Package 'brms.mmrm'

February 15, 2024

Title Bayesian MMRMs using 'brms'

Version 0.1.0

Description The mixed model for repeated measures (MMRM) is a popular model for longitudinal clinical trial data with continuous endpoints, and 'brms' is a powerful and versatile package for fitting Bayesian regression models. The 'brms.mmrm' R package leverages 'brms' to run MMRMs, and it supports a simplified interfaced to reduce difficulty and align with the best practices of the life sciences. References: Bürkner (2017) <doi:10.18637/jss.v080.i01>, Mallinckrodt (2008) <doi:10.1177/009286150804200402>.

License MIT + file LICENSE

URL https://openpharma.github.io/brms.mmrm/, https://github.com/openpharma/brms.mmrm

BugReports https://github.com/openpharma/brms.mmrm/issues

Depends R (>= 4.0.0)

Imports brms (>= 2.19.0), coda, dplyr, emmeans (>= 1.8.7), ggplot2, ggridges, MASS, posterior, purrr, rlang, stats, tibble, tidyr, tidyselect, trialr, utils, zoo

Suggests BH, fst, gt, gtsummary, knitr (>= 1.30), markdown (>= 1.1), mmrm, parallel, Rcpp, RcppEigen, RcppParallel, rmarkdown (>= 2.4), rstan, StanHeaders, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.1

NeedsCompilation no

Author William Michael Landau [aut, cre] (<https://orcid.org/0000-0003-1878-3253>), Kevin Kunzmann [aut] (<https://orcid.org/0000-0002-1140-7143>),

Yoni Sidi [aut],
Christian Stock [aut] (<https://orcid.org/0000-0002-3493-3234>),
Eli Lilly and Company [cph, fnd],
Boehringer Ingelheim Pharma GmbH & Co. KG [cph, fnd]

Maintainer William Michael Landau <will.landau.oss@gmail.com>

Repository CRAN

Date/Publication 2024-02-15 07:50:02 UTC

R topics documented:

`brms.mmrm-package` *brms.mmrm: Bayesian MMRMs using* `brms`

Description

The mixed model for repeated measures (MMRM) is a popular model for longitudinal clinical trial data with continuous endpoints, and `brms` a is powerful and versatile package for fitting Bayesian regression models. The `brms.mmrm` R package leverages `brms` to run MMRMs, and it supports a simplified interfaced to reduce difficulty and align with the best practices of the life sciences.

References

- Bürkner, P.-C. (2017), "brms: An R package for Bayesian multilevel models using Stan," Journal of Statistical Software, 80, 1–28. https://doi.org/10.18637/jss.v080.i01.
- Mallinckrodt, C. H., Lane, P. W., Schnell, D., and others (2008), "Recommendations for the primary analysis of continuous endpoints in longitudinal clinical trials," Therapeutic Innovation and Regulatory Science, 42, 303–319. https://doi.org/10.1177/009286150804200402.
- Mallinckrodt, C. H., and Lipkovich, I. (2017), Analyzing longitudinal clinical trial data: A practical guide, CRC Press, Taylor & Francis Group.

`brm_data` *Create and preprocess an MMRM dataset.*

Description

Create a dataset to analyze with an MMRM.

Usage

```
brm_data(
  data,
  outcome = "CHG",
  role = "change",
  baseline = NULL,
  group = "TRT01P",
  subgroup = NULL,
  time = "AVISIT",
  patient = "USUBJID",
  covariates = character(0L),
  missing = NULL,
  reference_group = "Placebo",
  level_control = NULL,
  reference_subgroup = NULL,
  reference_time = NULL,
  level_baseline = NULL
)
```

Arguments

`data`	Data frame or tibble with longitudinal data.
`outcome`	Character of length 1, name of the outcome variable.
`role`	Character of length 1. Either `"response"` if `outcome` is the raw response variable (e.g. AVAL) or `"change"` if `outcome` is change from baseline (e.g. CHG).
`baseline`	Character of length 1, name of the baseline response variable. Only relevant if the response variable is change from baseline. Supply `NULL` to ignore or omit.

group — Character of length 1, name of the treatment group variable. Must point to a character vector in the data. Factors are converted to characters.

subgroup — Character of length 1, optional name of the a discrete subgroup variable. Set to NULL to omit the subgroup (default).

time — Character of length 1, name of the discrete time variable. Must point to a character vector in the data. Factors are converted to characters.

patient — Character of length 1, name of the patient ID variable.

covariates — Character vector of names of other covariates.

missing — Character of length 1, name of an optional variable in a simulated dataset to indicate which outcome values should be missing. Set to NULL to omit.

reference_group — Character of length 1, Level of the group column to indicate the control group. reference_group only applies to the post-processing that happens in functions like brm_marginal_draws() downstream of the model. It does not control the fixed effect parameterization in the model matrix that brms derives from the formula from brm_formula().

level_control — Deprecated on 2024-01-11 (version 0.2.0.9002). Use reference_group instead.

reference_subgroup — Character of length 1, level of the subgroup column to use as a reference for pairwise differences in when computing marginal means downstream of the model. It does not control the fixed effect parameterization in the model matrix that brms derives from the formula from brm_formula().

reference_time — Character of length 1 or NULL, level of the time column to indicate the baseline time point. This value should not be present in the data if the outcome variable is change from baseline, but it must be in the data if the outcome variable is the raw response so that brms.mmrm can produce model-based marginal estimates of change from baseline. In other words, set reference_time to NULL if role is "change", and set reference_time to a non-null value in data[[time]] if role is "response".

Note: reference_time only applies to the post-processing that happens in functions like brm_marginal_draws() downstream of the model. It does not control the fixed effect parameterization in the model matrix that brms derives from the formula from brm_formula().

level_baseline — Deprecated on 2024-01-11 (version 0.2.0.9002). Use reference_time instead.

Value

A classed tibble with attributes which denote features of the data such as the treatment group and discrete time variables.

Preprocessing

The preprocessing steps in brm_data() are as follows:

- Perform basic assertions to make sure the data and other arguments are properly formatted.

- Convert the group and time columns to character vectors.
- Sanitize the levels of the group and time columns using `make.names(unique = FALSE, allow_ = TRUE)` to ensure agreement between the data and the output of `brms`.
- For each implicitly missing outcome observation, add explicit row with the outcome variable equal to `NA_real_`.
- Arrange the rows of the data by group, then patient, then discrete time.
- Select only the columns of the data relevant to an MMRM analysis.

Separation string

Post-processing in `brm_marginal_draws()` names each of the group-by-time marginal means with the delimiting character string from `Sys.getenv("BRM_SEP", unset = "|")`. Neither the column names nor element names of the group and time variables can contain this string. To set a custom string yourself, use `Sys.setenv(BRM_SEP = "YOUR_CUSTOM_STRING")`.

See Also

Other data: `brm_data_change()`

Examples

```
set.seed(0)
data <- brm_simulate_simple()$data
colnames(data) <- paste0("col_", colnames(data))
data
brm_data(
  data = data,
  outcome = "col_response",
  role = "response",
  group = "col_group",
  time = "col_time",
  patient = "col_patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
```

---

brm_data_change *Convert to change from baseline.*

Description

Convert a dataset from raw response to change from baseline.

Usage

```
brm_data_change(data, name_change = "change", name_baseline = "baseline")
```

Arguments

data	A classed tibble (e.g. from brm_data()) with raw response as the outcome variable (role = "response" in brm_data()).
name_change	Character of length 1, name of the new outcome column for change from baseline.
name_baseline	Character of length 1, name of the new column for the original baseline response.

Value

A classed tibble with change from baseline as the outcome variable and the internal attributes modified accordingly. A special baseline column is also created, and the original raw response column is removed. The new baseline column is comprised of the elements of the response variable corresponding to the reference_time argument of brm_data().

If there is a column to denote missing values for simulation purposes, e.g. the "missing" column generated by brm_simulate_outline(), then missing baseline values are propagated accordingly such that change from baseline will be missing if either the post-baseline response is missing or the baseline response is missing.

See Also

Other data: brm_data()

Examples

```
set.seed(0)
data <- brm_data(
  data = dplyr::rename(brm_simulate_simple()$data, y_values = response),
  outcome = "y_values",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
data
attr(data, "brm_role")
attr(data, "brm_outcome")
attr(data, "brm_baseline")
attr(data, "brm_reference_time")
changed <- brm_data_change(data = data, name_change = "delta")
changed
attr(changed, "brm_role")
attr(changed, "brm_outcome")
attr(changed, "brm_baseline")
attr(data, "brm_reference_time")
```

brm_formula *Model formula*

Description

Build a model formula for an MMRM.

Usage

```
brm_formula(
  data,
  intercept = TRUE,
  baseline = !is.null(attr(data, "brm_baseline")),
 baseline_subgroup = !is.null(attr(data, "brm_baseline")) && !is.null(attr(data,
    "brm_subgroup")),
 baseline_subgroup_time = !is.null(attr(data, "brm_baseline")) && !is.null(attr(data,
    "brm_subgroup")),
  baseline_time = !is.null(attr(data, "brm_baseline")),
  group = TRUE,
  group_subgroup = !is.null(attr(data, "brm_subgroup")),
  group_subgroup_time = !is.null(attr(data, "brm_subgroup")),
  group_time = TRUE,
  subgroup = !is.null(attr(data, "brm_subgroup")),
  subgroup_time = !is.null(attr(data, "brm_subgroup")),
  time = TRUE,
  correlation = "unstructured",
  effect_baseline = NULL,
  effect_group = NULL,
  effect_time = NULL,
  interaction_baseline = NULL,
  interaction_group = NULL
)
```

Arguments

data	A classed data frame from brm_data().
intercept	Logical of length 1. TRUE (default) to include an intercept, FALSE to omit.
baseline	Logical of length 1. TRUE to include an additive effect for baseline response, FALSE to omit. Default is TRUE if brm_data() previously declared a baseline variable in the dataset.
baseline_subgroup	Logical of length 1. TRUE to include baseline-by-subgroup interaction, FALSE to omit. Default is TRUE if brm_data() previously declared baseline and subgroup variables in the dataset.

`baseline_subgroup_time`
: Logical of length 1. `TRUE` to include baseline-by-subgroup-by-time interaction, `FALSE` to omit. Default is `TRUE` if `brm_data()` previously declared baseline and subgroup variables in the dataset.

`baseline_time`
: Logical of length 1. `TRUE` to include baseline-by-time interaction, `FALSE` to omit. Default is `TRUE` if `brm_data()` previously declared a baseline variable in the dataset.

`group`
: Logical of length 1. `TRUE` (default) to include additive effects for treatment groups, `FALSE` to omit.

`group_subgroup`
: Logical of length 1. `TRUE` to include group-by-subgroup interaction, `FALSE` to omit. Default is `TRUE` if `brm_data()` previously declared a subgroup variable in the dataset.

`group_subgroup_time`
: Logical of length 1. `TRUE` to include group-by-subgroup-by-time interaction, `FALSE` to omit. Default is `TRUE` if `brm_data()` previously declared a subgroup variable in the dataset.

`group_time`
: Logical of length 1. `TRUE` (default) to include group-by-time interaction, `FALSE` to omit.

`subgroup`
: Logical of length 1. `TRUE` to include additive fixed effects for subgroup levels, `FALSE` to omit. Default is `TRUE` if `brm_data()` previously declared a subgroup variable in the dataset.

`subgroup_time`
: Logical of length 1. `TRUE` to include subgroup-by-time interaction, `FALSE` to omit. Default is `TRUE` if `brm_data()` previously declared a subgroup variable in the dataset.

`time`
: Logical of length 1. `TRUE` (default) to include a additive effect for discrete time, `FALSE` to omit.

`correlation`
: Character of length 1, name of the correlation structure. Only `"unstructured"` is currently supported.

`effect_baseline`
: Deprecated on 2024-01-16 (version 0.0.2.9002). Use `baseline` instead.

`effect_group`
: Deprecated on 2024-01-16 (version 0.0.2.9002). Use `group` instead.

`effect_time`
: Deprecated on 2024-01-16 (version 0.0.2.9002). Use `time` instead.

`interaction_baseline`
: Deprecated on 2024-01-16 (version 0.0.2.9002). Use `baseline_time` instead.

`interaction_group`
: Deprecated on 2024-01-16 (version 0.0.2.9002). Use `group_time` instead.

Details

`brm_formula()` builds an R formula for an MMRM based on the details in the data and your choice of parameterization. Customize your parameterization by toggling on or off the various `TRUE`/`FALSE` arguments of `brm_formula()`, such as `intercept`, `baseline`, and `group_time`. All plausible additive effects, two-way interactions, and three-way interactions can be specified. The following interactions are not supported:

- Any interactions with the concomitant covariates you specified in the `covariates` argument of `brm_data()`.
- Any interactions which include baseline response and treatment group together. Rationale: in a randomized controlled experiment, baseline and treatment group assignment should be uncorrelated.

Value

An object of class `"brmsformula"` returned from `brms::brmsformula()`. It contains the fixed effect parameterization, correlation structure, and residual variance structure.

Parameterization

The formula is not the only factor that determines the fixed effect parameterization. The ordering of the categorical variables in the data, as well as the `contrast` option in R, affect the construction of the model matrix. To see the model matrix that will ultimately be used in `brm_model()`, run `brms::make_standata()` and examine the `X` element of the returned list. See the examples below for a demonstration.

See Also

Other models: `brm_model()`

Examples

```
set.seed(0)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
brm_formula(data)
brm_formula(data = data, intercept = FALSE, baseline = FALSE)
formula <- brm_formula(
  data = data,
  intercept = FALSE,
  baseline = FALSE,
  group = FALSE
)
formula
# Optional: set the contrast option, which determines the model matrix.
options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
# See the fixed effect parameterization you get from the data:
head(brms::make_standata(formula = formula, data = data)$X)
# Specify a different contrast method to use an alternative
# parameterization when fitting the model with brm_model():
```

```
options(
  contrasts = c(unordered = "contr.treatment", ordered = "contr.poly")
)
# different model matrix than before:
head(brms::make_standata(formula = formula, data = data)$X)
```

`brm_marginal_data` *Marginal summaries of the data.*

Description

Marginal summaries of the data.

Usage

```
brm_marginal_data(
  data,
  level = 0.95,
  use_subgroup = !is.null(attr(data, "brm_subgroup"))
)
```

Arguments

`data`	A classed data frame from `brm_data()`.
`level`	Numeric of length 1 from 0 to 1, level of the confidence intervals.
`use_subgroup`	Logical of length 1, whether to summarize the data by each subgroup level.

Value

A tibble with one row per summary statistic and the following columns:

- `group`: treatment group.
- `subgroup`: subgroup level. Only included if the `subgroup` argument of `brm_marginal_data()` is `TRUE`.
- `time`: discrete time point.
- `statistic`: type of summary statistic.
- `value`: numeric value of the estimate.

The `statistic` column has the following possible values:

- `mean`: observed mean response after removing missing values.
- `median`: observed median response after removing missing values.
- `sd`: observed standard deviation of the response after removing missing values.
- `lower`: lower bound of a normal equal-tailed confidence interval with confidence level determined by the `level` argument.

- `upper`: upper bound of a normal equal-tailed confidence interval with confidence level determined by the `level` argument.
- `n_observe`: number of non-missing values in the response.
- `n_total`: number of total records in the data for the given group/time combination, including both observed and missing values.

See Also

Other marginals: `brm_marginal_draws()`, `brm_marginal_draws_average()`, `brm_marginal_probabilities()`, `brm_marginal_summaries()`

Examples

```
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
brm_marginal_data(data = data)
```

---

brm_marginal_draws *MCMC draws from the marginal posterior of an MMRM*

Description

Get marginal posterior draws from a fitted MMRM.

Usage

```
brm_marginal_draws(
  model,
  data,
  use_subgroup = !is.null(attr(data, "brm_subgroup")),
  control = NULL,
  baseline = NULL
)
```

Arguments

model	Fitted brms model object from brm_model().
data	Classed tibble with preprocessed data from brm_data().
use_subgroup	Logical of length 1, whether to summarize the draws by each subgroup level. If TRUE, subgroup-specific marginals are given. Otherwise, the subgroup is marginalized out.
control	Deprecated. Set the control group level in brm_data().
baseline	Deprecated. Set the control group level in brm_data().

Value

A named list of tibbles of MCMC draws of the marginal posterior distribution of each treatment group and time point (or group-by-subgroup-by-time, if applicable). In each tibble, there is 1 row per posterior sample and one column for each type of marginal distribution (i.e. each combination of treatment group and discrete time point. The specific tibbles in the returned list are described below:

- response: on the scale of the response variable.
- difference_time: change from baseline: the response at a particular time minus the response at baseline (reference_time). Only returned if the role argument of brm_data() was "response". (If role is "change", then response already represents change from baseline.)
- difference_group: treatment effect: the the difference_time at each active group minus the difference_time at the control group (reference_group). If role is "change", then treatment group is instead the difference between response at each active group minus the response at the control group.
- difference_subgroup: subgroup differences: the difference_group at each subgroup level minus the difference_group at the subgroup reference level (reference_subgroup).

Separation string

Post-processing in brm_marginal_draws() names each of the group-by-time marginal means with the delimiting character string from Sys.getenv("BRM_SEP", unset = "|"). Neither the column names nor element names of the group and time variables can contain this string. To set a custom string yourself, use Sys.setenv(BRM_SEP = "YOUR_CUSTOM_STRING").

See Also

Other marginals: brm_marginal_data(), brm_marginal_draws_average(), brm_marginal_probabilities(), brm_marginal_summaries()

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
```

```
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      model <- brm_model(
        data = data,
        formula = formula,
        chains = 1,
        iter = 100,
        refresh = 0
      )
    )
  )
)
brm_marginal_draws(model = model, data = data)
}
```

`brm_marginal_draws_average` *Average marginal MCMC draws across time points.*

Description

Simple un-weighted arithmetic mean of marginal MCMC draws across time points.

Usage

```
brm_marginal_draws_average(draws, data, times = NULL, label = "average")
```

Arguments

`draws`	List of posterior draws from `brm_marginal_draws()`.
`data`	Classed tibble with preprocessed data from `brm_data()`.
`times`	Character vector of discrete time point levels over which to average the MCMC samples within treatment group levels. Set to `NULL` to average across all time points. Levels are automatically sanitized with `make.names(unique = FALSE, allow_ = TRUE)` to ensure agreement with `brms` variable names in downstream computations.

`label` Character of length 1, time point label for the averages. Automatically sanitized with `make.names(unique = FALSE, allow_ = TRUE)`. Must not conflict with any existing time point labels in the data after the label and time points are sanitized.

Value

A named list of tibbles of MCMC draws of the marginal posterior distribution of each treatment group and time point (or group-by-subgroup-by-time, if applicable). In each tibble, there is 1 row per posterior sample sand one column for each type of marginal distribution (i.e. each combination of treatment group and discrete time point. The specific `tibbles` in the returned list are described below:

- `response`: on the scale of the response variable.
- `difference_time`: change from baseline: the `response` at a particular time minus the `response` at baseline (`reference_time`). Only returned if the `role` argument of `brm_data()` was `"response"`. (If `role` is `"change"`, then `response` already represents change from baseline.)
- `difference_group`: treatment effect: the the `difference_time` at each active group minus the `difference_time` at the control group (`reference_group`). If `role` is `"change"`, then treatment group is instead the difference between `response` at each active group minus the `response` at the control group.
- `difference_subgroup`: subgroup differences: the `difference_group` at each subgroup level minus the `difference_group` at the subgroup reference level (`reference_subgroup`).

Separation string

Post-processing in `brm_marginal_draws()` names each of the group-by-time marginal means with the delimiting character string from `Sys.getenv("BRM_SEP", unset = "|")`. Neither the column names nor element names of the group and time variables can contain this string. To set a custom string yourself, use `Sys.setenv(BRM_SEP = "YOUR_CUSTOM_STRING")`.

See Also

Other marginals: `brm_marginal_data()`, `brm_marginal_draws()`, `brm_marginal_probabilities()`, `brm_marginal_summaries()`

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
```

```
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      model <- brm_model(
        data = data,
        formula = formula,
        chains = 1,
        iter = 100,
        refresh = 0
      )
    )
  )
)
draws <- brm_marginal_draws(model = model, data = data)
brm_marginal_draws_average(draws = draws, data = data)
brm_marginal_draws_average(
  draws = draws,
  data = data,
  times = c("time_1", "time_2"),
  label = "mean"
)
}
```

`brm_marginal_probabilities` *Marginal probabilities on the treatment effect for an MMRM.*

Description

Marginal probabilities on the treatment effect for an MMRM.

Usage

```
brm_marginal_probabilities(draws, direction = "greater", threshold = 0)
```

Arguments

`draws`	Posterior draws of the marginal posterior obtained from `brm_marginal_draws()`.
`direction`	Character vector of the same length as `threshold`. `"greater"` to compute the marginal posterior probability that the treatment effect is greater than the threshold, `"less"` to compute the marginal posterior probability that the treatment effect is less than the threshold. Each element `direction[i]` corresponds to `threshold[i]` for all `i` from 1 to `length(direction)`.

threshold Numeric vector of the same length as direction, treatment effect threshold for computing posterior probabilities. Each element direction[i] corresponds to threshold[i] for all i from 1 to length(direction).

Value

A tibble of probabilities of the form Prob(treatment effect > threshold | data) and/or Prob(treatment effect < threshold | data). It has one row per probability and the following columns: * group: treatment group. * subgroup: subgroup level, if applicable. * time: discrete time point, * direction: direction of the comparison in the marginal probability: "greater" for >, "less" for < * threshold: treatment effect threshold in the probability statement. * value: numeric value of the estimate of the probability.

See Also

Other marginals: brm_marginal_data(), brm_marginal_draws(), brm_marginal_draws_average(), brm_marginal_summaries()

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      model <- brm_model(
        data = data,
        formula = formula,
        chains = 1,
        iter = 100,
        refresh = 0
      )
    )
  )
)
draws <- brm_marginal_draws(model = model, data = data)
brm_marginal_probabilities(draws, direction = "greater", threshold = 0)
```

```
}
```

`brm_marginal_summaries` *Summary statistics of the marginal posterior of an MMRM.*

Description

Summary statistics of the marginal posterior of an MMRM.

Usage

```
brm_marginal_summaries(draws, level = 0.95)
```

Arguments

`draws`	Posterior draws of the marginal posterior obtained from `brm_marginal_draws()`.
`level`	Numeric of length 1 between 0 and 1, credible level for the credible intervals.

Value

A tibble with one row per summary statistic and the following columns:

- `marginal`: type of marginal distribution. If `outcome` was `"response"` in `brm_marginal_draws()`, then possible values include `"response"` for the response on the raw scale, `"change"` for change from baseline, and `"difference"` for treatment difference in terms of change from baseline. If `outcome` was `"change"`, then possible values include `"response"` for the response one the change from baseline scale and `"difference"` for treatment difference.
- `statistic`: type of summary statistic. `"lower"` and `"upper"` are bounds of an equal-tailed quantile-based credible interval.
- `group`: treatment group.
- `subgroup`: subgroup level, if applicable.
- `time`: discrete time point.
- `value`: numeric value of the estimate.
- `mcse`: Monte Carlo standard error of the estimate. The `statistic` column has the following possible values:
- `mean`: posterior mean.
- `median`: posterior median.
- `sd`: posterior standard deviation of the mean.
- `lower`: lower bound of an equal-tailed credible interval of the mean, with credible level determined by the `level` argument.
- `upper`: upper bound of an equal-tailed credible interval with credible level determined by the `level` argument.

See Also

Other marginals: brm_marginal_data(), brm_marginal_draws(), brm_marginal_draws_average(), brm_marginal_probabilities()

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      model <- brm_model(
        data = data,
        formula = formula,
        chains = 1,
        iter = 100,
        refresh = 0
      )
    )
  )
)
draws <- brm_marginal_draws(model = model, data = data)
suppressWarnings(brm_marginal_summaries(draws))
}
```

---

brm_model *Basic MMRM*

Description

Fit a basic MMRM model using brms.

Usage

```
brm_model(data, formula, prior = NULL, ...)
```

Arguments

data	A tidy data frame with one row per patient per discrete time point.
formula	An object of class "brmsformula" from brm_formula() or brms::brmsformula(). Should include the full parameterization of the model, including fixed effects, residual correlation, and heterogeneity in the discrete-time-specific residual variance components.
prior	Either NULL for default priors or a "brmsprior" object from brms::prior().
...	Arguments to brms::brm() other than data, formula, and prior.

Value

A fitted model object from brms.

Parameterization

The formula is not the only factor that determines the fixed effect parameterization. The ordering of the categorical variables in the data, as well as the contrast option in R, affect the construction of the model matrix. To see the model matrix that will ultimately be used in brm_model(), run brms::make_standata() and examine the X element of the returned list. See the examples below for a demonstration.

See Also

Other models: brm_formula()

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
# Optional: set the contrast option, which determines the model matrix.
options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
# See the fixed effect parameterization you get from the data:
head(brms::make_standata(formula = formula, data = data)$X)
# Specify a different contrast method to use an alternative
# parameterization when fitting the model with brm_model():
```

```
options(
  contrasts = c(unordered = "contr.treatment", ordered = "contr.poly")
)
# different model matrix than before:
head(brms::make_standata(formula = formula, data = data)$X)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      model <- brm_model(
        data = data,
        formula = formula,
        chains = 1,
        iter = 100,
        refresh = 0
      )
    )
  )
)
# The output model is a brms model fit object.
model
# The `prior_summary()` function shows the full prior specification
# which reflects the fully realized fixed effects parameterization.
brms::prior_summary(model)
}
```

`brm_plot_compare` *Visually compare the marginals of multiple models and/or datasets.*

Description

Visually compare the marginals of multiple models and/or datasets.

Usage

```
brm_plot_compare(
  ...,
  marginal = "response",
  compare = "source",
  axis = "time",
  facet = c("group", "subgroup")
)
```

Arguments

`...`	Named `tibbles` of marginals posterior summaries from `brm_marginal_summaries()` and/or `brm_marginal_data()`.
`marginal`	Character of length 1, which kind of marginal to visualize. Must be a value in the `marginal` column of the supplied `tibbles` in the `...` argument. Only applies to MCMC output, the data is always on the scale of the response variable.

compare	Character of length 1 identifying the variable to display using back-to-back interval plots of different colors. This is the primary comparison of interest. Must be one of "source" (the source of the marginal summaries, e.g. a model or dataset), "time" or "group" (in the non-subgroup case). Can also be "subgroup" if the marginal summaries are subgroup-specific. The value must not be in axis or facet.
axis	Character of length 1 identifying the quantity to put on the horizontal axis. Must be be one of "source" (the source of the marginal summaries, e.g. a model or dataset), "time", or "group" (in the non-subgroup case). If the marginals are subgroup-specific, then axis can also be "subgroup". The value must not be in compare or facet.
facet	Character vector of length 1 or 2 with quantities to generate facets. Each element must be "source" (the source of the marginal summaries, e.g. a model or dataset), "time", "group", or "subgroup", and c(axis, facet) must all have unique elements. "subgroup" is automatically removed if the marginals have no subgroup. If facet has length 1, then faceting is wrapped. If facet has length 2, then faceting is in a grid, and the first element is horizontal facet.

Details

By default, brm_plot_compare() compares multiple models and/or datasets side-by-side. The compare argument selects the primary comparison of interest, and arguments axis and facet control the arrangement of various other components of the plot.

Value

A ggplot object.

See Also

Other visualization: brm_plot_draws()

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
```

```
)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      model <- brm_model(
        data = data,
        formula = formula,
        chains = 1,
        iter = 100,
        refresh = 0
      )
    )
  )
)
draws <- brm_marginal_draws(model = model, data = data)
suppressWarnings(summaries_draws <- brm_marginal_summaries(draws))
summaries_data <- brm_marginal_data(data)
brm_plot_compare(
  model1 = summaries_draws,
  model2 = summaries_draws,
  data = summaries_data
)
brm_plot_compare(
  model1 = summaries_draws,
  model2 = summaries_draws,
  marginal = "difference"
)
}
```

`brm_plot_draws` *Visualize posterior draws of marginals.*

Description

Visualize posterior draws of marginals.

Usage

```
brm_plot_draws(draws, axis = "time", facet = c("group", "subgroup"))
```

Arguments

`draws`	A data frame of draws from an element of the output list of `brm_marginal_summaries()`.
`axis`	Character of length 1 identifying the quantity to put on the horizontal axis. Must be be one of `"time"` or `"group"` if the marginal summaries are not subgroup-specific. If the marginals are subgroup-specific, then `axis` must be one of `"time"`, `"group"`, or `"subgroup"`.

facet — Character vector of length 1 or 2 with quantities to generate facets. Each element must be "time", "group", or "subgroup", and c(axis, facet) must all have unique elements. "subgroup" is automatically removed if the marginals have no subgroup. If facet has length 1, then faceting is wrapped. If facet has length 2, then faceting is in a grid, and the first element is horizontal facet.

Value

A ggplot object.

See Also

Other visualization: brm_plot_compare()

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_data(
  data = brm_simulate_simple()$data,
  outcome = "response",
  role = "response",
  group = "group",
  time = "time",
  patient = "patient",
  reference_group = "group_1",
  reference_time = "time_1"
)
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      model <- brm_model(
        data = data,
        formula = formula,
        chains = 1,
        iter = 100,
        refresh = 0
      )
    )
  )
)
draws <- brm_marginal_draws(model = model, data = data)
brm_plot_draws(draws = draws$difference_time)
}
```

brm_prior_simple *Simple prior for a* brms *MMRM*

Description

Generate a simple prior for a brms MMRM.

Usage

```
brm_prior_simple(
  data,
  formula,
  intercept = "student_t(3, 0, 2.5)",
  coefficients = "student_t(3, 0, 2.5)",
  sigma = "student_t(3, 0, 2.5)",
  correlation = "lkj(1)"
)
```

Arguments

data	A tidy data frame with one row per patient per discrete time point.
formula	An object of class "brmsformula" from brm_formula() or brms::brmsformula(). Should include the full parameterization of the model, including fixed effects, residual correlation, and heterogeneity in the discrete-time-specific residual variance components.
intercept	Character of length 1, Stan code for the prior to set on the intercept parameter.
coefficients	Character of length 1, Stan code for the prior to set independently on each of the non-intercept model coefficients.
sigma	Character of length 1, Stan code for the prior to set independently on each of the log-scale standard deviation parameters. Should be a symmetric prior in most situations.
correlation	Character of length 1, Stan code for the prior on the correlation matrix for the residuals of a given patient. (Different patients are modeled as independent, and each patient has the same correlation structure as each other patient.) Should be an LKJ prior in most situations.

Details

In brm_prior_simple(), you can separately choose priors for the intercept, model coefficients, log-scale standard deviations, and pairwise correlations between time points within patients. However, each class of parameters is set as a whole. In other words, brm_prior_simple() cannot assign different priors to different fixed effect parameters.

Value

A classed data frame with the brms prior.

Examples

```
set.seed(0L)
data <- brm_simulate_outline()
data <- brm_simulate_continuous(data, names = c("age", "biomarker"))
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
brm_prior_simple(
  data = data,
  formula = formula,
  intercept = "student_t(3, 0, 2.5)",
  coefficients = "normal(0, 10)",
  sigma = "student_t(2, 0, 4)",
  correlation = "lkj(2.5)"
)
```

---

brm_simulate_categorical

Append simulated categorical covariates

Description

Simulate and append non-time-varying categorical covariates to an existing `brm_data()` dataset.

Usage

```
brm_simulate_categorical(data, names, levels, probabilities = NULL)
```

Arguments

`data`	Classed `tibble` as from `brm_data()` or `brm_simulate_outline()`.
`names`	Character vector with the names of the new covariates to simulate and append. Names must all be unique and must not already be column names of `data`.
`levels`	Character vector of unique levels of the simulated categorical covariates.
`probabilities`	Either `NULL` or a numeric vector of length `length(levels)` with levels between 0 and 1 where all elements sum to 1. If `NULL`, then all levels are equally likely to be drawn. If not `NULL`, then `probabilities` is a vector of sampling probabilities corresponding to each respective level of `levels`.

Details

Each covariate is a new column of the dataset with one independent random categorical draw for each patient, using a fixed set of levels (via `base::sample()` with `replace = TRUE`). All covariates simulated this way are independent of everything else in the data, including other covariates (to the extent that the random number generators in R work as intended).

Value

A classed tibble, like from brm_data() or brm_simulate_outline(), but with new categorical covariate columns and with the names of the new covariates appended to the brm_covariates attribute. Each new categorical covariate column is a character vector, not the factor type in base R.

See Also

Other simulation: brm_simulate_continuous(), brm_simulate_outline(), brm_simulate_prior(), brm_simulate_simple()

Examples

```
data <- brm_simulate_outline()
brm_simulate_categorical(
  data = data,
  names = c("site", "region"),
  levels = c("area1", "area2")
)
brm_simulate_categorical(
  data = data,
  names = c("site", "region"),
  levels = c("area1", "area2"),
  probabilities = c(0.1, 0.9)
)
```

---

brm_simulate_continuous

Append simulated continuous covariates

Description

Simulate and append non-time-varying continuous covariates to an existing brm_data() dataset.

Usage

```
brm_simulate_continuous(data, names, mean = 0, sd = 1)
```

Arguments

data	Classed tibble as from brm_data() or brm_simulate_outline().
names	Character vector with the names of the new covariates to simulate and append. Names must all be unique and must not already be column names of data.
mean	Numeric of length 1, mean of the normal distribution for simulating each covariate.
sd	Positive numeric of length 1, standard deviation of the normal distribution for simulating each covariate.

Details

Each covariate is a new column of the dataset with one independent random univariate normal draw for each patient. All covariates simulated this way are independent of everything else in the data, including other covariates (to the extent that the random number generators in R work as intended).

Value

A classed `tibble`, like from `brm_data()` or `brm_simulate_outline()`, but with new numeric covariate columns and with the names of the new covariates appended to the `brm_covariates` attribute.

See Also

Other simulation: `brm_simulate_categorical()`, `brm_simulate_outline()`, `brm_simulate_prior()`, `brm_simulate_simple()`

Examples

```
data <- brm_simulate_outline()
brm_simulate_continuous(
  data = data,
  names = c("age", "biomarker")
)
brm_simulate_continuous(
  data = data,
  names = c("biomarker1", "biomarker2"),
  mean = 1000,
  sd = 100
)
```

---

brm_simulate_outline *Start a simulated dataset*

Description

Begin creating a simulated dataset.

Usage

```
brm_simulate_outline(
  n_group = 2L,
  n_subgroup = NULL,
  n_patient = 100L,
  n_time = 4L,
  rate_dropout = 0.1,
  rate_lapse = 0.05
)
```

Arguments

n_group	Positive integer of length 1, number of treatment groups.
n_subgroup	Positive integer of length 1, number of subgroup levels. Set to NULL to omit the subgroup entirely.
n_patient	Positive integer of length 1. If n_subgroup is NULL, then n_patient is the number of patients per treatment group. Otherwise, n_patient is the number of patients per treatment group *per subgroup*. In both cases, the total number of patients in the whole simulated dataset is usually much greater than the n_patients argument of brm_simulate_outline().
n_time	Positive integer of length 1, number of discrete time points (e.g. scheduled study visits) per patient.
rate_dropout	Numeric of length 1 between 0 and 1, post-baseline dropout rate. A dropout is an intercurrent event when data collection for a patient stops permanently, causing the outcomes for that patient to be missing during and after the dropout occurred. The first time point is assumed to be baseline, so dropout is there. Dropouts are equally likely to occur at each of the post-baseline time points.
rate_lapse	Numeric of length 1, expected proportion of post-baseline outcomes that are missing. Missing outcomes of this type are independent and uniformly distributed across the data.

Value

A data frame from brm_data() with attributes to define roles for various columns in the dataset. The data frame has one row per patient per time point and the following columns:

- group: integer index of the treatment group.
- patient: integer index of the patient.
- time: integer index of the discrete time point.

See Also

Other simulation: brm_simulate_categorical(), brm_simulate_continuous(), brm_simulate_prior(), brm_simulate_simple()

Examples

```
brm_simulate_outline()
```

`brm_simulate_prior` *Prior predictive draws.*

Description

Simulate the outcome variable from the prior predictive distribution of an MMRM using `brms`.

Usage

```
brm_simulate_prior(
  data,
  formula,
  prior = brms.mmrm::brm_prior_simple(data = data, formula = formula),
  ...
)
```

Arguments

`data`	A tidy data frame with one row per patient per discrete time point.
`formula`	An object of class `"brmsformula"` from `brm_formula()` or `brms::brmsformula()`. Should include the full parameterization of the model, including fixed effects, residual correlation, and heterogeneity in the discrete-time-specific residual variance components.
`prior`	A valid `brms` prior object with proper priors for parameters `b` (model coefficients), `b_sigma` (log residual standard deviations for each time point), and `cortime` (residual correlations among time points within patients). See the `brm_prior_simple()` function for an example.
`...`	Arguments to `brms::brm()` other than `data`, `formula`, and `prior`.

Details

`brm_simulate_prior()` calls `brms::brm()` with `sample_prior = "only"`, which sets the default intercept prior using the outcome variable and requires at least some elements of the outcome variable to be non-missing in advance. So to provide feasible and consistent output, `brm_simulate_prior()` temporarily sets the outcome variable to all zeros before invoking `brms::brm()`.

Value

A list with the following elements:

- `data`: a classed `tibble` with the outcome variable simulated as a draw from the prior predictive distribution (the final row of `outcome` in the output). If you simulated a missingness pattern with `brm_simulate_outline()`, then that missingness pattern is applied so that the appropriate values of the outcome variable are set to `NA`.
- `model`: the `brms` model fit object.
- `model_matrix`: the model matrix of the fixed effects, obtained from `brms::make_standata()`.

- `outcome`: a numeric matrix with one column per row of `data` and one row per saved prior predictive draw.
- `parameters`: a `tibble` of saved parameter draws from the prior predictive distribution.

See Also

Other simulation: `brm_simulate_categorical()`, `brm_simulate_continuous()`, `brm_simulate_outline()`, `brm_simulate_simple()`

Examples

```
if (identical(Sys.getenv("BRM_EXAMPLES", unset = ""), "true")) {
set.seed(0L)
data <- brm_simulate_outline()
data <- brm_simulate_continuous(data, names = c("age", "biomarker"))
formula <- brm_formula(
  data = data,
  baseline = FALSE,
  baseline_time = FALSE
)
tmp <- utils::capture.output(
  suppressMessages(
    suppressWarnings(
      out <- brm_simulate_prior(
        data = data,
        formula = formula
      )
    )
  )
)
out$data
}
```

`brm_simulate_simple` *Simple MMRM simulation.*

Description

Simple function to simulate a dataset from a simple specialized MMRM.

Usage

```
brm_simulate_simple(
  n_group = 2L,
  n_patient = 100L,
  n_time = 4L,
  hyper_beta = 1,
  hyper_tau = 0.1,
  hyper_lambda = 1
)
```

Arguments

n_group	Positive integer of length 1, number of treatment groups.
n_patient	Positive integer of length 1, number of patients per treatment group.
n_time	Positive integer of length 1, number of discrete time points (e.g. scheduled study visits) per patient.
hyper_beta	Positive numeric of length 1, hyperparameter. Prior standard deviation of the fixed effect parameters beta.
hyper_tau	Positive numeric of length 1, hyperparameter. Prior standard deviation parameter of the residual log standard deviation parameters tau
hyper_lambda	Positive numeric of length 1, hyperparameter. Prior shape parameter of the LKJ correlation matrix of the residuals among discrete time points.

Details

Refer to the methods vignette for a full model specification. The brm_simulate_simple() function simulates a dataset from a simple pre-defined MMRM. It assumes a cell means structure for fixed effects, which means there is one fixed effect scalar parameter (element of vector beta) for each unique combination of levels of treatment group and discrete time point. The elements of beta have independent univariate normal priors with mean 0 and standard deviation hyper_beta. The residual log standard deviation parameters (elements of vector tau) have normal priors with mean 0 and standard deviation hyper_tau. The residual correlation matrix parameter lambda has an LKJ correlation prior with shape parameter hyper_lambda.

Value

A list of three objects:

- data: A tidy dataset with one row per patient per discrete time point and columns for the outcome and ID variables.
- model_matrix: A matrix with one row per row of data and columns that represent levels of the covariates.
- parameters: A named list of parameter draws sampled from the prior:
  - beta: numeric vector of fixed effects.
  - tau: numeric vector of residual log standard parameters for each time point.
  - sigma: numeric vector of residual standard parameters for each time point. sigma is equal to exp(tau).
  - lambda: correlation matrix of the residuals among the time points within each patient.
  - covariance: covariance matrix of the residuals among the time points within each patient. covariance is equal to diag(sigma) %*% lambda %*% diag(sigma).

See Also

Other simulation: brm_simulate_categorical(), brm_simulate_continuous(), brm_simulate_outline(), brm_simulate_prior()

Examples

```
set.seed(0L)
simulation <- brm_simulate_simple()
simulation$data
```

Index