Package ‘brmsmargins’

December 16, 2021

Title  Bayesian Marginal Effects for 'brms' Models
Version  0.1.1

URL  https://joshuawiley.com/brmsmargins/,
     https://github.com/JWiley/brmsmargins

BugReports  https://github.com/JWiley/brmsmargins/issues

Description  Calculate Bayesian marginal effects and average marginal effects for models fit using the 'brms' package including fixed effects, mixed effects, and location scale models. These are based on marginal predictions that integrate out random effects if necessary (see for example <doi:10.1186/s12874-015-0046-6>).

License  GPL (>= 3)
Encoding  UTF-8
RoxygenNote  7.1.2
Depends  R (>= 4.0.0)
Imports  methods, stats, data.table (>= 1.12.0), extraoperators (>= 0.1.1), brms, bayestestR, Rcpp, posterior
Suggests  testthat (>= 3.0.0), covr, withr, knitr, rmarkdown, margins, betareg

Config/testthat/edition  3
Config/testthat/parallel  true
LinkingTo  RcppArmadillo, Rcpp
VignetteBuilder  knitr

NeedsCompilation  yes

Author  Joshua F. Wiley [aut, cre] (<https://orcid.org/0000-0002-0271-6702>)
Maintainer  Joshua F. Wiley <jwiley.psych@gmail.com>

Repository  CRAN
Date/Publication  2021-12-16 10:20:05 UTC
**R topics documented:**

- `brmsmargins` ........................................................................... 2
- `bsummary` .............................................................................. 6
- `integratemvn` ........................................................................ 8
- `integratre` ............................................................................. 9
- `prediction` ............................................................................. 10
- `rowBootMeans` ......................................................................... 11
- `tab2mat` ................................................................................. 12

---

## `brmsmargins`

*Calculate Marginal Effects from 'brms' Models*

### Description

This function is designed to help calculate marginal effects including average marginal effects (AMEs) from `brms` models.

### Usage

```r
brmsmargins(
  object,
  at = NULL,
  add = NULL,
  newdata = model.frame(object),
  CI = 0.99,
  CIType = "HDI",
  contrasts = NULL,
  ROPE = NULL,
  MID = NULL,
  subset = NULL,
  dpar = NULL,
  seed,
  ...
)
```

### Arguments

- **object**: A fitted `brms` model object. Required.
- **at**: An optional object inheriting from data frame indicating the values to hold specific variables at when calculating average predictions. This is intended for AMEs from categorical variables. Currently only one of `at` or `add` can be specified.
add An optional object inheriting from data frame indicating the values to add to specific variables at when calculating average predictions. This is intended for AMEs for continuous variables. Currently only one of at or add can be specified.

newdata An object inheriting from data frame indicating the baseline values to use for predictions and AMEs. Defaults to be the model frame.

CI A numeric value specifying the width of the credible interval. Defaults to 0.99.

CIType A character string specifying the type of credible interval (e.g., highest density interval). It is passed down to bsummary which in turn passes it to ci. Defaults to “HDI”.

contrasts An optional contrast matrix. The posterior predictions matrix is post multiplied by the contrast matrix, so they must be conformable. The posterior predictions matrix has a separate column for each row in the at or add object, so the contrast matrix should have the same number of rows. It can have multiple columns, if you desire multiple specific contrasts.

ROPE Either left as NULL, the default, or a numeric vector of length 2, specifying the lower and upper thresholds for the Region of Practical Equivalence (ROPE).

MID Either left as NULL, the default, or a numeric vector of length 2, specifying the lower and upper thresholds for a Minimally Important Difference (MID). Unlike the ROPE, percentages for the MID are calculated as at or exceeding the bounds specified by this argument, whereas the ROPE is the percentage of the posterior at or inside the bounds specified.

subset A character string that is a valid R expression used to subset the dataset passed in newdata, prior to analysis. Defaults to NULL.

dpar Parameter passed on the dpar argument of fitted() in brms. Defaults to NULL indicating the mean or location parameter typically.

seed Argument that controls whether (and if so what) random seed to use. This does not matter when using fixed effects only. However, when using Monte Carlo integration to integrate out random effects from mixed effects models, it is critical if you are looking at a continuous marginal effect with some small offset value as otherwise the Monte Carlo error from one set of predictions to another may exceed the true predicted difference. If seed is left missing, the default, than a single, random integer between +\(-1e7\) is chosen and used to set the seed before each prediction. If manually chosen (recommended for reproducibility), the seed should either be a single value, in which case this single value is used to set the seed before each prediction. Alternately, it can be a vector of seeds with either the same length as the number of rows in at or add, whichever was specified. This is probably generally not what you want, as it means that even for the same input data, you would get slightly different predictions (when integrating out random effects) due to Monte Carlo variation. Finally, rather than being missing, you can explicitly set seed = NULL, if you do not want any seed to be set. This would be fine, for instance, when only using fixed effects, or if you know what you are doing and intend that behavior when integrating out random effects.

... Additional arguments passed on to prediction. In particular, the effects argument of prediction() is important for mixed effects models to control how
random effects are treated in the predictions, which subsequently changes the marginal effect estimates.

Details

The main parts required for the function are a fitted model object, (via the object argument) a dataset to be used for prediction, (via the newdata argument which defaults to the model frame), and a dataset passed to either at or add. The steps are as follows:

1. Check that the function inputs (model object, data, etc.) are valid.
2. Take the dataset from the newdata argument and either add the values from the first row of add or replace the values using the first row of at. Only variables specified in at or add are modified. Other variables are left as is.
3. Use the fitted() function to generate predictions based on this modified dataset. If effects is set to “fixedonly” (meaning only generate predictions using fixed effects) or to “includeRE” (meaning generate predictions using fixed and random effects), then predictions are generated entirely using the fitted() function and are, typically back transformed to the response scale. For mixed effects models with fixed and random effects where effects is set to “integrateoutRE”, then fitted() is only used to generate predictions using the fixed effects on the linear scale. For each prediction generated, the random effects are integrated out by drawing k random samples from the model assumed random effect(s) distribution. These are added to the fixed effects predictions, back transformed, and then averaged over all k random samples to perform numerical Monte Carlo integration.
4. All the predictions for each posterior draw, after any back transformation has been applied, are averaged, resulting in one, marginal value for each posterior draw. These are marginal predictions. They are average marginal predictions if averaging over the sample dataset, or may be marginal predictions at the means, if the initial input dataset used mean values, etc.
5. Steps two to four are repeated for each row of at or add. Results are combined into a matrix where the columns are different rows from at or add and the rows are different posterior draws.
6. If contrasts were specified, using a contrast matrix, the marginal prediction matrix is post multiplied by the contrast matrix. Depending on the choice(s) of add or at and the values in the contrast matrix, these can then be average marginal effects (AMEs) by using numerical integration (add with 0 and a very close to 0 value) or discrete difference (at with say 0 and 1 as values) for a given predictor(s).
7. The marginal predictions and the contrasts, if specified are summarized.

Although brmsmargins() is focused on helping to calculate marginal effects, it can also be used to generate marginal predictions, and indeed these marginal predictions are the foundation of any marginal effect estimates. Through manipulating the input data, at or add and the contrast matrix, other types of estimates averaged or weighting results in specific ways are also possible.

Value

A list with four elements.

- Posterior: Posterior distribution of all predictions. These predictions default to fixed effects only, but by specifying options to prediction() they can include random effects or be predictions integrating out random effects.
• Summary
A summary of the predictions.

• Contrasts
Posterior distribution of all contrasts, if a contrast matrix was specified.

• ContrastSummary
A summary of the posterior distribution of all contrasts, if specified

References

Examples
## Not run:
#### Testing ####
## sample data and logistic model with brms
set.seed(1234)
Tx <- rep(0:1, each = 50)
ybin <- c(rep(0:1, c(40,10)), rep(0:1, c(10,40)))
logitd <- data.frame(Tx = Tx, ybin = ybin)
logitd$x <- rnorm(100, mean = logitd$ybin, sd = 2)

mbin <- brms::brm(ybin ~ Tx + x, data = logitd, family = brms::bernoulli())
summary(mbin)

## now check AME for Tx
tmp <- brmsmargins(
  object = mbin,
  at = data.table::data.table(Tx = 0:1),
  contrasts = matrix(c(-1, 1), nrow = 2),
  ROPE = c(-.05, +.05),
  MID = c(-.10, +.10))
tmp$Summary
tmp$ContrastSummary ## Tx AME

## now check AME for Tx with bootstrapping the AME population
tmpalt <- brmsmargins(
  object = mbin,
  at = data.table::data.table(Tx = 0:1),
  contrasts = matrix(c(-1, 1), nrow = 2),
  ROPE = c(-.05, +.05),
  MID = c(-.10, +.10),
  resample = 100L)
tmpalt$Summary
tmpalt$ContrastSummary ## Tx AME
## now check AME for continuous predictor, x
## use .01 as an approximation for first derivative
## 1 / .01 in the contrast matrix to get back to a one unit change metric

```r
tmp2 <- brmsmargins(
  object = mbin,
  add = data.table::data.table(x = c(0, .01)),
  contrasts = matrix(c(-1/.01, 1/.01), nrow = 2),
  ROPE = c(-.05, +.05),
  MID = c(-.10, +.10))
```

```r
summary(fit, prob = 0.99)
```

```r
tmp <- brmsmargins(
  object = fit,
  at = data.table::data.table(Days = 0:1),
  contrasts = matrix(c(-1, 1), nrow = 2),
  ROPE = c(-.05, +.05),
  MID = c(-.10, +.10), CIType = "ETI", effects = "integrateoutRE", k = 5L)
```

```r
tmp$Summary
tmp$ContrastSummary
```

## End(Not run)

---

**bsummary**

**Personal Preference Based Bayesian Summary**

**Description**

Returns a summary of a posterior distribution for a single parameter / value. It is based on personal preference. Notably, it does not only use `bayestestR::describe_posterior`, an excellent function, because of the desire to also describe the percentage of the full posterior distribution that is at or exceeding the value of a Minimally Important Difference (MID). MIDs are used in clinical studies with outcome measures where there are pre-defined differences that are considered clinically important, which is distinct from the ROPE or general credible intervals capturing uncertainty.

**Usage**

```r
bsummary(x, CI = 0.99, CIType = "HDI", ROPE = NULL, MID = NULL)
```
Arguments

- **x**: The posterior distribution of a parameter
- **CI**: A numeric value indicating the desired width of the credible interval. Defaults to 0.99 currently, but this is subject to change. A 99% interval was chosen as the default as there have been recent arguments made in the realm of meta science that there are, essentially, too many false positives and that many of the “findings” in science are not able to be replicated. In any case, users should ideally specify a desired CI width, and not rely on defaults.
- **CIType**: A character string indicating the type of credible interval, passed on to the **ci** function as the method for CIs.
- **ROPE**: Either left as NULL, the default, or a numeric vector of length 2, specifying the lower and upper thresholds for the Region of Practical Equivalence (ROPE).
- **MID**: Either left as NULL, the default, or a numeric vector of length 2, specifying the lower and upper thresholds for a Minimally Important Difference (MID). Unlike the ROPE, percentages for the MID are calculated as at or exceeding the bounds specified by this argument, whereas the ROPE is the percentage of the posterior at or inside the bounds specified.

Value

A data.table with the mean, \( M \)

- **\( M \)**: the mean of the posterior samples
- **\( \text{Mdn} \)**: the median of the posterior samples
- **\( \text{LL} \)**: the lower limit of the credible interval
- **\( \text{UL} \)**: the upper limit of the credible interval
- **\( \text{PercentROPE} \)**: the percentage of posterior samples falling into the ROPE
- **\( \text{PercentMID} \)**: the percentage of posterior samples falling at or beyond the MID
- **\( \text{CI} \)**: the width of the credible interval used
- **\( \text{CIType} \)**: the type of credible interval used (e.g., highest density interval)
- **\( \text{ROPE} \)**: a label describing the values included in the ROPE
- **\( \text{MID} \)**: a label describing the values included in the MID

References


Examples

```r
bsummary(rnorm(1000))

bsummary(rnorm(1000), ROPE = c(-.5, .5), MID = c(-1, 1))
```
**integratemvn**  
*Integrate over Multivariate Normal Random Effects*

**Description**

Used in the process of Monte Carlo integration over multivariate normal random effects. This generates the random draws from the multivariate normal distribution and multiplies these by the data. Not intended to be called directly by most users.

**Usage**

integratemvn(X, k, sd, chol)

integratemvnR(X, k, sd, chol)

**Arguments**

- **X**: A numeric matrix of the data to be multiplied by the random effects
- **k**: An integer, the number of random samples to be used for numerical integration
- **sd**: A numeric vector of the standard deviations
- **chol**: A numeric matrix, which should be the Cholesky decomposition of the correlation matrix of the multivariate normal distribution.

**Value**

A numeric matrix with random values

**Functions**

- integratemvnR: Pure R implementation of integratemvn

**Examples**

integratemvn(
    X = matrix(1, 1, 2),
    k = 100L,
    sd = c(10, 5),
    chol = chol(matrix(c(1, .5, .5, 1), 2)))

integratemvn(matrix(1, 1, 1), 100L, c(5), matrix(1))
**integratere**

**Integrate over Random Effects**

**Description**

Used to conduct Monte Carlo integration over Gaussian random effects. Not intended to be called directly by most users.

**Usage**

```r
integratere(d, sd, L, k, yhat, backtrans)
integratereR(d, sd, L, k, yhat, backtrans)
```

**Arguments**

- `d`: A list with model matrices for each random effect block.
- `sd`: A list with standard deviation matrices for each random effect block where rows are different posterior draws.
- `L`: A list with matrices for each random effect block containing the parts of the L matrix, the Cholesky decomposition of the random effect correlation matrix.
- `k`: An integer, the number of samples for Monte Carlo integration.
- `yhat`: A matrix of the fixed effects predictions.
- `backtrans`: An integer, indicating the type of back transformation. 0 indicates inverse logit (e.g., for logistic regression). 1 indicates exponential (e.g., for poisson or negative binomial regression or if outcome was natural log transformed). 2 indicates square (e.g., if outcome was square root transformed). Any other integer may be used for no transformation.

**Value**

A numeric matrix with the Monte Carlo integral calculated.

**Functions**

- `integratereR`: Pure R implementation of `integratere`

**Examples**

```r
integratere(
  d = list(matrix(1, 1, 1)),
  sd = list(matrix(1, 2, 1)),
  L = list(matrix(1, 2, 1)),
  k = 10L,
  yhat = matrix(0, 2, 1),
  backtrans = 0L)
```
prediction

Marginal Posterior Predictions from a 'brms' Model

Description

Calculate marginal predictions from a brms model. Marginal predictions average over the input data for each posterior draw. Marginal predictions for models with random effects will integrate over random effects.

Usage

prediction(
  object,  # A fitted brms model object. Required.
  data,    # A data frame or data table passed to fitted() as the new data to be used for predictions. Required.
  summarize = TRUE,  # A logical value, whether or not to calculate summaries of the posterior predictions. Defaults to TRUE.
  posterior = FALSE,  # A logical value whether or not to save and return the posterior samples. Defaults to FALSE as the assumption is a typical use case is to return the summaries only.
  index,     # An optional integer vector, giving the posterior draws to be used in the calculations. If omitted, defaults to all posterior draws.
  dpar = NULL,  # Parameter passed on the dpar argument of fitted() in brms. Defaults to NULL indicating the mean or location parameter typically.
  resample = 0L,  # An integer indicating the number of bootstrap resamples of the posterior predictions to use when calculating summaries. Defaults to 0L. See documentation from .averagePosterior() for more details.
  resampleseed = FALSE,  # A seed for random number generation. Defaults to FALSE, which means no seed is set. Only used if resample is a positive, non-zero integer. See documentation from .averagePosterior() for more details.
  ...  # ...
**rowBootMeans**

**effects** A character string indicating the type of prediction to be made. Can be one of “fixedonly” meaning only use fixed effects, “includeRE” meaning that random effects should be included in the predictions, or “integrateoutRE” meaning that random effects should be integrated out / over in the predictions.

**backtrans** A character string indicating the type of back transformation to be applied. Can be one of “response” meaning to use the response scale, “linear” or “identity” meaning to use the linear predictor scale, or a specific back transformation desired, from a possible list of “invlogit”, “exp”, or “square”. Custom back transformations should only be needed if, for example, the outcome variable was transformed prior to fitting the model.

**k** An integer providing the number of random draws to use for integrating out the random effects. Only relevant when effects is “integrateoutRE”.

... Additional arguments passed to fitted()

**Value**

A list with Summary and Posterior. Some of these may be NULL depending on the arguments used.

**References**


---

**Description**

This takes a numeric matrix, bootstrap resamples each row, and then calculates the mean. The intended use case is for Bayesian posterior predictions from sample data. Instead of directly calculating the average marginal effect (AME) across all observed values, these can be bootstrapped, so that uncertainty in the target population, and thus the AME in the target population, can be incorporated. Model uncertainty is already assumed to be handled by the different posterior samples, which are assumed to be across rows.

**Usage**

rowBootMeans(x)

**Arguments**

x A numeric matrix

**Value**

A numeric vector with the simple bootstrapped row means of the matrix
**Examples**

```r
x <- matrix(1:9, byrow = TRUE, 3)
replicate(10, rowBootMeans(x))
```

---

**tab2mat**  
*Convert a Row of a Table to a Square Matrix*

**Description**

Utility function to convert a row matrix to a square matrix. Used as the `brms` package returns things like the Cholesky decomposition matrix as separate columns where rows are posterior draws. Not intended to be called directly by most users.

**Usage**

```r
tab2mat(X)
tab2matR(X)
```

**Arguments**

- `X`  
  a matrix

**Value**

A numeric matrix with one row.

**Functions**

- `tab2matR`: Pure R implementation of `tab2mat`

**Examples**

```r
tab2mat(matrix(1:4, 1))
tab2mat(matrix(1:9, 1))
```
Index

.averagePosterior(), 10
brmsmargins, 2
bsummary, 3, 6
ci, 3, 7
integratemvn, 8
integratemvnR(integratemvn), 8
integratere, 9
integratereR(integratere), 9
prediction, 3, 10
prediction(), 3, 4
rowBootMeans, 11
tab2mat, 12
tab2matR(tab2mat), 12