Proposal

Felipe Fontana Vieira Jason Geller Bruno Rodrigues

2025-08-27

Computational reproducibility remains limited in psychology, in part because reproducibility exists on a spectrum [1] — from sharing isolated code fragments to providing fully executable pipelines that ensure identical results. Methodological research often depends on specialized software (e.g., R), domain-specific packages, and system dependencies. This reliance is especially pronounced in simulation studies, which involve computationally intensive procedures and multi-step pipelines. As B. S. Siepe, F. Bartoš, T. P. Morris, A.-L. Boulesteix, D. W. Heck, and S. Pawel [2] note, simply sharing code is insufficient, as it neglects the underlying computational environment —placing such practices at the lower end of the reproducibility spectrum.

To address these challenges, several tools target different layers of the problem. For example, renv and groundhog pin package versions, targets manages pipelines, and Docker encapsulates operating systems. While each of these solutions provides value, their adoption within psychological science has been limited. A key barrier is the perception of a slow learning curve and the lack of clear guidance on how these tools integrate with one another, which in turn constrains their practical impact on reproducible research.

Nix, a package manager for "deterministic computational environments," offers a more comprehensive solution. Unlike conventional tools, Nix simultaneously manages R, R packages, and system-level dependencies (e.g., BLAS/LAPACK), ensuring even low-level libraries are reproduced. Its central repository (nixpkgs) includes over 120,000 packages, including CRAN and Bioconductor, allowing for project-specific environments that remain stable for years, independent of host updates. In principle, Nix could replace fragmented solutions like renv, targets, and Docker by unifying their functionality in a single framework.

Although Nix has its own language, the R package {rix} [3] provides an accessible interface, enabling researchers to create Nix environments directly from R. This preserves Nix's reproducibility benefits while supporting multi-language setups (e.g., R, Python, Julia), which are increasingly relevant for psychological science.

This article will demonstrate an {rix} implementation for producing fully reproducible manuscripts. We will start with the introduction of Nix and {rix}, providing information on how it is connected within the entire reproducibility spectrum and existing tools. Then, we will showcase a GitHub repository that mimics collaborative simulation studies positioned at the lower end of the reproducibility spectrum. Subsequently, we illustrate how to use {rix} to achieve a comprehensive

computational reproducibility for such methodological manuscripts, providing practical guidance for researchers.

Bibliography

[1] R. D. Peng, "Reproducible research in computational science," *Science*, vol. 334, no. 6060, pp. 1226–1227, 2011, doi: 10.1126/science.1213847.

[2] B. S. Siepe, F. Bartoš, T. P. Morris, A.-L. Boulesteix, D. W. Heck, and S. Pawel, "Simulation studies for methodological research in psychology: A standardized template for planning, preregistration, and reporting," *Psychological Methods*, 2024, doi: 10.1037/met0000695.

[3] B. Rodrigues and P. Baumann, "rix: Reproducible Data Science Environments with 'Nix'." 2025. [Online]. Available: https://cran.r-project.org/package=rix