Austrian Journal of Statistics
2025, Volume 54, 1–8.
http://www.ajs.or.at/
doi:10.17713/ajs.v54i3.2052

Tooling for Reproducible Research: Considerations for the Past and Future of Data Analysis

Roger D. Peng
Department of Statistics and Data Sciences
University of Texas at Austin

Abstract

The concept of reproducible research has evolved significantly over the past 30 years, with the idea growing in popularity, awareness, and acceptance. Upon its introduction to the statistical and broader scientific community, computational reproducibility was proposed as an essential concept for communicating the process of computational research and for being able to understand what exactly was done to produce a result. However, in the early stages, computational reproducibility faced at least one significant challenge, which was the lack of tools to make it easier for people to implement reproducible workflows. Fritz Leisch made major contributions to this area with his development of Sweave for the R programming language and his general promotion of software tools for reproducibility. We consider these contributions in the context of the history of reproducible research and consider what the implications are for the future of data analysis.

Keywords: reproducibility, Sweave, data analysis, R.

1. Introduction

Over the past 30 years the nature of computational research has changed significantly with respect to how it is conducted, disseminated, and extended. One of the drivers of this change has been the concept of reproducibility, or reproducible research, which calls for data and software code to be made available to others so that published claims can independently be verified and reconstructed (Peng 2011). Introduced to the statistical community in the 1990s, the concept of reproducibility involved communicating the computational details of an analysis to colleagues, collaborators, students, funders, the public, and oneself (Buckheit and Donoho 1995; Schwab, Karrenbach, and Claerbout 2000). At the time, code and data would be transmitted over CD-ROMs or other hard media, introducing considerable friction and cost to others who might want to see the details. Since the development and the expansion of the Internet, many of the technical barriers to disseminating information in general have been reduced significantly.

In any work discussing reproducibility, one must be careful to distinguish between a few related, but different, ideas, as there is often confusion regarding their meaning (Barba 2018). The concept of reproducibility we discuss here is sometimes referred to as “computational reproducibility," whereby the code and data used to conduct a data analysis can be executed together to reproduce an original published result (usually exactly). A different, and perhaps broader concept is sometimes referred to as replication, whereby independent investigators collect new data to address a similar question examined in a previous publication. With replication, a success is often characterized as obtaining a result similar to, but not exactly the same as previous work. The definition of reproducibility used in this paper is characterized in greater detail in Patil, Peng, and Leek (2019).

In the early 2000s computational reproducibility still faced two significant challenges. First, in many corners of the scientific establishment there was resistance to the idea itself (Peng, Dominici, and Zeger 2006). Researchers cited a variety of reasons including the reluctance to share proprietary data, embarrassment over poorly written code, and the potential for misuse or even abuse (Peng 2011; Keiding 2010). Second, there was a lack of software tools that would allow researchers who had accepted reproducibility as an important concept to easily incorporate it into their data analysis workflow. Across the variety of statistical software tools that people used to analyze data (including the R environment), there were few tools that would allow researchers seamlessly to improve the reproducibility of their analyses.

Over time, the idea of reproducibility slowly gained acceptance within various scientific communities. A number of high profile failures of data analysis (e.g., Herndon, Ash, and Pollin 2014; Coombes, Wang, and Baggerly 2007) kept the issue of reproducibility alive and increased awareness of its importance. One consequence was that many scientific journals, encouraged by the research community, changed their article submission policies to encourage or require the submission of data and code along with their manuscripts (Peng 2009; Stodden, Guo, and Ma 2013; Wrobel, Hector, Crawford, D'Agostino McGowan, da Silva, Goldsmith, Hicks, Kane, Lee, Mayrink, Paciorek, Usher, and Wolfson 2024). Major funding sources, such as the U.S. National Institutes of Health have expanded policies around reproducibility, for example, to encourage deposition of data and code in central repositories (NIH 2024).

Addressing the need for software tools to build reproducible data analyses was a challenge because it required developing something that would be embedded deep within an analyst's workflow. Such tools would need to encourage reproducibility without introducing too much overhead. Fritz Leisch made a major contribution to the tooling for reproducible research by introducing and developing Sweave for the R programming environment (R Core Team 2024). In addition, Fritz actively promoted and encouraged the development of other software by co-editing the book *Implementing Reproducible Research* with Victoria Stodden and myself (Stodden, Leisch, and Peng 2014). These efforts, which we will discuss further in the next section, set an example for how to constructively face a difficult problem confronting all scientific researchers and laid a foundation on which many others would make their own contributions.

2. Tools for building reproducible data analyses

Sweave drew its inspiration from the literate programming ideas originally developed by Donald Knuth (Knuth 1984). There, Knuth envisioned a system by which computer programs and their documentation could be written together in a single document using both a programming language and a documentation language (his original WEB system used Pascal as the programming language and TeX as the documentation language). With a literate program one could then *tangle* it into a compilable program or *weave* it into a human-readable document. A key benefit was that documentation and code were placed in the same document, thereby making it easier for the programmer to maintain coherence between the two.

Data analysis and programming are not the same activities and therefore a system for writing reproducible analyses would likely differ from the original literate programming vision. However, the basic ideas would carry over to the Sweave system (Leisch 2002). Sweave's original format closely resembled Norman Ramsay's noweb format (Ramsey 1994) and used the weave and tangle verbs. Documents would be written using a programming language, in this case R, and a documentation language (LaTeX). Documents would be divided into text chunks with human-readable documentation and code chunks with machine readable code. Analysts could then use the Sweave system to weave the documents into nicely formatted human-readable documents, such as PDF files, or tangle them into pure R source files. Rather than compiling the code and creating an executable, as in Knuth's original system, Sweave executed the code chunks in the R interpreter (as part of the weave-ing process) and replaced the code with the results of the execution. This difference in behavior from the original literate programming ideas reflects the different context in which Sweave was used. While the weave and tangle verbs make some sense in a data analysis context, Fritz's adapting of these ideas made them more useful for the statistical programmers that would adopt the system in their work.

Very often, the result of executing code in a Sweave document was a plot or a table or some other numerical result derived from data. This feature of Sweave, whereby the results of executed code would be automatically embedded within the document, made it particularly useful for building dynamic documents or compendia, where results could be updated and kept in sync by re-weaving the original source files (Gentleman and Temple Lang 2007). Thus, there was no need to have separate files containing analysis code and results. Everything could be combined into a single document and the analyst could choose to present as much of the code, results, and data as was considered necessary. If there were a change to the data or the code, re-weaving the source file would update the results and confirm the reproducibility of the analysis (or not, if there were some error in the code).

It is perhaps easy to overlook the novelty of the original Sweave system, which was introduced to the R community in 2002 with R version 1.5.0. In particular, Sweave was not a new programming language or even a tool to help with programming, it was not a tool (like an editor) for writing papers or reports, and it was not a statistical method for analyzing a specific kind of data. Rather, it was a tool for building a reproducible data analysis workflow. Specifically, Sweave enabled the user to fundamentally improve the reproducibility of a data analysis by allowing the presentation of the results of that analysis to be connected directly with the data and the code. Critically, Sweave's use of LaTeX and R and plain text formats (as opposed to proprietary file formats like Microsoft Word) did not require many statisticians to learn different tools or languages. Sweave therefore gave analysts the ability to improve the quality and reliability of any document that relied on the output of code and data without having to make major workflow changes.

One task that Sweave seemed particularly well-suited for from the beginning was the development of tutorials and manuals for R packages. Version 1.5.0 of R also introduced the capability for developing R package *vignettes*, which could be written in the R/nowebd format mixing LaTeX with R code chunks (Leisch 2003). These vignettes could be longer than a typical R manual page and could provide more details about how to use the functions in the package. When an R source package was built, the vignette would be woven to produce a PDF version that would be distributed along with the package code. Users could then read the vignettes in the package with the `vignette()` function upon installing the package. A key feature of these vignettes is that the users could always be assured that the code and the output were up to date (at least as of the last build) and that the code had executed, because otherwise the package could not have been built in the first place. Some time later, the Bioconductor Project mandated the inclusion of at least one vignette in each of its packages and continues to maintain that as a minimum documentation standard (Bioconductor Project 2024).

Sweave has also inspired the development of numerous related tools that modified the existing elements of the Sweave system, including the programming language, the documentation language, and the documentation output. For example, the R packages **R2HTML** (Lecoutre 2003) and **odfWeave** (Kuhn 2006) allowed for the creation of HTML and Open Document Format output, respectively. The **IDynDocs** package substituted XML for LaTeX as the documentation language (Nolan and Temple Lang 2007) and **SASweave** moved away from the R system and allowed for the integration of SAS code into documents (Lenth and Højsgaard

2007). The **cacher** (Peng 2008) and **weaver** (Falcon 2009) packages introduced object caching in order to speed up the weaving process for large documents. The widely used **knitr** system (and the later Quarto system from Posit) added Markdown as a documentation language (in addition to LaTeX) in the now popular R Markdown format (Xie 2017). The integration of **knitr** into the original RStudio interactive development environment increased Sweave's accessibility to a wide range of new developers.

Since its introduction, Sweave has been adopted far beyond the statistical community for producing reproducible data analysis reports (e.g., Garbade and Burgard 2006; Meredith and Racine 2009; Koenker and Zeileis 2009). Others have advocated for its use in collaborative workflows to ensure reproducibility and efficient documentation of statistical analyses (Baggerly and Coombes 2009). Beyond its use in research applications, Sweave has found considerable interest in teaching settings and has transformed the development of computational textbooks. One application that leverages the use of live code and dynamic computation is the development of scalable randomized exams in a classroom setting (Grün and Zeileis 2009; Gómez, Mulero, Nueda, Pascual, and Molina 2013; Zeileis, Umlauf, and Leisch 2014). For publishing, entire book series, such as Springer's *Use R!* or Chapman & Hall's *R Series*, have been established to highlight the use of R, and many of these have been written in Sweave or a descendant tool (e.g., Peng and Dominici 2008). Tools like Sweave, and later **bookdown** (Xie 2016), are well suited to books which need frequent updating due to changes to the underlying software being demonstrated (Holmes and Huber 2019).

3. Building better data analyses

Since the publication of the book *Implementing Reproducible Research* in 2014, much in the world of data analysis has changed. That book had a simple purpose, which was highlighted in the Preface:

> Assuming one agrees that reproducibility of a scientific result is a good thing, *how do we do it*?

The answer to that question, from a software perspective, is largely answered, as the proliferation of software tools in the over ten years since the book's publication demonstrates. Fritz's development of Sweave played a major role in this effort and his impact continues to this day. The general problem of reproducibility still faces challenges though, primarily regarding the need for infrastructure to support the distribution and maintenance of data and code (Peng and Hicks 2021).

A key goal of reproducible research, and all of the tooling developed to support it, is to allow the scientific community to see what was done in an analysis. But achieving this goal leads us to ask whether answering the question, "What was done in this analysis?" is enough for all of our purposes. Ultimately, a broader question would be to ask "Do I understand and trust this data analysis?" While reproducibility seems to provide the means to answer this question, it does not answer the question itself, as even a poor data analysis can be reproducible (Leek and Peng 2015).

One of the most pathological examples of the shortcomings of reproducibility is the paper of Potti, Dressman, Bild, Riedel, Chan, Sayer, Cragun, Cottrill, Kelley, Petersen, Harpole, Marks, Berchuck, Ginsburg, Febbo, Lancaster, and Nevins (2006) and the investigations that ensued to figure out what was done. Keith Baggerly and Kevin Coombes ultimately did reproduce the original investigators' work (after deliberately introducing numerous data and statistical errors) but estimate that they took over 1,000 person-hours to do so. If the paper had been fully reproducible from the beginning, it is highly likely that the number of hours dedicated to obtaining the original results would have been reduced. However, reproducibility would have only reduced the time to produce an incorrect result! Arguably, much of the time spent by Baggerly and Coombes was spent figuring out what the results *should have been*, had the original investigators done things properly. Such painstaking "forensic bioinformatics" is time-consuming and tends to go far beyond what was done in the original analysis.

Pathological data analyses aside, there has been a growing concern over the lack of replication and reproducibility of significant findings in areas such as psychology and medicine, whereby independent investigators are sometimes unable to confirm published findings, either with new datasets or with the original datasets (Patil, Peng, and Leek 2016; Ioannidis 2005). While there are likely many causes for the lack of replication of findings, one question is whether better tooling could be invoked to prevent such problems from occurring. Baggerly and Coombes directly suggested that the use of Sweave could prevent irreproducibility in some analyses (Baggerly and Coombes 2009). More broadly, the implementation of reproducible software workflows could improve the quality of data analysis overall. A number of entities, such as the Center for Open Science have been building tools to encourage reproducibility and replicability. Such efforts will likely play an important role in mitigating some of the worst outcomes. That said, it is unlikely that software tooling alone can improve all data analyses. Rather, problems may reach into issues of study design or statistical methodology that need to be addressed via other means, such as education.

Perhaps another goal of tools like Sweave could be framed as allowing for and encouraging more "frictionless reproducibility" (Donoho 2024). It is worth noting that even in ideal circumstances, when code and data are made available to others, reproducing a published result can still be challenging (Barba 2024). One could think of individual data analyses or papers much the same way we view software packages—as re-usable entities upon which we can build larger or more complex analyses or models. While much work has been done to consider formats for enabling this approach, such as caching and distributing computational results (Peng 2008), the transition from writing primarily human-readable analyses to developing machine-readable data products is ongoing (Barba 2024).

Moving beyond the minimum data analysis standard of reproducibility will require re-thinking and updating our goals for data analysis. While reproducibility, and the tools supporting it, will continue to play a key role in data analysis, ultimately we want to know that an analysis does not suffer from a significant failure. Specifically, it would be useful to know from the presentation of a data analysis that (1) there are no errors in the analysis that, if fixed, would change the results; and (2) there does not exist a plausible alternative explanation that is consistent with the data but is inconsistent with the primary claim. In order to make such assurances, we likely will need to develop data analysis representations that go beyond showing what was done to produce the final result. For example, it could be valuable to see analyses that were done but not chosen to be included in the final presentation, especially if those analyses considered alternative hypotheses. However, a comprehensive dump of every analytic twist and turn would likely be overwhelming. New tools will need to be developed that can open up the data analysis process in a compact and usable way, while also providing significant advantages to the analyst, much as Sweave did when it was first released. Such tools would further allow the community to extend and rely on published data analyses to answer new questions.

The impact of Fritz Leisch on the scientific community has been substantial and his development of tools for reproducible research has played a key role in encouraging a culture of transparency and openness in research. By providing the means to implement reproducible data analyses, he allowed advocates of reproducibility to demonstrate its value and importance in a constructive manner. Ultimately, the succession of software tools implementing his original ideas in a variety of other contexts is an important legacy that has extended his impact across time and disciplines.

References

Baggerly KA, Coombes KR (2009). "Deriving Chemosensitivity from Cell Lines: Forensic Bioinformatics and Reproducible Research in High-Throughput Biology." *The Annals of Applied Statistics*, **3**(4), 1309–1334. doi:10.1214/09-aoas291.

Barba L (2024). "The Path to Frictionless Reproducibility is Still under Construction." *Harvard Data Science Review*, **6**(1). doi:10.1162/99608f92.d73c0559.

Barba LA (2018). "Terminologies for Reproducible Research." *arXiv 1802.03311*, arXiv.org E-Print Archive. doi:10.48550/arxiv.1802.03311.

Bioconductor Project (2024). "Bioconductor Packages: Development, Maintenance, and Peer Review." Accessed 2024-09-13, URL https://contributions.bioconductor.org/docs.html.

Buckheit JB, Donoho DL (1995). "**WaveLab** and Reproducible Research." In A Antoniadis, G Oppenheim (eds.), *Wavelets in Statistics*, volume 103 of *Lecture Notes in Statistics*, pp. 55–81. Springer-Verlag, New York. doi:10.1007/978-1-4612-2544-7_5.

Coombes KR, Wang J, Baggerly KA (2007). "Microarrays: Retracing Steps." *Nature Medicine*, **13**(11), 1276–1277. doi:10.1038/nm1107-1276b.

Donoho D (2024). "Data Science at the Singularity." *Harvard Data Science Review*, **6**(1). doi:10.1162/99608f92.b91339ef.

Falcon S (2009). "Caching Code Chunks in Dynamic Documents: The **weaver** Package." *Computational Statistics*, **24**(2), 255–261. doi:10.1007/s00180-008-0125-9.

Garbade S, Burgard P (2006). "Using R/Sweave in Everyday Clinical Practice." *R News*, **6**(2), 26–31. URL https://journal.R-project.org/articles/RN-2006-012/.

Gentleman R, Temple Lang D (2007). "Statistical Analyses and Reproducible Research." *Journal of Computational and Graphical Statistics*, **16**(1), 1–23. doi:10.1198/106186007x178663.

Gómez DS, Mulero J, Nueda MJ, Pascual A, Molina MD (2013). "Random Exams Using Sweave." In *INTED2013 Proceedings*, pp. 4759–4766.

Grün B, Zeileis A (2009). "Automatic Generation of Exams in R." *Journal of Statistical Software*, **29**(10), 1–14. doi:10.18637/jss.v029.i10.

Herndon T, Ash M, Pollin R (2014). "Does High Public Debt Consistently Stifle Economic Growth? A Critique of Reinhart and Rogoff." *Cambridge Journal of Economics*, **38**(2), 257–279. doi:10.1093/cje/bet075.

Holmes S, Huber W (2019). *Modern Statistics for Modern Biology*. Cambridge University Press. ISBN 9781108705295.

Ioannidis JPA (2005). "Why Most Published Research Findings Are False." *PLOS Medicine*, **2**(8), e124. doi:10.1371/journal.pmed.0020124.

Keiding N (2010). "Reproducible Research and the Substantive Context." *Biostatistics*, **11**(3), 376–378. doi:10.1093/biostatistics/kxq033.

Knuth DE (1984). "Literate Programming." *Computer Journal*, **27**(2), 97–111. doi:10.1093/comjnl/27.2.97.

Koenker R, Zeileis A (2009). "On Reproducible Econometric Research." *Journal of Applied Econometrics*, **24**(5), 833–847. doi:10.1002/jae.1083.

Kuhn M (2006). "Sweave and the Open Document Format – The **odfWeave** Package." *R News*, **6**(4), 2–8. URL `https://journal.R-project.org/articles/RN-2006-025/`.

Lecoutre E (2003). "The **R2HTML** Package." *R News*, **3**(3), 33–36. URL `https://journal.R-project.org/articles/RN-2003-019/`.

Leek JT, Peng RD (2015). "Opinion: Reproducible Research Can Still Be Wrong: Adopting a Prevention Approach." *Proceedings of the National Academy of Sciences of the United States of America*, **112**(6), 1645–1646. `doi:10.1073/pnas.1421412111`.

Leisch F (2002). "Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), *COMPSTAT 2002 – Proceedings in Computational Statistics*, pp. 575–580. Physica Verlag, Heidelberg. `doi:10.1007/978-3-642-57489-4_89`.

Leisch F (2003). "Sweave, Part II: Package Vignettes." *R News*, **3**(2), 21–24. URL `https://journal.R-project.org/articles/RN-2003-013/`.

Lenth RV, Højsgaard S (2007). "**SASWeave**: Literate Programming Using SAS." *Journal of Statistical Software*, **19**(8), 1–20. `doi:10.18637/jss.v019.i08`.

Meredith E, Racine J (2009). "Towards Reproducible Econometric Research: The Sweave Framework." *Journal of Applied Econometrics*, **24**(2), 366–374. `doi:10.1002/jae.1030`.

NIH (2024). "Enhancing Reproducibility through Rigor and Transparency." Accessed 2024-09-09, URL `https://grants.nih.gov/policy/reproducibility/index.htm`.

Nolan D, Temple Lang D (2007). "Dynamic, Interactive Documents for Teaching Statistical Practice." *International Statistical Review*, **75**(3), 295–321. `doi:10.1111/j.1751-5823.2007.00025.x`.

Patil P, Peng RD, Leek JT (2016). "What Should Researchers Expect when They Replicate Studies? A Statistical View of Replicability in Psychological Science." *Perspectives on Psychological Science*, **11**(4), 539–544. `doi:10.1177/1745691616646366`.

Patil P, Peng RD, Leek JT (2019). "A Visual Tool for Defining Reproducibility and Replicability." *Nature Human Behaviour*, **3**(7), 650–652. `doi:10.1038/s41562-019-0629-z`.

Peng RD (2008). "Caching and Distributing Statistical Analyses in R." *Journal of Statistical Software*, **26**(7), 1–24. `doi:10.18637/jss.v026.i07`.

Peng RD (2009). "Reproducible Research and Biostatistics." *Biostatistics*, **10**(3), 405–408. `doi:10.1093/biostatistics/kxp014`.

Peng RD (2011). "Reproducible Research in Computational Science." *Science*, **334**(6060), 1226–1227. `doi:10.1126/science.1213847`.

Peng RD, Dominici F (2008). *Statistical Methods for Environmental Epidemiology in R: A Case Study in Air Pollution and Health*. Springer-Verlag. `doi:10.1007/978-0-387-78167-9`.

Peng RD, Dominici F, Zeger SL (2006). "Reproducible Epidemiologic Research." *American Journal of Epidemiology*, **163**(9), 783–789. `doi:10.1093/aje/kwj093`.

Peng RD, Hicks SC (2021). "Reproducible Research: A Retrospective." *Annual Review of Public Health*, **42**(1), 79–93. `doi:10.1146/annurev-publhealth-012420-105110`.

Potti A, Dressman HK, Bild A, Riedel RF, Chan G, Sayer R, Cragun J, Cottrill H, Kelley MJ, Petersen R, Harpole D, Marks J, Berchuck A, Ginsburg GS, Febbo P, Lancaster J, Nevins JR (2006). "Genomic Signatures to Guide the Use of Chemotherapeutics." *Nature Medicine*, **12**(11), 1294–1300. `doi:10.1038/nm1491`.

Ramsey N (1994). "Literate Programming Simplified." *IEEE Software*, **11**(5), 97–105. doi:10.1109/52.311070.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Schwab M, Karrenbach N, Claerbout J (2000). "Making Scientific Computations Reproducible." *Computing in Science & Engineering*, **2**(6), 61–67. doi:10.1109/5992.881708.

Stodden V, Guo P, Ma Z (2013). "Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals." *PLOS ONE*, **8**(6), e67111. doi:10.1371/journal.pone.0067111.

Stodden V, Leisch F, Peng RD (2014). *Implementing Reproducible Research*. Chapman and Hall/CRC. doi:10.1201/9781315373461.

Wrobel J, Hector EC, Crawford L, D'Agostino McGowan L, da Silva N, Goldsmith J, Hicks S, Kane M, Lee Y, Mayrink V, Paciorek CJ, Usher T, Wolfson J (2024). "Partnering With Authors to Enhance Reproducibility at JASA." *Journal of the American Statistical Association*, **119**(546), 795–797. doi:10.1080/01621459.2024.2340557.

Xie Y (2016). ***bookdown**: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC. doi:10.1201/9781315204963.

Xie Y (2017). *Dynamic Documents with R and **knitr***. Chapman and Hall/CRC. doi:10.1201/9781315382487.

Zeileis A, Umlauf N, Leisch F (2014). "Flexible Generation of E-Learning Exams in R: Moodle Quizzes, OLAT Assessments, and Beyond." *Journal of Statistical Software*, **58**(1), 1–36. doi:10.18637/jss.v058.i01.

Affiliation:

Roger D. Peng
Department of Statistics and Data Sciences
University of Texas at Austin
105 E 24th St D9800
Austin, Texas, 78705, USA
E-mail: roger.peng@austin.utexas.edu
URL: https://rdpeng.org

Austrian Journal of Statistics http://www.ajs.or.at/
published by the Austrian Society of Statistics http://www.osg.or.at/
Volume 54 *Submitted:* 2024-09-16
2025 *Accepted:* 2025-02-02