

# BayesPostEst: An R Package to Generate Postestimation Quantities for Bayesian MCMC Estimation

Shana Scogin<sup>1</sup>, Johannes Karreth<sup>2</sup>, Andreas Beger<sup>3</sup>, and Rob Williams<sup>4</sup>

<sup>1</sup> University of Notre Dame, South Bend, USA <sup>2</sup> Ursinus College, Collegeville, USA <sup>3</sup> Predictive Heuristics <sup>4</sup> Washington University in St. Louis, St. Louis, USA

DOI: [10.21105/joss.01722](https://doi.org/10.21105/joss.01722)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

**Submitted:** 05 September 2019

**Published:** 28 September 2019

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

BayesPostEst is an R (R Core Team, 2019) package with convenience functions to generate and present quantities of interest after estimating Bayesian regression models fit using MCMC via JAGS (Plummer, 2017), Stan (Stan Development Team, 2019), MCMCpack (Martin, Quinn, & Park, 2011), or other MCMC samplers. Quantities of interest include predicted probabilities and changes in probabilities in generalized linear models and analyses of model fit using ROC curves and precision-recall curves. The package also contains two functions to create publication-ready tables summarizing model results with an assessment of substantively meaningful effect sizes.

The package currently consists of seven functions:

- `mcmcTab`: Summarize Bayesian MCMC output in a table
- `mcmcReg`: Create regression tables for multiple Bayesian MCMC models using `texreg`
- `mcmcAveProb`: Calculate predicted probabilities using Bayesian MCMC estimates for the “Average Case”
- `mcmcObsProb`: Calculate predicted probabilities using Bayesian MCMC estimates using the “Observed Value” approach, calculating probabilities for the average of observed cases
- `mcmcFD`: Calculate first differences of a Bayesian logit or probit model
- `mcmcFDplot`: Plot first differences from MCMC output
- `mcmcRocPrc`: Generate ROC and precision-recall curves using Bayesian MCMC estimates

## Need and applications

A variety of existing packages offer outstanding functionalities to extract and visualize posterior distribution of estimates and some postestimation quantities; the final section of this paper lists these packages. BayesPostEst offers a further contribution by (1) providing convenient summaries of MCMC estimates as used by social scientists and (2) implementing methods for interpreting estimates in generalized linear models that are widely used in the social sciences. These methods are not currently available in this form within one package and workflow. BayesPostEst also brings the “region of practical equivalence” (Kruschke, 2013, 2018) to the widely used tool of calculating first differences after logit or probit models. Finally, BayesPostEst implements the popular model fit diagnostic of examining receiver-operating characteristic and precision-recall curves for Bayesian models.

## General setup

All functions in `BayesPostEst` work with distributions of posterior draws of regression coefficients. For further processing by `BayesPostEst` functions, these posterior draws need to be converted into a matrix. All functions in the package do this automatically for posterior draws generated by JAGS, BUGS, MCMCpack, rstan, and rstanarm. For posterior draws generated by other tools, users must convert these objects into a matrix, where rows represent iterations and columns represent parameters. The functions in `BayesPostEst` then use these posterior draws to generate postestimation quantities of interest that include measures of uncertainty based on posterior distributions.

## Summarizing Bayesian MCMC output in tables

`BayesPostEst` provides two functions to summarize Bayesian model results in tables: `mcmcTab` for summaries of one regression model with additional quantities of interest describing the posterior estimates, and `mcmcReg` for publication-ready regression tables summarizing estimates from one or multiple models.

First, `mcmcTab` generates a table summarizing the posterior distributions of all parameters contained in the model object. This table can then be used to summarize parameter quantities for one model in detail. By default, `mcmcTab` generates a dataframe with one row per parameter and columns containing the median, standard deviation, and 95% credible interval of each parameter's posterior distribution. Users can add a column to the table that calculates the percent of posterior draws that have the same sign as the median of the posterior distribution.

Users can also define a "region of practical equivalence" (ROPE, see Kruschke, 2013, 2018). This region is a band of values around 0 that are "practically equivalent" to 0, or a "negligible" effect. As discussed in the online supplement to Kruschke (2018), in an example of analyzing individual responses of opinion polls, one can think of the ROPE capturing the range of a margin of error in an aggregate poll. If a common margin of error is  $\pm 0.03$ , then for logistic regression coefficients in analyses of predictors of individual choice such a range of negligible coefficient values might be  $[-0.06; 0.06]$ .

```
mcmcTab(fit,
  pars = c("female", "neuroticism", "extraversion"),
  ROPE = c(-0.1, 0.1))
```

```
## This table contains an estimate for parameter values outside of the region of
## practical equivalence (ROPE). For this quantity to be meaningful, all
## must be on the same scale (e.g. standardized coefficients or first di
## Variable Median SD Lower Upper PrOutROPE
## 1 female 0.237 0.111 0.023 0.462 0.886
## 2 neuroticism 0.055 0.108 -0.144 0.276 0.342
## 3 extraversion 0.518 0.112 0.296 0.733 1.000
```

To define a ROPE, it can be useful for all parameters (e.g. regression coefficients) to be on the same scale because `mcmcTab` accepts only one definition of ROPE for all parameters. Users can standardize continuous predictors to achieve this, for instance by dividing them by two standard deviations (Gelman, 2008).

The output from `mcmcTab` can be exported to be inserted into a variety of document types using appropriate R packages, including `flextable` (Gohel, 2019), `xtable` (Dahl, Scott, Roosen, Magnusson, & Swinton, 2019), or `knitr` (Xie, 2019).

Second, `mcmcReg` generates tables summarizing estimates from *multiple* Bayesian models. The function then uses `texreg` (Leifeld, 2013) to produce publication-ready tables in HTML or LaTeX format. This gives users a convenient and time-saving option to present estimates from different model specifications alongside each other. Beyond the arguments explicitly mentioned in the help file for `mcmcReg`, users can also supply any of the arguments that are part of the `texreg` function. The code below produces the following table:

```
mcmcReg(mod = list(fit1, fit2),
  pars = c("(Intercept)", "female", "neuroticism", "extraversion"),
  pointest = "median",
  coefnames = list(
    c("Intercept", "Female"),
    c("Intercept", "Female", "Neuroticism", "Extraversion")),
  ci = 0.95,
  format = "latex",
  caption = "Determinants of volunteering",
  caption.above = TRUE)
```

Table 1: Determinants of volunteering

|              | Model 1                  | Model 2                  |
|--------------|--------------------------|--------------------------|
| Intercept    | −0.46*<br>[−0.62; −0.30] | −0.46*<br>[−0.62; −0.31] |
| Female       | 0.25*<br>[0.03; 0.47]    | 0.24*<br>[0.02; 0.46]    |
| Neuroticism  |                          | 0.06<br>[−0.14; 0.28]    |
| Extraversion |                          | 0.52*<br>[0.30; 0.73]    |

\* 0 outside 95% credible interval.

Figure 1: Table produced with the `mcmcReg` function.

## Predicted probabilities and first differences

`BayesPostEst` builds on estimates from a Bayesian generalized linear model with  $k$  covariates  $x$  and the inverse logit (or probit) link function, where

$$\Pr(y = 1|x_k) = \text{logit}^{-1}(\beta_1 + \beta_2 x_k)$$

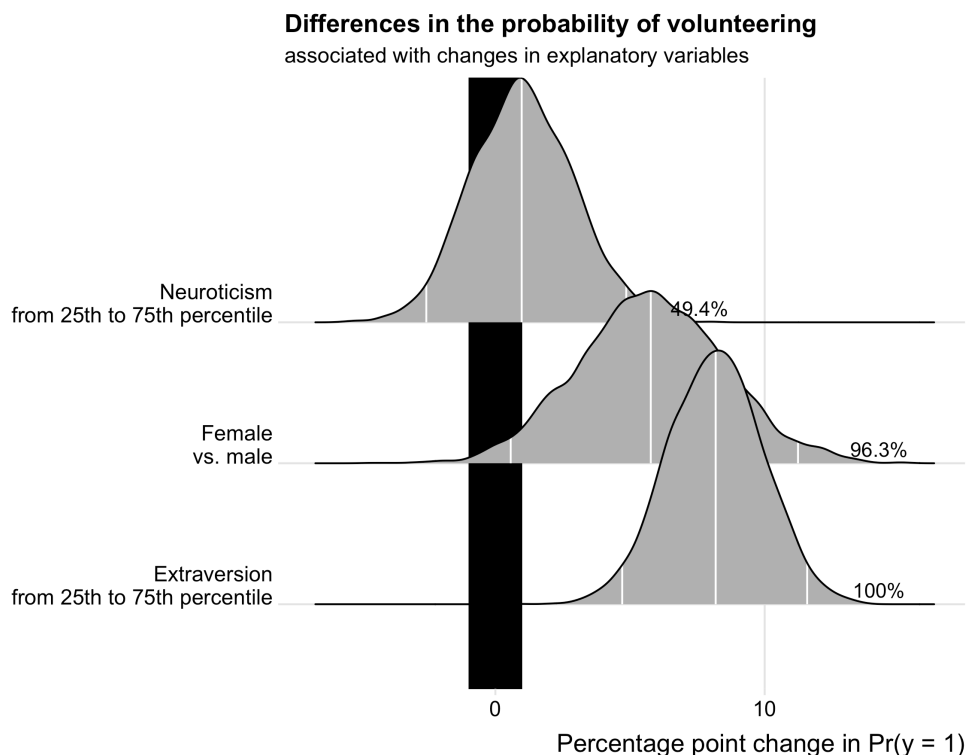
or

$$\Pr(y = 1|x_k) = \Phi(\beta_1 + \beta_2 x_k)$$

To evaluate the relationship between covariates and a binary outcome, `mcmcAveProb` calculates the predicted probability ( $Pr(y = 1)$ ) at pre-defined values of one covariate of interest ( $x$ ), while all other covariates are held at a “typical” value. This follows suggestions outlined in King, Tomz, & Wittenberg (2000) and elsewhere, which are commonly adopted by users of GLMs. The `mcmcAveProb` function by default calculates the median value of all covariates other than  $x$  as “typical” values. Users can then access the full posterior distributions of the predicted probability at different values of  $x$ , or the median and a credible interval of the same quantity.

As an alternative to probabilities for “typical” cases, Hanmer & Kalkan (2013) suggest to calculate predicted probabilities for all observed cases and then derive an “average effect”. In their words, the goal of this postestimation “is to obtain an estimate of the average effect in the population ... rather than seeking to understand the effect for the average case.” `BayesPostEst` allows users to calculate these quantities for binary and continuous predictors based on observed values of covariates, using the `mcmcObsProb` function. Users can then summarize or visualize these quantities as is common in the social sciences (Hanmer & Kalkan, 2013; King et al., 2000; Long, 1997)

To summarize typical effects across covariates rather than full predicted probabilities, users can generate “first differences” (King et al., 2000; Long, 1997). This quantity represents, for each covariate, the difference in predicted probabilities for cases with low and high values of the respective covariate. For each of these differences, all other variables are held constant at their median. To make use of the full posterior distribution of first differences, `BayesPostEst` provides a dedicated plotting function. `mcmcFDplot` returns a `ggplot2` object that can be further customized. The function is modeled after Figure 1 in Karreth (2018). Users can specify a region of practical equivalence and print the percent of posterior draws to the right or left of the ROPE. If ROPE is not specified, the figure automatically prints the percent of posterior draws to the left or right of 0.



**Figure 2:** Output from the `mcmcFDplot` function. See the package vignette for more details.

## Model fit analyses using ROC and Precision-Recall curves

One way to assess the fit of statistical models fit on binary outcomes is to calculate the area under the Receiver Operating Characteristic (ROC) and Precision-Recall curves. A short description of these curves and their utility for model assessment is provided in Beger (2016). Both methods assess the trade-off between true and false positives. ROC curves become less useful as outcomes of interest (or observed 1s) become rare; precision-recall curves are a more suitable tool for such rare outcomes. The `mcmcRocPrc` function produces an object with four elements: the area under the ROC curve, the area under the PR curve, and two dataframes to plot each curve. While the area under each curve can be a useful summary of model fit, plotting the curves can serve to assess the model's trade-off in more detail.

## Comparison to other packages

The following packages provide functions similar (concretely or in spirit) to `BayesPostEst`. `BayesPostEst` is inspired by some of the functionality in these packages, but aims to combine tools commonly used in the social sciences in one workflow.

- `tidybayes` (Kay, 2019) offers a tidy workflow for extracting posterior distributions of estimates, fits, and predictions.
- `bayesplot` (Gabry & Mahr, 2019) offers various plotting options for posterior quantities, including posterior predictive checks.
- `bayestable` (Ananyev, 2018) generates a regression table from MCMC estimates that can be passed on to the `texreg` package (Leifeld, 2013) for printing.
- The `sjstats` (Lüdtke, 2019a) and `sjPlot` (Lüdtke, 2019b) suite of packages allows for a variety of postestimation commands, including predicted probabilities, marginal effects, and a function to evaluate estimates in relationship to a user-defined ROPE.
- Similarly, `bayestestR` (Makowski, Ben-Shachar, & Lüdtke, 2019) offers a broad set of functions to analyze and describe posterior distributions of coefficients, but not other post-estimation quantities of interest.
- The `ggmcmc` (Fernández-i-Marín, 2016) package contains a function to plot the ROC curve after a regression model for binary outcomes.
- The `brms` (Bürkner, 2017) package offers a variety of convenient postestimation commands, including predicted probabilities, for Bayesian models estimated directly in `brms`.

## Future developments

Plans for future work include extending existing methods in this package to a broader class of models, including generalized linear models with other link functions (e.g. models for ordered, categorical, and count outcomes) and multilevel/hierarchical generalized linear models.

## Contributions

At the time of submission of this manuscript, the authors have contributed to the project as follows. S.S. created the package, improved functions, wrote function documentation, tested the package, and is the package maintainer. J.K. wrote most of the initial functions, the package vignette, and the JOSS manuscript. R.W. wrote the `mcmcReg` function and contributes to package development and optimization. A.B. rewrote the `mcmcRocPrc` function and reviewed code coverage.

## References

- Ananyev, M. (2018). *Bayestable: Summarizing and presenting results of mcmc inference in plain text/latex/html*. Retrieved from <https://github.com/ananyevm/bayestable/>
- Beger, A. (2016). Precision-recall curves. Available at SSRN: <https://ssrn.com/abstract=2765419>. Retrieved from <http://dx.doi.org/10.2139/ssrn.2765419>
- Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1), 1–28. doi:10.18637/jss.v080.i01
- Dahl, D. B., Scott, D., Roosen, C., Magnusson, A., & Swinton, J. (2019). *Xtable: Export tables to latex or html*. Retrieved from <https://CRAN.R-project.org/package=xtable>
- Fernández-i-Marín, X. (2016). ggmcmc: Analysis of MCMC samples and Bayesian inference. *Journal of Statistical Software*, 70(9), 1–20. doi:10.18637/jss.v070.i09
- Gabry, J., & Mahr, T. (2019). Bayesplot: Plotting for bayesian models. Retrieved from [mc-stan.org/bayesplot](http://mc-stan.org/bayesplot)
- Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations. *Statistics in Medicine*, 27(15), 2865–2873. doi:10.1002/sim.3107
- Gohel, D. (2019). *Flextable: Functions for tabular reporting*. Retrieved from <https://CRAN.R-project.org/package=flextable>
- Hanmer, M. J., & Kalkan, K. O. (2013). Behind the Curve: Clarifying the Best Approach to Calculating Predicted Probabilities and Marginal Effects from Limited Dependent Variable Models. *American Journal of Political Science*, 57(1), 263–277. doi:10.1111/j.1540-5907.2012.00602.x
- Karreth, J. (2018). The economic leverage of international organizations in interstate disputes. *International Interactions*, 44(3), 463–490. doi:10.1080/03050629.2018.1389728
- Kay, M. (2019). *tidybayes: Tidy data and geoms for Bayesian models*. doi:10.5281/zenodo.1308151
- King, G., Tomz, M., & Wittenberg, J. (2000). Making the Most of Statistical Analyses: Improving Interpretation and Presentation. *American Journal of Political Science*, 44(2), 347–361. Retrieved from <http://www.jstor.org/stable/2669316>
- Kruschke, J. K. (2013). Bayesian estimation supersedes the t-test. *Journal of Experimental Psychology: General*, 142(2), 573–603. doi:10.1037/a0029146
- Kruschke, J. K. (2018). Rejecting or accepting parameter values in bayesian estimation. *Advances in Methods and Practices in Psychological Science*, 1(2), 270–280. doi:10.1177/2515245918771304
- Leifeld, P. (2013). texreg: Conversion of statistical model output in R to LaTeX and HTML tables. *Journal of Statistical Software*, 55(8), 1–24. Retrieved from <http://www.jstatsoft.org/v55/i08/>
- Long, J. S. (1997). *Regression models for categorical and limited dependent variables*. Thousand Oaks: Sage Publications.
- Lüdtke, D. (2019a). *Sjstats: Statistical functions for regression models (version 0.17.5)*. doi:10.5281/zenodo.1284472
- Lüdtke, D. (2019b). *SjPlot: Data visualization for statistics in social science*. doi:10.5281/zenodo.1308157
- Makowski, D., Ben-Shachar, M. S., & Lüdtke, D. (2019). BayestestR: Describing effects and their uncertainty, existence and significance within the bayesian framework. *Journal of Open Source Software*, 4(40), 1541. doi:10.21105/joss.01541

Martin, A. D., Quinn, K. M., & Park, J. H. (2011). MCMCpack: Markov chain monte carlo in R. *Journal of Statistical Software*, 42(9), 22. Retrieved from <http://www.jstatsoft.org/v42/i09/>

Plummer, M. (2017). JAGS Version 4.3.0 User Manual. Retrieved from <http://sourceforge.net/projects/mcmc-jags/files/Manuals/4.x/>

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>

Stan Development Team. (2019). *RStan: The R interface to Stan*. Retrieved from <http://mc-stan.org/>

Xie, Y. (2019). *Knitr: A general-purpose package for dynamic report generation in r*. Retrieved from <https://yihui.name/knitr/>