The Journal of Open Source Software

posterior: Tools for Working with Posterior Distributions in R

Paul-Christian Bürkner [1], **Jonah Gabry** [2], **Matthew Kay** [3], **and Aki Vehtari** [4]

1 TU Dortmund University, Germany **2** Columbia University, USA **3** Northwestern University, USA **4** Aalto University and ELLIS Institute Finland, Finland

DOI: 10.21105/joss.10526

Software

- Review
- Repository
- Archive

Editor: Adithi R Upadhya

Reviewers:

- @ruarai
- @markean

Submitted: 06 February 2026
Published: 25 June 2026

Summary

Modern Bayesian inference is often performed via sampling algorithms that produce draws (samples) from the model's posterior distribution. The most important class of such algorithms is Markov chain Monte Carlo (MCMC), but also other algorithm classes such as variational inference and neural posterior estimation rely on posterior draws as their primary output representation. Regardless of their specific origin, these draws have to be stored and post-processed to obtain insights into the Bayesian inference results. In this context, relevant questions for the Bayesian modeler include in which format to store the posterior draws, which diagnostics to run to assess the trustworthiness of the obtained draws, and how to best summarize the draws for inference and decision-making. Due to the widespread use of sampling algorithms in Bayesian inference, essentially all Bayesian modelers face these questions during their data analyses, thus strongly benefiting from modern, efficient, and easy-to-use solutions.

Statement of need

The posterior R package (https://mc-stan.org/posterior) is intended to provide useful tools for both users and developers of Bayesian modeling software, focusing on manipulating, summarizing, and diagnosing the output of Bayesian models. The primary goals of `posterior` are three-fold:

(a) Efficiently convert between many different useful formats of draws. Existing packages for storing posterior draws, most notably coda (Plummer et al., 2006), support only a limited set of formats. However, in practice, different analyses and workflows require different formats. For example, when MCMC chain information is required, say, for convergence diagnostics, storing chains in an extra dimension is beneficial. In contrast, once convergence has been established, chain information becomes irrelevant and so dropping this information simplifies any subsequent operations. To accommodate users' and developers' practical needs, `posterior` natively supports several formats, with the ability to seamlessly switch between them.

(b) Provide consistent methods for operations commonly performed on draws, for example, subsetting, binding, mutating, or summarizing draws. These operations work consistently across all draws formats, thus providing a single interface regardless of format.

(c) Provide lightweight implementations of state-of-the-art posterior inference diagnostics (e.g., Vehtari et al., 2021). Especially for draws obtained via MCMC, convergence diagnostics are essential to assess the trustworthiness of the conclusions obtained from the draws. By providing a consistent and extensible interface for all these diagnostics, we not only ensure safe and easy use, but also easy extensibility whenever new, promising diagnostics are being proposed.

State of the field

The need to store, diagnose, and summarize posterior draws is an important challenge for essentially all Bayesian software ecosystems. The `coda` package (Plummer et al., 2006) has historically been the most widely used tool for handling MCMC draws in R. It defines the `mcmc` and `mcmc.list` objects to store draws from one chain and multiple chains, respectively. It also provides convergence diagnostics such as older versions of the $\widehat{R}$ diagnostic (Gelman & Rubin, 1992) and effective sample size measures. However, `coda` is limited to two draws formats and has not kept pace with more recently developed diagnostics or the diversity of sampling backends now commonly used in practice.

The `tidybayes` R package (Kay, 2024) aims to integrate Bayesian modeling into a tidy data framework, allowing users to leverage the full power of the `tidyverse` (Wickham et al., 2019) for Bayesian analysis. While `tidybayes` also summarizes posteriors, it does so mostly with the aim of visualization and uses `posterior` for many backend operations on the posterior draws. As such, it builds on rather than competes with `posterior`. Similarly, the `distributional` package (O'Hara-Wild et al., 2026) builds on `posterior` to create vectorized distribution objects for manipulating and visualizing probability distributions.

Outside of R, the Python library `ArviZ` (Kumar et al., 2019) provides diagnostics and visualizations, using the `xarray.DataTree` format (Hoyer & Hamman, 2017) for storing posterior draws. `ArviZ` has become the de facto standard for posterior analysis in Python. It shares several design goals with `posterior`, particularly the goal of being backend-agnostic. Both implement modern $\widehat{R}$ diagnostics and effective sample size measures (Vehtari et al., 2021). While `ArviZ` and `posterior` serve analogous roles in their respective languages, they differ in their approach to which draws formats they support: `posterior` exposes multiple native R formats (matrices, arrays, data frames, lists) to minimize friction for users moving in and out of the framework, whereas `ArviZ` unifies the storage of draws into a single `InferenceData` format.

Software design

`posterior`'s design follows the central idea of providing consistent and safe interfaces between all draws formats, operations, and diagnostics. As such, users and developers alike can rely on predictable output structures and error handling regardless of their draws format of choice. All draws formats contain information on (a) the *variables* (parameters and derived quantities) from whose distributions draws have been sampled and (b) the *draw indices* to safely map draws across variables, thus preserving their dependence structure. Some formats also contain information on (c) the *chain indices*, primarily useful for MCMC, to ensure the correct behavior of convergence diagnostics (Vehtari et al., 2021).

Except for one special format (see below), all formats are directly built on R base formats (matrices, arrays, lists, and data.frames) such that the objects are also easily usable outside of the methods that `posterior` itself provides. As such, users can easily move in and out of the `posterior` framework without facing major difficulties of converting their draws into any special format. The deliberate choice to support multiple formats required extra work on the internal interfaces and introduced the requirements for format-transforming methods. Nevertheless, we believe that the practical benefit of the multi-format approach justifies the internal code overhead necessary for its support.

The special `rvar` format (see https://mc-stan.org/posterior/articles/rvar.html) offers a multidimensional, sample-based representation of random variables. It is designed to act as much like base R arrays as possible, but removing the necessity to interact with the underlying draws directly. As such, `rvars` provides a convenient interface for working with random variables without having to worry about their internal representation. This forms the basis, for example, for the probability distribution objects of the `distributional` package.

Research impact statement

The `posterior` package is part of the Stan ecosystem of packages (https://mc-stan.org/tools), being tightly integrated into the post-processing methods of widely-used Stan-based packages such as `brms` (Bürkner, 2017), `rstanarm` (Goodrich et al., 2025), and `cmdstanr` (Gabry et al., 2025), among others. These packages have together been cited well over 10,000 times (according to Google Scholar; 2026/06/25) since they started to use `posterior` for handling posterior draws. Although it is not cited directly in most of these applications, `posterior` is sure to have been used in almost all of them for posterior summaries and convergence diagnostics, which are essential within the Bayesian workflow (Gelman et al., 2020). `posterior` itself has been cited directly over 100 times (according to Google Scholar; 2026/06/25) by methods, software, and application papers alike.

Despite its close connection to Stan, its modules and methods are generic and thus compatible with essentially all implementations of sampling algorithms available in R. It is currently imported or suggested by a growing list of over 60 other packages on CRAN, many of which build on sampling backends other than Stan, for example, `BayesMultiMode` (Baştürk et al., 2025), `bayesQRsurvey` (Rodríguez Taborda et al., 2025), or `jagstargets` (Landau, 2021). From the Posit CRAN mirror alone, `posterior` has been downloaded over 3.7 million times since its first release in August 2021. In 2025, it has been downloaded over 1 million times, which corresponds to roughly 20,000 weekly downloads on average. Overall, this showcases the importance of `posterior` for the area of sampling-based Bayesian inference.

AI usage disclosure

No generative AI tools were used in the development of this software, the writing of this manuscript, or the preparation of supporting materials.

Acknowledgements

We acknowledge contributions from Måns Magnusson, Rok Češnovar, Ben Lambert, Ozan Adıgüzel, Jacob Socolar, Noa Kallioinen, and Teemu Säilynoja.

References

Baştürk, N., Cross, J., de Knijff, P., Hoogerheide, L., Labonne, P., & van Dijk, H. (2025). *BayesMultiMode: Bayesian mode inference*. https://doi.org/10.32614/CRAN.package.BayesMultiMode

Bürkner, P.-C. (2017). brms: An R Package for Bayesian Multilevel Models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Gabry, J., Češnovar, R., Johnson, A., & Bronder, S. (2025). *cmdstanr: R interface to 'CmdStan'*. https://mc-stan.org/cmdstanr/

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*(4), 457–472. https://doi.org/10.1214/ss/1177011136

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., & Modrák, M. (2020). Bayesian workflow. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2011.01808

Goodrich, B., Gabry, J., Ali, I., & Brilleman, S. (2025). *rstanarm: Bayesian applied regression modeling via Stan*. https://doi.org/10.32614/cran.package.rstanarm

Hoyer, S., & Hamman, J. (2017). xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Kay, M. (2024). *tidybayes: Tidy data and geoms for Bayesian models*. https://doi.org/10.5281/zenodo.1308151

Kumar, R., Carroll, C., Hartikainen, A., & Martin, O. (2019). ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, *4*(33). https://doi.org/10.21105/joss.01143

Landau, W. M. (2021). The jagstargets R package: A reproducible workflow framework for Bayesian data analysis with JAGS. *Journal of Open Source Software*, *6*(68), 3877. https://doi.org/10.21105/joss.03877

O'Hara-Wild, M., Kay, M., Hayes, A., & Hyndman, R. (2026). *distributional: Vectorised probability distributions*. https://doi.org/10.32614/CRAN.package.distributional

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. In *R News* (No. 1; Vol. 6, pp. 7–11). https://journal.r-project.org/archive/

Rodríguez Taborda, T., Cardona Jiménez, J., Nascimento, M. L., & Mota Gonçalves, K. C. (2025). *bayesQRsurvey: Bayesian quantile regression models for complex survey data analysis*. https://doi.org/10.32614/CRAN.package.bayesQRsurvey

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-Normalization, Folding, and Localization: An Improved Rhat for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, *16*(2), 667–718. https://doi.org/10.1214/20-BA1221

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686