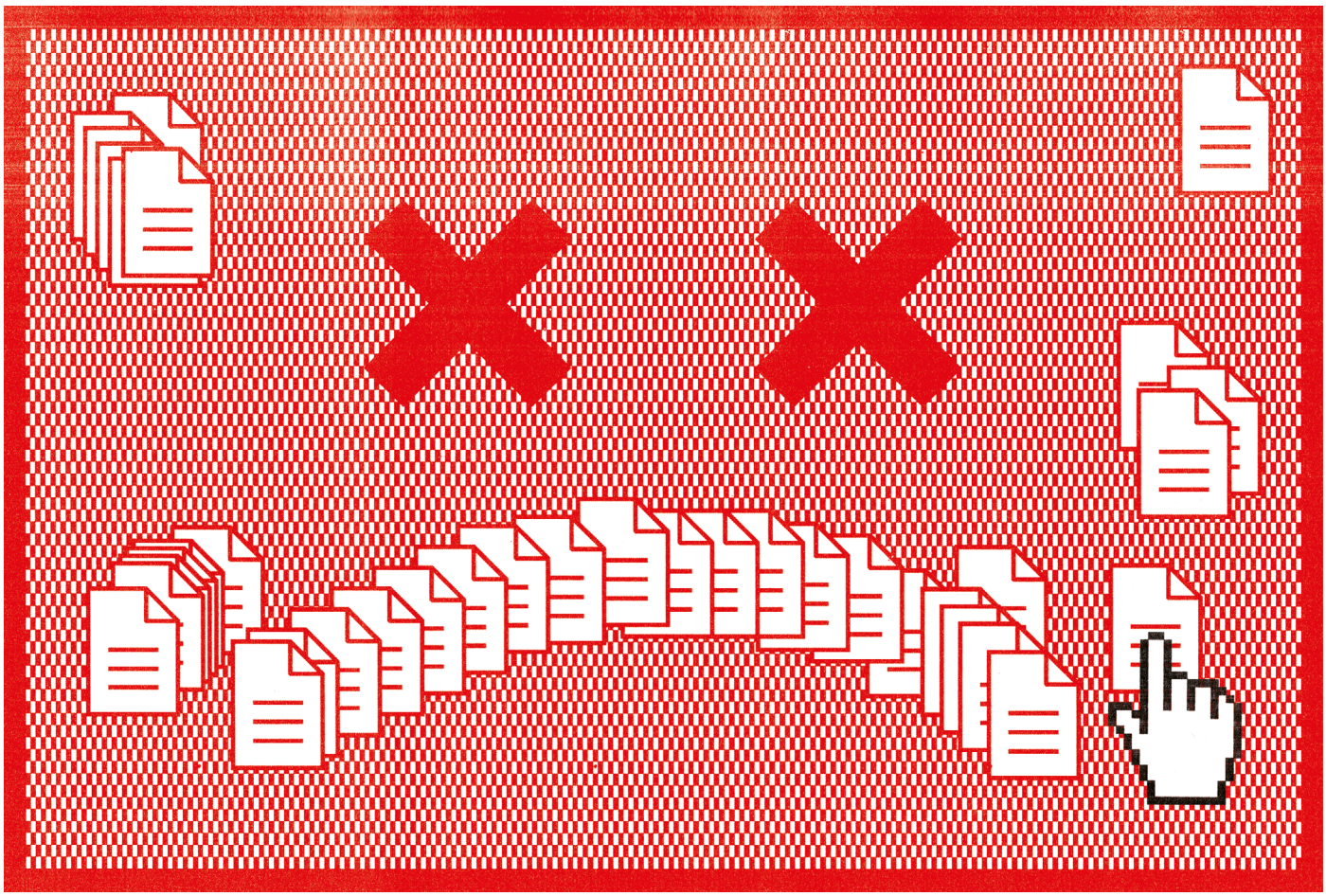

ILLUSTRATION BY THE PROJECT TWINS

AN END TO COPY-AND-PASTE ERRORS

'Executable manuscripts' allow results to be inserted directly into documents, eliminating common mistakes. **By Jeffrey M. Perkel**

If you've written a scientific manuscript, there's a good chance you're familiar with the app-switching two-step that happens when you copy your data from one program and paste them into another. That time-tested workflow does the job, but it isn't always the most efficient process. Perhaps you receive new samples and need to update your numbers. Or maybe you have to fix an error you made when processing your data. In any event, you must repeat the analysis, then comb through the manuscript line by line to find all the values that are now out of date. Oversights are inevitable.

Many tech-savvy researchers take a different path. These researchers use computational notebook systems such as R Markdown, Jupyter Book and Observable to create 'executable manuscripts', which insert data as the document is rendered, rather than copying and pasting them in. As long as the underlying data are up to date and the computations accurate, so, too, will be the final product.

Bjørn Peare Bartholdy, a bioarchaeologist at Leiden University in the Netherlands, used that approach when preparing a preprint he posted on bioRxiv last October (B. P. Bartholdy and A. G. Henry Preprint at bioRxiv https://doi.org/hf5d; 2021). As he wrote up his findings on what starch granules in dental calculus can tell us about diet, Bartholdy realized that he had made a mistake in extrapolating the final counts. "All of the numbers changed," he says. But because those values were computed in R Markdown, it took him all of two minutes to correct his work. "I don't know how much time that would have saved," he adds.

It's not the easiest way to write a paper, Bartholdy concedes. It requires computational know-how and a steep learning curve. And flexibility is needed when collaborating with less tech-savvy co-authors. But many argue that the pay-off is worth the investment. "It reduces the amount of stupid manual things that you have to do," says Sarah Pederzani, a geochemist at the Max Planck Institute for Evolutionary Anthropology in Leipzig, Germany. Bartholdy concurs: "I now work infinitely more efficiently than I did before."

Transparency

Researchers in the physical sciences and mathematics have long blended workflow engines such as Make and Snakemake with the LaTeX typesetting system to create beautifully formatted PDFs ready to post on the arXiv preprint server. But LaTeX is an unforgiving language. Today, many researchers write in Markdown, which is easier to learn, and then convert that into LaTeX and other outputs.

R Markdown, so named because it includes and can execute R code; Jupyter Book, a tool that was created to build online books from Jupyter Notebooks and text files; and Observable, a commercial JavaScript notebook system, all use Markdown to format text.

Ben Marwick, an archaeologist at the University of Washington in Seattle, has written "around a dozen" papers using R Markdown. He says that the workflow dovetails with his broader interest in open science and scientific transparency. Data science, he says, involves multiple "very small decisions" — data cleaning and filtering steps, for instance, which are crucially important, but difficult to document. And journal page limits preclude exposition. But by blending code, data and text in a single document, researchers can show just how their results were generated. "It's an extremely efficient way to communicate as much of the process as we can," Marwick says. "It makes your analyses and everything much cleaner and easier to reproduce," says Pederzani, "because you're basically making a self-contained analysis file and manuscript in one."

Version control

Executable documents, like all software code, can be posted to the platform GitHub. They can be version-controlled when the document changes, and rendered into multiple output formats. Using BibTeX, a bibliographic format supported by most citation managers, researchers can build bibliographies. And using 'styles', they can format documents to meet journal specifications. I created an example R Markdown manuscript (see go.nature.com/3jkjkt9), which can be converted to HTML, Word or PDF with a template used by Springer Nature, which publishes *Nature*. (See go.nature.com/3jgf2es for a comparable manuscript in Observable.)

Although text and code can be contained in a single file, many authors separate those elements. R Markdown, for instance, allows authors to import 'child' documents into a manuscript, which simplifies version control and collaboration, says Mine Çetinkaya-Rundel, a statistician at Duke University in Durham, North Carolina. (Our example notebook uses this approach.)

Authors can also 'cache' blocks of code that are computationally intensive, as well as import pre-built images and data rather than computing them anew with each build. Taylor Reiter, a computational biologist at the University of Colorado Anschutz Medical Campus in Aurora, compiled her PhD thesis in R Markdown by cobbling together figures she had created throughout her studies, shortening her thesis build time from about 12 minutes to 30 seconds. "These eleven-and-a-half extra minutes were key to my mental sanity during the dissertation-writing period," she jokes.

Tiffany Timbers, a statistician at the University of British Columbia in Vancouver, Canada, says that executable manuscripts provide transparency by detailing how results were generated and making it straightforward to replicate them. "You really lack this when you use something like Word or a Google Doc for writing a manuscript that involves data analysis," she says.

> **"The inline code just completely allows you to sleep well at night."**

And perhaps nowhere is that transparency clearer than when programming code is used to insert the relevant numbers into the text as the document builds — a technique known as inline execution. "In the 'compute in R and type in Word' workflow, the human in-between is responsible for making sure the latest results are reflected in the document. That's a lot of copying and pasting and keeping track of stuff," says Çetinkaya-Rundel. But with inline execution, "there's really no way to break that reproducibility, because as you update your code and you render your document, you end up with the latest results".

R Markdown, Jupyter Book and Observable all support inline code execution. Authors could, for instance, indicate the number of samples in a study by counting the rows in a table, or insert the version number of a computational package in their methods. "The inline code just completely allows you to sleep well at night," Marwick says.

Features and formats

RStudio, a development environment for R (free for academic users), includes a barebones what-you-see-is-what-you-get visual editor to ease the R Markdown writing process. A toolbar provides basic formatting options such as bold and italic, as well as the ability to insert tables and citations. Libraries such as 'Bookdown' (an R package that automatically numbers document sections, figures and tables when creating online books) and 'Rticles' (which provides article templates for Springer Nature and several other scientific publishers), enhance the experience. Observable provides a slick browser-based editing environment, whereas Jupyter Book uses a blend of browser and command-line tools.

Whatever the platform, executable manuscripts require technical skill and speciality tools. Bartholdy's paper, he notes, required several years of work. "I'm not gonna lie, it was a little painful. And it is a steep learning curve."

Mariana Montes, a linguist at the Catholic University of Leuven in Belgium, advises starting small, for instance by writing up individual experiments or analyses. "Do it for a report for yourself while you get comfortable with R Markdown, and do not start with R Markdown with your thesis — that's going to be crazy," she says.

Formatting can be particularly painful. R Markdown uses a tool called Pandoc to transform Markdown into the desired output, often through a LaTeX intermediate, and it's easy to fall foul of the LaTeX rendering engine. A misplaced backslash, for instance, can lead to "strange error messages that people have a hard time understanding", Pederzani says.

Collaboration tricks

The other main difficulty involves collaboration. Computed manuscripts are generally written in plain-text editors rather than in word processors, and collaborative writing and commenting are rarely supported. (Observable is an exception, allowing Google Docs-style collaboration.) Instead, collaborators can make comments in the form of GitHub 'pull requests' — suggested code (or text) changes that can be reviewed and incorporated into the document directly. That's how Reiter worked with one of her thesis advisers, computational biologist C. Titus Brown. But for her other, less tech-savvy adviser, she knit her thesis into a Word document and then manually folded the suggestions back into R Markdown.

As an alternative to pull requests, Timbers suggests that collaborators take advantage of GitHub's 'issues' interface, which is conventionally used to discuss bugs and suggest features. "You don't need any version-control skills to open an issue, it's like posting on a forum," she says.

Developers have created tools that can help to ease the collaborative workflow. The Trackdown package, for instance, can push and pull R Markdown files to Google Docs so that collaborators can work on them. A package called Redoc provides similar functionality for Word documents. RStudio is also developing a next-generation tool called Quarto, which helps users to build computational documents with Python, R and JavaScript through integration with Jupyter, Observable and an R package called Knitr. According to chief executive J. J. Allaire, planned improvements will ease researchers' ability to collaborate by allowing them to review Quarto manuscripts in an editor "that will kind of look and feel a lot like Google Docs".

The bottom line is that computed manuscripts can be a powerful tool for scientific writing. But they're not for everyone. Reiter found it a relatively easy way to turn text into a dissertation, but she's adept at using computational tools. "For the trade-off of not having to format my thesis, in a heartbeat I would do that again," she says. But would she advise others to use it? "Soft recommend," she laughs.

Jeffrey M. Perkel is Technology Editor at *Nature*.