posterior: Tools for Working with Posterior Distributions in R

Paul-Christian Bürkner Jonah Gabry Matthew Kay
Aki Vehtari

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 packages for fitting Bayesian models or working with output from 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 single format. 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.

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 https://mc-stan.org/posterior/articles/rvar.html), 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.

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) 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 GoogleScholar) 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, Nalan, Jamie Cross, Peter de Knijff, Lennart Hoogerheide, Paul Labonne, and Herman van Dijk. 2025. *BayesMultiMode: Bayesian Mode Inference*. https://doi.org/10.32614/CRAN.package.BayesMultiMode.

Bürkner, Paul-Christian. 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, Jonah, Rok Češnovar, Andrew Johnson, and Steve Bronder. 2025. *Cmdstanr: R Interface to 'CmdStan'*. https://mc-stan.org/cmdstanr/.

Gelman, Andrew, Aki Vehtari, Daniel Simpson, Charles C Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. 2020. "Bayesian Workflow." *arXiv Preprint*. https://doi.org/10.48550/arXiv.2011.01808.

Goodrich, Ben, Jonah Gabry, Imad Ali, and Sam Brilleman. 2025. "Rstanarm: Bayesian Applied Regression Modeling via Stan." https://mc-stan.org/rstanarm/.

Landau, William Michael. 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.

Plummer, Martyn, Nicky Best, Kate Cowles, and Karen Vines. 2006. "CODA: Convergence Diagnosis and Output Analysis for MCMC." *R News* 6 (1): 7–11. https://journal.r-project.org/archive/.

Rodríguez Taborda, Tomás, Johnatan Cardona Jiménez, Marcus L. Nascimento, and Kelly Cristina Mota Gonçalves. 2025. *bayesQRsurvey: Bayesian Quantile Regression Models for Complex Survey Data Analysis*. https://doi.org/10.32614/CRAN.package.bayesQRsurvey.

Vehtari, Aki, Andrew Gelman, Daniel Simpson, and Carpenter. 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.