# Package: historicalborrow (via r-universe)

October 11, 2024

**Title** Non-Longitudinal Bayesian Historical Borrowing Models

**Description** Historical borrowing in clinical trials can improve precision and operating characteristics. This package supports a hierarchical model and a mixture model to borrow historical control data from other studies to better characterize the control response of the current study. It also quantifies the amount of borrowing through benchmark models (independent and pooled). Some of the methods are discussed by Viele et al. (2013) <doi:10.1002/pst.1589>.

**Version** 1.1.0

**License** MIT + file LICENSE

**URL** https://wlandau.github.io/historicalborrow/, https://github.com/wlandau/historicalborrow

**BugReports** https://github.com/wlandau/historicalborrow/issues

**Depends** R (>= 3.5.0)

**Imports** dplyr, ggplot2, Matrix, posterior, rjags, rlang, stats, tibble, tidyr, tidyselect, utils

**Suggests** knitr, markdown, rmarkdown, testthat (>= 3.0.0)

**Encoding** UTF-8

**Language** en-US

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** William Michael Landau [aut, cre] (<https://orcid.org/0000-0003-1878-3253>), Albert Man [rev], Eli Lilly and Company [cph]

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

**Repository** CRAN

**Date/Publication** 2024-09-10 15:00:08 UTC

# Contents



---

`historicalborrow-package` *historicalborrow: Bayesian historical borrowing models for clinical studies.*

---

## Description

Bayesian historical borrowing models for clinical studies.

---

`hb_convergence` *Check convergence diagnostics*

---

## Description

Check the convergence diagnostics on a model.

## Usage

```
hb_convergence(mcmc)
```

## Arguments

| | |
|---|---|
| `mcmc` | A wide data frame of posterior samples returned by `hb_mcmc_hierarchical()` or similar MCMC function. |

## Value

A data frame of summarized convergence diagnostics. `max_rhat` is the maximum univariate Gelman/Rubin potential scale reduction factor over all the parameters of the model, `min_ess_bulk` is the minimum bulk effective sample size over the parameters, and `min_ess_tail` is the minimum tail effective sample size. `max_rhat` should be below 1.01, and the ESS metrics should both be above 100 times the number of MCMC chains. If any of these conditions are not true, the MCMC did not converge, and it is recommended to try running the model for more saved iterations (and if `max_rhat` is high, possibly more warmup iterations).

## See Also

Other mcmc: `hb_mcmc_hierarchical()`, `hb_mcmc_independent()`, `hb_mcmc_mixture()`, `hb_mcmc_mixture_hyperparam`
`hb_mcmc_pool()`

## Examples

```
data <- hb_sim_pool(n_continuous = 2)$data
mcmc <- hb_mcmc_pool(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 200,
  n_iterations = 200
)
hb_convergence(mcmc)
```

---

# hb_data *Standardize data*

---

## Description

Standardize a tidy input dataset.

## Usage

```
hb_data(
  data,
  response,
  study,
  study_reference,
  group,
  group_reference,
  patient,
  covariates
)
```

## Arguments

| | |
|---|---|
| data | A tidy data frame or tibble with the data. |
| response | Character of length 1, name of the column in data with the response/outcome variable. data[[response]] must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| study | Character of length 1, name of the column in data with the study ID. |
| study_reference | Atomic of length 1, element of the study column that indicates the current study. (The other studies are historical studies.) |
| group | Character of length 1, name of the column in data with the group ID. |
| group_reference | Atomic of length 1, element of the group column that indicates the control group. (The other groups may be treatment groups.) |
| patient | Character of length 1, name of the column in data with the patient ID. |
| covariates | Character vector of column names in data with the columns with baseline covariates. These can be continuous, categorical, or binary. Regardless, historicalborrow derives the appropriate model matrix. |

## Details

Users do not normally need to call this function. It mainly serves exposes the indexing behavior of studies and group levels to aid in interpreting summary tables.

## Value

A standardized tidy data frame with one row per patient and the following columns:

- response: continuous response/outcome variable. (Should be change from baseline of an outcome of interest.)
- study_label: human-readable label of the study.
- study: integer study index with the max index equal to the current study (at study_reference).
- group_label: human-readable group label (e.g. treatment arm name).
- group: integer group index with an index of 1 equal to the control group (at group_reference).
- patient_label: original patient ID.
- patient: integer patient index.
- covariate_*: baseline covariate columns.

## Examples

```
data <- hb_sim_independent(n_continuous = 1, n_study = 2)$data
data <- dplyr::select(
  data,
  study,
  group,
  patient,
  response,
  tidyselect::everything()
)
colnames(data) <- c("trial", "arm", "subject", "change", "cov1", "cov2")
data$trial <- paste0("trial", data$trial)
data$arm <- paste0("arm", data$arm)
hb_data(
  data = data,
  response = "change",
  study = "trial",
  study_reference = "trial1",
  group = "arm",
  group_reference = "arm1",
  patient = "subject",
  covariates = c("cov1", "cov2")
)
```

---

# hb_ess *Effective sample size (ESS)*

---

## Description

Quantify borrowing with effective sample size (ESS) as cited and explained in the methods vignette at https://wlandau.github.io/historicalborrow/articles/methods.html.

## Usage

```
hb_ess(
  mcmc_pool,
  mcmc_hierarchical,
  data,
  response = "response",
  study = "study",
  study_reference = max(data[[study]]),
  group = "group",
  group_reference = min(data[[group]]),
  patient = "patient"
)
```

## Arguments

| | |
|---|---|
| mcmc_pool | A fitted model from hb_mcmc_pool(). |
| mcmc_hierarchical | A fitted model from hb_mcmc_hierarchical(). |
| data | A tidy data frame or tibble with the data. |
| response | Character of length 1, name of the column in data with the response/outcome variable. data[[response]] must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| study | Character of length 1, name of the column in data with the study ID. |
| study_reference | Atomic of length 1, element of the study column that indicates the current study. (The other studies are historical studies.) |
| group | Character of length 1, name of the column in data with the group ID. |
| group_reference | Atomic of length 1, element of the group column that indicates the control group. (The other groups may be treatment groups.) |
| patient | Character of length 1, name of the column in data with the patient ID. |

## Value

A data frame with one row and the following columns:

- v0: posterior predictive variance of the control group mean of a hypothetical new study given the pooled model. Calculated as the mean over MCMC samples of 1 / sum(sigma_i ^ 2), where each sigma_i is the residual standard deviation of study i estimated from the pooled model.
- v_tau: posterior predictive variance of a hypothetical new control group mean under the hierarchical model. Calculated by averaging over predictive draws, where each predictive draw is from rnorm(n = 1, mean = mu_, sd = tau_) and mu_ and tau_ are the mu and tau components of an MCMC sample.
- n: number of non-missing historical control patients.
- weight: strength of borrowing as a ratio of variances: v0 / v_tau.
- ess: strength of borrowing as an effective sample size: n v0 / v_tau, where n is the number of non-missing historical control patients.

## See Also

Other summary: hb_summary()

## Examples

```
data <- hb_sim_independent(n_continuous = 2)$data
data$group <- sprintf("group%s", data$group)
data$study <- sprintf("study%s", data$study)
pool <- hb_mcmc_pool(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
hierarchical <- hb_mcmc_hierarchical(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
hb_ess(
  mcmc_pool = pool,
  mcmc_hierarchical = hierarchical,
  data = data
)
```

---

# hb_mcmc_hierarchical *Hierarchical model MCMC*

---

## Description

Run the hierarchical model with MCMC.

## Usage

```
hb_mcmc_hierarchical(
  data,
  response = "response",
  study = "study",
  study_reference = max(data[[study]]),
  group = "group",
  group_reference = min(data[[group]]),
  patient = "patient",
  covariates = grep("^covariate", colnames(data), value = TRUE),
  s_delta = 30,
  s_beta = 30,
  s_sigma = 30,
  s_mu = 30,
  s_tau = sd(data[[response]], na.rm = TRUE),
  d_tau = 1,
```

```
  prior_tau = "half_t",
  n_chains = 4,
  n_adapt = 2000,
  n_warmup = 4000,
  n_iterations = 20000,
  quiet = TRUE
)
```

## Arguments

| | |
|---|---|
| data | Tidy data frame with one row per patient, indicator columns for the response variable, study, group, and patient, and covariates. All columns must be atomic vectors (e.g. not lists). The data for the mixture and simple models should have just one study, and the others should have data from more than one study. The simple model can be used to get the historical data components of m_omega and s_omega for the mixture model. |
| response | Character of length 1, name of the column in data with the response/outcome variable. data[[response]] must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| study | Character of length 1, name of the column in data with the study ID. |
| study_reference | Atomic of length 1, element of the study column that indicates the current study. (The other studies are historical studies.) |
| group | Character of length 1, name of the column in data with the group ID. |
| group_reference | Atomic of length 1, element of the group column that indicates the control group. (The other groups may be treatment groups.) |
| patient | Character of length 1, name of the column in data with the patient ID. |
| covariates | Character vector of column names in data with the columns with baseline covariates. These can be continuous, categorical, or binary. Regardless, historicalborrow derives the appropriate model matrix. |
| s_delta | Numeric of length 1, prior standard deviation of the study-by-group effect parameters delta. |
| s_beta | Numeric of length 1, prior standard deviation of the fixed effects beta. |
| s_sigma | Numeric of length 1, prior upper bound of the residual standard deviations. |
| s_mu | Numeric of length 1, prior standard deviation of mu. |
| s_tau | Non-negative numeric of length 1. If prior_tau is "half_t", then s_tau is the scale parameter of the Student t prior of tau and analogous to the sigma parameter of the Student-t parameterization given at https://mc-stan.org/docs/functions-reference/unbounded_continuous_distributions.html. # nolint If prior_tau is "uniform", then s_tau is the upper bound of tau. Upper bound on tau if prior_tau is "uniform". |

In the case of prior_tau equal to "half_t", the defaults s_tau = sd(data[[response]], na.rm = TRUE) and d_tau = 1 specify a weakly informative scaled half-Cauchy distribution. This choice is only provisional. The prior on tau is extremely important, especially for small numbers of historical studies, and the user should set a reasonable value for the use case, ideally informed by the results of sensitivity analyses and simulations.

d_tau — Positive numeric of length 1. Degrees of freedom of the Student t prior of tau if prior_tau is "half_t".

In the case of prior_tau equal to "half_t", the defaults s_tau = sd(data[[response]], na.rm = TRUE) and d_tau = 1 specify a weakly informative scaled half-Cauchy distribution. This choice is only provisional. The prior on tau is extremely important, especially for small numbers of historical studies, and the user should set a reasonable value for the use case, ideally informed by the results of sensitivity analyses and simulations.

prior_tau — Character string, family of the prior of tau. If prior_tau equals "uniform", then the prior on tau is a uniform prior with lower bound 0 and upper bound s_tau. If prior_tau equals "half_t", then the prior on tau is a half Student-t prior with center 0, lower bound 0, scale parameter s_tau, and degrees of freedom d_tau. The scale parameter s_tau is analogous to the sigma parameter of the Student-t parameterization given at https://mc-stan.org/docs/functions-reference/unbounded_continuous_distributions.html. # nolint

n_chains — Number of MCMC chains to run.

n_adapt — Number of adaptation iterations to run.

n_warmup — Number of warmup iterations per chain to run.

n_iterations — Number of saved MCMC iterations per chain to run.

quiet — Logical of length 1, TRUE to suppress R console output.

## Value

A tidy data frame of parameter samples from the posterior distribution. Columns .chain, .iteration, and .draw have the meanings documented in the posterior package.

## See Also

Other mcmc: hb_convergence(), hb_mcmc_independent(), hb_mcmc_mixture(), hb_mcmc_mixture_hyperparameters(
hb_mcmc_pool()

## Examples

```
if (!identical(Sys.getenv("HB_TEST", unset = ""), "")) {
data <- hb_sim_hierarchical(n_continuous = 2)$data
hb_mcmc_hierarchical(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
```

```
)
}
```

---

`hb_mcmc_independent` *Independent-study model MCMC*

---

## Description

Run the independent-study model with MCMC.

## Usage

```
hb_mcmc_independent(
  data,
  response = "response",
  study = "study",
  study_reference = max(data[[study]]),
  group = "group",
  group_reference = min(data[[group]]),
  patient = "patient",
  covariates = grep("^covariate", colnames(data), value = TRUE),
  s_alpha = 30,
  s_delta = 30,
  s_beta = 30,
  s_sigma = 30,
  n_chains = 4,
  n_adapt = 2000,
  n_warmup = 4000,
  n_iterations = 20000,
  quiet = TRUE
)
```

## Arguments

| | |
|---|---|
| `data` | Tidy data frame with one row per patient, indicator columns for the response variable, study, group, and patient, and covariates. All columns must be atomic vectors (e.g. not lists). The data for the mixture and simple models should have just one study, and the others should have data from more than one study. The simple model can be used to get the historical data components of `m_omega` and `s_omega` for the mixture model. |
| `response` | Character of length 1, name of the column in `data` with the response/outcome variable. `data[[response]]` must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| `study` | Character of length 1, name of the column in `data` with the study ID. |

study_reference
: Atomic of length 1, element of the study column that indicates the current study. (The other studies are historical studies.)

group
: Character of length 1, name of the column in data with the group ID.

group_reference
: Atomic of length 1, element of the group column that indicates the control group. (The other groups may be treatment groups.)

patient
: Character of length 1, name of the column in data with the patient ID.

covariates
: Character vector of column names in data with the columns with baseline covariates. These can be continuous, categorical, or binary. Regardless, historicalborrow derives the appropriate model matrix.

s_alpha
: Numeric of length 1, prior standard deviation of the study-specific control group mean parameters alpha.

s_delta
: Numeric of length 1, prior standard deviation of the study-by-group effect parameters delta.

s_beta
: Numeric of length 1, prior standard deviation of the fixed effects beta.

s_sigma
: Numeric of length 1, prior upper bound of the residual standard deviations.

n_chains
: Number of MCMC chains to run.

n_adapt
: Number of adaptation iterations to run.

n_warmup
: Number of warmup iterations per chain to run.

n_iterations
: Number of saved MCMC iterations per chain to run.

quiet
: Logical of length 1, TRUE to suppress R console output.

## Value

A tidy data frame of parameter samples from the posterior distribution. Columns .chain, .iteration, and .draw have the meanings documented in the posterior package.

## See Also

Other mcmc: hb_convergence(), hb_mcmc_hierarchical(), hb_mcmc_mixture(), hb_mcmc_mixture_hyperparameters
hb_mcmc_pool()

## Examples

```
if (!identical(Sys.getenv("HB_TEST", unset = ""), "")) {
data <- hb_sim_independent(n_continuous = 2)$data
hb_mcmc_independent(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
}
```

`hb_mcmc_mixture` *Mixture model MCMC*

## Description

Run the mixture model with MCMC.

## Usage

```
hb_mcmc_mixture(
  data,
  response = "response",
  study = "study",
  study_reference = max(data[[study]]),
  group = "group",
  group_reference = min(data[[group]]),
  patient = "patient",
  covariates = grep("^covariate", colnames(data), value = TRUE),
  s_delta = 30,
  s_beta = 30,
  s_sigma = 30,
  m_omega = c(0, 0),
  s_omega = c(30, 30),
  p_omega = 1/length(m_omega),
  n_chains = 4,
  n_adapt = 2000,
  n_warmup = 4000,
  n_iterations = 20000,
  quiet = TRUE
)
```

## Arguments

| | |
|---|---|
| `data` | Tidy data frame with one row per patient, indicator columns for the response variable, study, group, and patient, and covariates. All columns must be atomic vectors (e.g. not lists). The data for the mixture and simple models should have just one study, and the others should have data from more than one study. The simple model can be used to get the historical data components of `m_omega` and `s_omega` for the mixture model. |
| `response` | Character of length 1, name of the column in `data` with the response/outcome variable. `data[[response]]` must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| `study` | Character of length 1, name of the column in `data` with the study ID. |

`study_reference`
: Atomic of length 1, element of the `study` column that indicates the current study. (The other studies are historical studies.)

`group`
: Character of length 1, name of the column in `data` with the group ID.

`group_reference`
: Atomic of length 1, element of the `group` column that indicates the control group. (The other groups may be treatment groups.)

`patient`
: Character of length 1, name of the column in `data` with the patient ID.

`covariates`
: Character vector of column names in `data` with the columns with baseline covariates. These can be continuous, categorical, or binary. Regardless, `historicalborrow` derives the appropriate model matrix.

`s_delta`
: Numeric of length 1, prior standard deviation of the study-by-group effect parameters `delta`.

`s_beta`
: Numeric of length 1, prior standard deviation of the fixed effects `beta`.

`s_sigma`
: Numeric of length 1, prior upper bound of the residual standard deviations.

`m_omega`
: Numeric with length equal to the number of supposed studies (but only the current study is in the data). `m_omega` is the prior control group mean of each study. The last element corresponds to the current study, and the others are for historical studies.

`s_omega`
: Numeric with length equal to the number of supposed studies (but only the current study is in the data). `s_omega` is the prior control group standard deviation of each study. The last element corresponds to the current study, and the others are for historical studies.

`p_omega`
: Numeric with length equal to the number of supposed studies (but only the current study is in the data). `p_omega` is the prior control group mixture proportion of each study. The last element corresponds to the current study, and the others are for historical studies.

`n_chains`
: Number of MCMC chains to run.

`n_adapt`
: Number of adaptation iterations to run.

`n_warmup`
: Number of warmup iterations per chain to run.

`n_iterations`
: Number of saved MCMC iterations per chain to run.

`quiet`
: Logical of length 1, `TRUE` to suppress R console output.

## Details

The study-specific components of the mixture prior are all fixed in advance. Mixture components are normal distributions with means in `m_omega` and standard deviations in `s_omega`. These vectors are ordered with historical studies first and the current study last. These mixture components can be computed using `hb_mcmc_mixture_hyperparameters()` on a full set of data (all the historical studies and the current study together). Then the `m_omega` and `s_omega` columns of the output can be plugged directly into `hb_mcmc_mixture()`. See the examples for a demonstration.

## Value

A tidy data frame of parameter samples from the posterior distribution. Columns `.chain`, `.iteration`, and `.draw` have the meanings documented in the `posterior` package.

## See Also

Other mcmc: hb_convergence(), hb_mcmc_hierarchical(), hb_mcmc_independent(), hb_mcmc_mixture_hyperparame
hb_mcmc_pool()

## Examples

```
data_all_studies <- hb_sim_independent(n_continuous = 2)$data
data_all_studies$study <- paste0("study", data_all_studies$study)
hyperparameters <- hb_mcmc_mixture_hyperparameters(
  data = data_all_studies,
  response = "response",
  study = "study",
  study_reference = "study5",
  group = "group",
  group_reference = 1,
  patient = "patient",
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
print(hyperparameters)
data_current_study <- dplyr::filter(data_all_studies, study == max(study))
hb_mcmc_mixture(
  data = data_current_study,
  response = "response",
  study = "study",
  study_reference = "study5",
  group = "group",
  group_reference = 1,
  patient = "patient",
  m_omega = hyperparameters$m_omega, # use hyperparams from historical data
  s_omega = hyperparameters$s_omega, # use hyperparams from historical data
  p_omega = rep(1 / nrow(hyperparameters), nrow(hyperparameters)),
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
```

---

# hb_mcmc_mixture_hyperparameters

*Mixture model MCMC hyperparameters*

---

## Description

Run a simple model separately on each historical study control group and compute hyperparameters for hb_mcmc_mixture().

## Usage

```
hb_mcmc_mixture_hyperparameters(
  data,
  response = "response",
  study = "study",
  study_reference = max(data[[study]]),
  group = "group",
  group_reference = min(data[[group]]),
  patient = "patient",
  m_mu = 0,
  s_mu = 30,
  s_sigma = 30,
  m_omega_current = 0,
  s_omega_current = 30,
  n_chains = 4,
  n_adapt = 2000,
  n_warmup = 4000,
  n_iterations = 20000,
  quiet = TRUE
)
```

## Arguments

| | |
|---|---|
| `data` | Tidy data frame with one row per patient, indicator columns for the response variable, study, group, and patient, and covariates. All columns must be atomic vectors (e.g. not lists). The data for the mixture and simple models should have just one study, and the others should have data from more than one study. The simple model can be used to get the historical data components of `m_omega` and `s_omega` for the mixture model. |
| `response` | Character of length 1, name of the column in `data` with the response/outcome variable. `data[[response]]` must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| `study` | Character of length 1, name of the column in `data` with the study ID. |
| `study_reference` | Atomic of length 1, element of the `study` column that indicates the current study. (The other studies are historical studies.) |
| `group` | Character of length 1, name of the column in `data` with the group ID. |
| `group_reference` | Atomic of length 1, element of the `group` column that indicates the control group. (The other groups may be treatment groups.) |
| `patient` | Character of length 1, name of the column in `data` with the patient ID. |
| `m_mu` | Numeric of length 1, prior mean of the mean `mu` in the simple model. |

| | |
|---|---|
| s_mu | Numeric of length 1, prior standard deviation of the mean mu in the simple model. |
| s_sigma | Numeric of length 1, uniform prior upper bound of the residual standard deviation sigma in the simple model. |
| m_omega_current | Numeric with length 1, m_omega value of the current study. Inserted as the final component of the m_omega column in the output. |
| s_omega_current | Numeric with length 1, s_omega value of the current study. Inserted as the final component of the s_omega column in the output. |
| n_chains | Number of MCMC chains to run. |
| n_adapt | Number of adaptation iterations to run. |
| n_warmup | Number of warmup iterations per chain to run. |
| n_iterations | Number of saved MCMC iterations per chain to run. |
| quiet | Logical of length 1, TRUE to suppress R console output. |

## Details

The model is a simple Bayesian model with a normal likelihood, an unknown mean mu, and an unknown standard deviation sigma. For each historical study, the posterior mean of mu becomes the corresponding component of m_omega in the output, and the posterior standard deviation of mu becomes the corresponding component of s_omega in the output. See the examples in this help file for a demonstration. m_omega and s_omega define the components of the mixture prior in hb_mcmc_mixture() that act as the contribution of the historical studies to the model.

## Value

A tidy data frame of hyperparameter values for hb_mcmc_mixture(). The first several rows are for historical studies, and the last row is for the current study. Studies/rows are sorted in the order hb_mcmc_mixture() sorts them, so you can use columns m_omega and s_omega for the same dataset and same values of other arguments directly in hb_mcmc_mixture().

## See Also

Other mcmc: hb_convergence(), hb_mcmc_hierarchical(), hb_mcmc_independent(), hb_mcmc_mixture(), hb_mcmc_pool()

## Examples

```
data_all_studies <- hb_sim_independent(n_continuous = 2)$data
data_all_studies$study <- paste0("study", data_all_studies$study)
hyperparameters <- hb_mcmc_mixture_hyperparameters(
  data = data_all_studies,
  response = "response",
  study = "study",
  study_reference = "study5",
  group = "group",
  group_reference = 1,
```

```
  patient = "patient",
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
print(hyperparameters)
data_current_study <- dplyr::filter(data_all_studies, study == max(study))
hb_mcmc_mixture(
  data = data_current_study,
  response = "response",
  study = "study",
  study_reference = "study5",
  group = "group",
  group_reference = 1,
  patient = "patient",
  m_omega = hyperparameters$m_omega, # use hyperparams from historical data
  s_omega = hyperparameters$s_omega, # use hyperparams from historical data
  p_omega = rep(1 / nrow(hyperparameters), nrow(hyperparameters)),
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
```

---

hb_mcmc_pool *Non-longitudinal pooled MCMC*

---

## Description

Run the non-longitudinal pooled model with MCMC.

## Usage

```
hb_mcmc_pool(
  data,
  response = "response",
  study = "study",
  study_reference = max(data[[study]]),
  group = "group",
  group_reference = min(data[[group]]),
  patient = "patient",
  covariates = grep("^covariate", colnames(data), value = TRUE),
  s_alpha = 30,
  s_delta = 30,
  s_beta = 30,
  s_sigma = 30,
  n_chains = 4,
  n_adapt = 2000,
```

```
  n_warmup = 4000,
  n_iterations = 20000,
  quiet = TRUE
)
```

## Arguments

| | |
|---|---|
| data | Tidy data frame with one row per patient, indicator columns for the response variable, study, group, and patient, and covariates. All columns must be atomic vectors (e.g. not lists). The data for the mixture and simple models should have just one study, and the others should have data from more than one study. The simple model can be used to get the historical data components of m_omega and s_omega for the mixture model. |
| response | Character of length 1, name of the column in data with the response/outcome variable. data[[response]] must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| study | Character of length 1, name of the column in data with the study ID. |
| study_reference | Atomic of length 1, element of the study column that indicates the current study. (The other studies are historical studies.) |
| group | Character of length 1, name of the column in data with the group ID. |
| group_reference | Atomic of length 1, element of the group column that indicates the control group. (The other groups may be treatment groups.) |
| patient | Character of length 1, name of the column in data with the patient ID. |
| covariates | Character vector of column names in data with the columns with baseline covariates. These can be continuous, categorical, or binary. Regardless, historicalborrow derives the appropriate model matrix. |
| s_alpha | Numeric of length 1, prior standard deviation of the study-specific control group mean parameters alpha. |
| s_delta | Numeric of length 1, prior standard deviation of the study-by-group effect parameters delta. |
| s_beta | Numeric of length 1, prior standard deviation of the fixed effects beta. |
| s_sigma | Numeric of length 1, prior upper bound of the residual standard deviations. |
| n_chains | Number of MCMC chains to run. |
| n_adapt | Number of adaptation iterations to run. |
| n_warmup | Number of warmup iterations per chain to run. |
| n_iterations | Number of saved MCMC iterations per chain to run. |
| quiet | Logical of length 1, TRUE to suppress R console output. |

## Value

A tidy data frame of parameter samples from the posterior distribution. Columns `.chain`, `.iteration`, and `.draw` have the meanings documented in the `posterior` package.

## See Also

Other mcmc: `hb_convergence()`, `hb_mcmc_hierarchical()`, `hb_mcmc_independent()`, `hb_mcmc_mixture()`, `hb_mcmc_mixture_hyperparameters()`

## Examples

```
if (!identical(Sys.getenv("HB_TEST", unset = ""), "")) {
data <- hb_sim_pool(n_continuous = 2)$data
hb_mcmc_pool(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
}
```

---

# hb_plot_borrow *Plot a borrowing model response against the benchmark models.*

---

## Description

Plot the response from a borrowing model (hierarchical or mixture) against the independent and pooled benchmark models.

## Usage

```
hb_plot_borrow(borrow, pool, independent, outcome = c("response", "diff"))
```

## Arguments

| | |
|---|---|
| `borrow` | A data frame returned by `hb_summary()` for the mixture or hierarchical model. |
| `pool` | A data frame returned by `hb_summary()` for the pooled model. |
| `independent` | A data frame returned by `hb_summary()` for the independent model. |
| `outcome` | Character of length 1, either `"response"` or `"diff"`, the quantity to plot on the vertical axis. |

## Value

A `ggplot` object

## See Also

Other plot: hb_plot_group(), hb_plot_tau()

## Examples

```
if (!identical(Sys.getenv("HB_TEST", unset = ""), "")) {
data <- hb_sim_independent(n_continuous = 2)$data
mcmc_borrow <- hb_mcmc_hierarchical(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 100,
  n_iterations = 200
)
mcmc_pool <- hb_mcmc_pool(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 200,
  n_iterations = 200
)
mcmc_independent <- hb_mcmc_independent(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 200,
  n_iterations = 200
)
borrow <- hb_summary(mcmc_borrow, data)
pool <- hb_summary(mcmc_pool, data)
independent <- hb_summary(mcmc_independent, data)
hb_plot_borrow(
  borrow = borrow,
  pool = pool,
  independent = independent
)
}
```

---

# hb_plot_group *Plot the groups of a borrowing model and its benchmark models.*

---

## Description

Plot the groups against one another for a borrowing model (hierarchical or mixture) and the independent and pooled benchmark models.

## Usage

```
hb_plot_group(borrow, pool, independent, outcome = c("response", "diff"))
```

## Arguments

| | |
|---|---|
| borrow | A data frame returned by hb_summary() for the mixture or hierarchical model. |
| pool | A data frame returned by hb_summary() for the pooled model. |
| independent | A data frame returned by hb_summary() for the independent model. |
| outcome | Character of length 1, either "response" or "diff", the quantity to plot on the vertical axis. |

## Value

A ggplot object

## See Also

Other plot: hb_plot_borrow(), hb_plot_tau()

## Examples

```
if (!identical(Sys.getenv("HB_TEST", unset = ""), "")) {
data <- hb_sim_independent(n_continuous = 2)$data
mcmc_borrow <- hb_mcmc_hierarchical(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 100,
  n_iterations = 200
)
mcmc_pool <- hb_mcmc_pool(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 200,
  n_iterations = 200
)
mcmc_independent <- hb_mcmc_independent(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 200,
  n_iterations = 200
)
borrow <- hb_summary(mcmc_borrow, data)
pool <- hb_summary(mcmc_pool, data)
independent <- hb_summary(mcmc_independent, data)
hb_plot_group(
  borrow = borrow,
  pool = pool,
  independent = independent
)
}
```

## hb_plot_tau *Plot tau*

### Description

Plot the tau parameter of a fitted hierarchical model.

### Usage

```
hb_plot_tau(mcmc)
```

### Arguments

| | |
|---|---|
| mcmc | Data frame of posterior samples generated by hb_mcmc_hierarchical(). |

### Value

A ggplot object

### See Also

Other plot: hb_plot_borrow(), hb_plot_group()

### Examples

```
data <- hb_sim_independent(n_continuous = 2)$data
mcmc <- hb_mcmc_hierarchical(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 100,
  n_iterations = 200
)
hb_plot_tau(mcmc = mcmc)
```

## hb_sim_hierarchical *Non-longitudinal hierarchical simulations.*

### Description

Simulate from the non-longitudinal hierarchical model.

## Usage

```
hb_sim_hierarchical(
  n_study = 5,
  n_group = 3,
  n_patient = 100,
  n_continuous = 0,
  n_binary = 0,
  s_delta = 1,
  s_beta = 1,
  s_sigma = 1,
  s_mu = 1,
  s_tau = 1,
  d_tau = 4,
  prior_tau = "half_t",
  alpha = NULL,
  delta = stats::rnorm(n = n_group - 1, mean = 0, sd = s_delta),
 beta = stats::rnorm(n = n_study * (n_continuous + n_binary), mean = 0, sd = s_delta),
  sigma = stats::runif(n = n_study, min = 0, max = s_sigma),
  mu = stats::rnorm(n = 1, mean = 0, sd = s_mu),
  tau = NULL
)
```

## Arguments

| | |
|---|---|
| n_study | Number of studies to simulate. |
| n_group | Number of groups (e.g. study arms) to simulate per study. |
| n_patient | Number of patients to simulate per study per group. |
| n_continuous | Number of continuous covariates to simulate (all from independent standard normal distributions). |
| n_binary | Number of binary covariates to simulate (all from independent Bernoulli distributions with p = 0.5). |
| s_delta | Numeric of length 1, prior standard deviation of the study-by-group effect parameters delta. |
| s_beta | Numeric of length 1, prior standard deviation of the fixed effects beta. |
| s_sigma | Numeric of length 1, prior upper bound of the residual standard deviations. |
| s_mu | Numeric of length 1, prior standard deviation of mu. |
| s_tau | Non-negative numeric of length 1. If prior_tau is "half_t", then s_tau is the scale parameter of the Student t prior of tau and analogous to the sigma parameter of the Student-t parameterization given at https://mc-stan.org/docs/functions-reference/unbounded_continuous_distributions.html. # nolint If prior_tau is "uniform", then s_tau is the upper bound of tau. Upper bound on tau if prior_tau is "uniform". |
| d_tau | Positive numeric of length 1. Degrees of freedom of the Student t prior of tau if prior_tau is "half_t". |

prior_tau — Character string, family of the prior of tau. If prior_tau equals "uniform", then the prior on tau is a uniform prior with lower bound 0 and upper bound s_tau. If prior_tau equals "half_t", then the prior on tau is a half Student-t prior with center 0, lower bound 0, scale parameter s_tau, and degrees of freedom d_tau. The scale parameter s_tau is analogous to the sigma parameter of the Student-t parameterization given at https://mc-stan.org/docs/functions-reference/unbounded_continuous_distributions.html. # nolint

alpha — Numeric vector of length 1 for the pooled and mixture models and length n_study for the independent and hierarchical models. alpha is the vector of control group mean parameters. alpha enters the model by multiplying with $matrices$x_alpha (see the return value). The control group in the data is the one with the group column equal to 1.

delta — Numeric vector of length n_group - 1 of treatment effect parameters. delta enters the model by multiplying with $matrices$x_delta (see the return value). The control (non-treatment) group in the data is the one with the group column equal to 1.

beta — Numeric vector of n_study * (n_continuous + n_binary) fixed effect parameters. Within each study, the first n_continuous betas are for the continuous covariates, and the rest are for the binary covariates. All the betas for one study appear before all the betas for the next study, and studies are arranged in increasing order of the sorted unique values in $data$study in the output. betas enters the model by multiplying with $matrices$x_alpha (see the return value).

sigma — Numeric vector of n_study study-specific residual standard deviations.

mu — Numeric of length 1, mean of the control group means alpha.

tau — Numeric of length 1, standard deviation of the control group means alpha.

## Value

A list with the following elements:

- data: tidy long-form dataset with the patient-level data. one row per patient and indicator columns for the study, group (e.g. treatment arm), and patient ID. The response columns is the patient response. The other columns are baseline covariates. The control group is the one with the group column equal to 1, and the current study (non-historical) is the one with the maximum value of the study column. Only the current study has any non-control-group patients, the historical studies have only the control group.
- parameters: named list of model parameter values. See the model specification vignette for details.
- matrices: A named list of model matrices. See the model specification vignette for details.

## See Also

Other simulate: hb_sim_independent(), hb_sim_mixture(), hb_sim_pool()

## Examples

```
hb_sim_hierarchical()$data
```

---

`hb_sim_independent` *Non-longitudinal independent simulations.*

---

## Description

Simulate from the non-longitudinal independent model.

## Usage

```
hb_sim_independent(
  n_study = 5,
  n_group = 3,
  n_patient = 100,
  n_continuous = 0,
  n_binary = 0,
  s_alpha = 1,
  s_delta = 1,
  s_beta = 1,
  s_sigma = 1,
  alpha = stats::rnorm(n = n_study, mean = 0, sd = s_alpha),
  delta = stats::rnorm(n = n_group - 1, mean = 0, sd = s_delta),
 beta = stats::rnorm(n = n_study * (n_continuous + n_binary), mean = 0, sd = s_delta),
  sigma = stats::runif(n = n_study, min = 0, max = s_sigma)
)
```

## Arguments

| | |
|---|---|
| `n_study` | Number of studies to simulate. |
| `n_group` | Number of groups (e.g. study arms) to simulate per study. |
| `n_patient` | Number of patients to simulate per study per group. |
| `n_continuous` | Number of continuous covariates to simulate (all from independent standard normal distributions). |
| `n_binary` | Number of binary covariates to simulate (all from independent Bernoulli distributions with p = 0.5). |
| `s_alpha` | Numeric of length 1, prior standard deviation of the study-specific control group mean parameters `alpha`. |
| `s_delta` | Numeric of length 1, prior standard deviation of the study-by-group effect parameters `delta`. |
| `s_beta` | Numeric of length 1, prior standard deviation of the fixed effects `beta`. |
| `s_sigma` | Numeric of length 1, prior upper bound of the residual standard deviations. |

| | |
|---|---|
| alpha | Numeric vector of length 1 for the pooled and mixture models and length n_study for the independent and hierarchical models. alpha is the vector of control group mean parameters. alpha enters the model by multiplying with $matrices$x_alpha (see the return value). The control group in the data is the one with the group column equal to 1. |
| delta | Numeric vector of length n_group - 1 of treatment effect parameters. delta enters the model by multiplying with $matrices$x_delta (see the return value). The control (non-treatment) group in the data is the one with the group column equal to 1. |
| beta | Numeric vector of n_study * (n_continuous + n_binary) fixed effect parameters. Within each study, the first n_continuous betas are for the continuous covariates, and the rest are for the binary covariates. All the betas for one study appear before all the betas for the next study, and studies are arranged in increasing order of the sorted unique values in $data$study in the output. betas enters the model by multiplying with $matrices$x_alpha (see the return value). |
| sigma | Numeric vector of n_study study-specific residual standard deviations. |

## Value

A list with the following elements:

- data: tidy long-form dataset with the patient-level data. one row per patient and indicator columns for the study, group (e.g. treatment arm), and patient ID. The response columns is the patient response. The other columns are baseline covariates. The control group is the one with the group column equal to 1, and the current study (non-historical) is the one with the maximum value of the study column. Only the current study has any non-control-group patients, the historical studies have only the control group.
- parameters: named list of model parameter values. See the model specification vignette for details.
- matrices: A named list of model matrices. See the model specification vignette for details.

## See Also

Other simulate: hb_sim_hierarchical(), hb_sim_mixture(), hb_sim_pool()

## Examples

```
hb_sim_independent()$data
```

---

# hb_sim_mixture *Non-longitudinal mixture simulations.*

---

## Description

Simulate from the non-longitudinal mixture model.

## Usage

```
hb_sim_mixture(
  n_study = 5,
  n_group = 3,
  n_patient = 100,
  n_continuous = 0,
  n_binary = 0,
  s_delta = 1,
  s_beta = 1,
  s_sigma = 1,
  m_omega = 0,
  s_omega = 1,
  p_omega = 1/n_study,
  alpha = omega[pi],
  delta = stats::rnorm(n = n_group - 1, mean = 0, sd = s_delta),
  beta = stats::rnorm(n = n_continuous + n_binary, mean = 0, sd = s_delta),
  sigma = stats::runif(n = 1, min = 0, max = s_sigma),
  pi = sample.int(n = n_study, size = 1, prob = p_omega),
  omega = stats::rnorm(n = n_study, mean = m_omega, sd = s_omega)
)
```

## Arguments

| | |
|---|---|
| n_study | Number of studies to simulate. |
| n_group | Number of groups (e.g. study arms) to simulate per study. |
| n_patient | Number of patients to simulate per study per group. |
| n_continuous | Number of continuous covariates to simulate (all from independent standard normal distributions). |
| n_binary | Number of binary covariates to simulate (all from independent Bernoulli distributions with p = 0.5). |
| s_delta | Numeric of length 1, prior standard deviation of the study-by-group effect parameters delta. |
| s_beta | Numeric of length 1, prior standard deviation of the fixed effects beta. |
| s_sigma | Numeric of length 1, prior upper bound of the residual standard deviations. |
| m_omega | Numeric of length 1 or n_study, prior control group mean of each study. If length n_study, then the last element corresponds to the current study, and the others are for historical studies. |
| s_omega | Numeric of length 1 or n_study, prior control group standard deviation of each study. If length n_study, the the last element corresponds to the current study, and the others are for historical studies. |
| p_omega | Numeric of length n_study, prior mixture proportion of each study. If length n_study, then the last element corresponds to the current study, and the others are for historical studies. |
| alpha | Numeric vector of length 1 for the pooled and mixture models and length n_study for the independent and hierarchical models. alpha is the vector of control |

| | |
|---|---|
| | group mean parameters. alpha enters the model by multiplying with $matrices$x_alpha (see the return value). The control group in the data is the one with the group column equal to 1. |
| delta | Numeric vector of length n_group - 1 of treatment effect parameters. delta enters the model by multiplying with $matrices$x_delta (see the return value). The control (non-treatment) group in the data is the one with the group column equal to 1. |
| beta | Numeric vector of n_continuous + n_binary fixed effect parameters. The first n_continuous betas are for the continuous covariates, and the rest are for the binary covariates. betas enters the model by multiplying with $matrices$x_alpha (see the return value). |
| sigma | Numeric vector of n_study study-specific residual standard deviations. |
| pi | Integer of length 1, index of the mixture component randomly chosen for alpha. |
| omega | Numeric of length n_study, Candidate placebo mean parameters drawn from each of the mixture components. |

## Value

A list with the following elements:

- data: tidy long-form dataset with the patient-level data. one row per patient and indicator columns for the study, group (e.g. treatment arm), and patient ID. The response columns is the patient response. The other columns are baseline covariates. The control group is the one with the group column equal to 1, and the current study (non-historical) is the one with the maximum value of the study column. Only the current study has any non-control-group patients, the historical studies have only the control group.
- parameters: named list of model parameter values. See the model specification vignette for details.
- matrices: A named list of model matrices. See the model specification vignette for details.

## See Also

Other simulate: hb_sim_hierarchical(), hb_sim_independent(), hb_sim_pool()

## Examples

```
hb_sim_mixture()$data
```

---

# hb_sim_pool *Non-longitudinal pooled simulations.*

---

## Description

Simulate from the non-longitudinal pooled model.

## Usage

```
hb_sim_pool(
  n_study = 5,
  n_group = 3,
  n_patient = 100,
  n_continuous = 0,
  n_binary = 0,
  s_alpha = 1,
  s_delta = 1,
  s_beta = 1,
  s_sigma = 1,
  alpha = stats::rnorm(n = 1, mean = 0, sd = s_alpha),
  delta = stats::rnorm(n = n_group - 1, mean = 0, sd = s_delta),
 beta = stats::rnorm(n = n_study * (n_continuous + n_binary), mean = 0, sd = s_delta),
  sigma = stats::runif(n = n_study, min = 0, max = s_sigma)
)
```

## Arguments

| | |
|---|---|
| n_study | Number of studies to simulate. |
| n_group | Number of groups (e.g. study arms) to simulate per study. |
| n_patient | Number of patients to simulate per study per group. |
| n_continuous | Number of continuous covariates to simulate (all from independent standard normal distributions). |
| n_binary | Number of binary covariates to simulate (all from independent Bernoulli distributions with p = 0.5). |
| s_alpha | Numeric of length 1, prior standard deviation of the study-specific control group mean parameters alpha. |
| s_delta | Numeric of length 1, prior standard deviation of the study-by-group effect parameters delta. |
| s_beta | Numeric of length 1, prior standard deviation of the fixed effects beta. |
| s_sigma | Numeric of length 1, prior upper bound of the residual standard deviations. |
| alpha | Numeric vector of length 1 for the pooled and mixture models and length n_study for the independent and hierarchical models. alpha is the vector of control group mean parameters. alpha enters the model by multiplying with $matrices$x_alpha (see the return value). The control group in the data is the one with the group column equal to 1. |
| delta | Numeric vector of length n_group - 1 of treatment effect parameters. delta enters the model by multiplying with $matrices$x_delta (see the return value). The control (non-treatment) group in the data is the one with the group column equal to 1. |
| beta | Numeric vector of n_study * (n_continuous + n_binary) fixed effect parameters. Within each study, the first n_continuous betas are for the continuous covariates, and the rest are for the binary covariates. All the betas for one study appear before all the betas for the next study, and studies are arranged |

| | |
|---|---|
| | in increasing order of the sorted unique values in `$data$study` in the output. `betas` enters the model by multiplying with `$matrices$x_alpha` (see the return value). |
| `sigma` | Numeric vector of `n_study` study-specific residual standard deviations. |

## Value

A list with the following elements:

- `data`: tidy long-form dataset with the patient-level data. one row per patient and indicator columns for the study, group (e.g. treatment arm), and patient ID. The `response` columns is the patient response. The other columns are baseline covariates. The control group is the one with the `group` column equal to 1, and the current study (non-historical) is the one with the maximum value of the `study` column. Only the current study has any non-control-group patients, the historical studies have only the control group.
- `parameters`: named list of model parameter values. See the model specification vignette for details.
- `matrices`: A named list of model matrices. See the model specification vignette for details.

## See Also

Other simulate: `hb_sim_hierarchical()`, `hb_sim_independent()`, `hb_sim_mixture()`

## Examples

```
hb_sim_pool(n_continuous = 1)$data
```

---

# hb_summary *Model summary*

---

## Description

Summarize a fitted model in a table.

## Usage

```
hb_summary(
  mcmc,
  data,
  response = "response",
  study = "study",
  study_reference = max(data[[study]]),
  group = "group",
  group_reference = min(data[[group]]),
  patient = "patient",
  covariates = grep("^covariate", colnames(data), value = TRUE),
  eoi = 0,
  direction = "<"
)
```

## Arguments

| | |
|---|---|
| mcmc | A wide data frame of posterior samples returned by hb_mcmc_hierarchical() or similar MCMC function. |
| data | Tidy data frame with one row per patient, indicator columns for the response variable, study, group, and patient, and covariates. All columns must be atomic vectors (e.g. not lists). The data for the mixture and simple models should have just one study, and the others should have data from more than one study. The simple model can be used to get the historical data components of m_omega and s_omega for the mixture model. |
| response | Character of length 1, name of the column in data with the response/outcome variable. data[[response]] must be a continuous variable, and it *should* be the change from baseline of a clinical endpoint of interest, as opposed to just the raw response. Treatment differences are computed directly from this scale, please supply change from baseline unless you are absolutely certain that treatment differences computed directly from this quantity are clinically meaningful. |
| study | Character of length 1, name of the column in data with the study ID. |
| study_reference | Atomic of length 1, element of the study column that indicates the current study. (The other studies are historical studies.) |
| group | Character of length 1, name of the column in data with the group ID. |
| group_reference | Atomic of length 1, element of the group column that indicates the control group. (The other groups may be treatment groups.) |
| patient | Character of length 1, name of the column in data with the patient ID. |
| covariates | Character vector of column names in data with the columns with baseline covariates. These can be continuous, categorical, or binary. Regardless, historicalborrow derives the appropriate model matrix. |
| eoi | Numeric of length at least 1, vector of effects of interest (EOIs) for critical success factors (CSFs). |
| direction | Character of length length(eoi) indicating how to compare the treatment effect to each EOI. ">" means Prob(treatment effect > EOI), and "<" means Prob(treatment effect < EOI). All elements of direction must be either ">" or "<". |

## Details

The hb_summary() function post-processes the results from the model. It estimates marginal means of the response, treatment effect, and other quantities of interest.

## Value

A tidy data frame with one row per group (e.g. treatment arm) and the columns in the following list. Unless otherwise specified, the quantities are calculated at the group level. Some are calculated for the current (non-historical) study only, while others pertain to the combined dataset which includes all historical studies. The mixture model is an exception because the data argument only includes the current study, so other quantities that include historical information will need to borrow from an hb_summary() call on one of the other models.

- group: group label.
- data_mean: observed mean response specific to the current study.
- data_sd: observed standard deviation of the response specific to the current study.
- data_lower: lower bound of a simple frequentist 95% confidence interval of the observed mean specific to the current study.
- data_upper: upper bound of a simple frequentist 95% confidence interval of the observed mean specific to the current study.
- data_n: number of non-missing observations in the combined dataset with all studies.
- data_N: total number of observations (missing and non-missing) in the combined dataset with all studies.
- data_n_study_*: number of non-missing observations separately for each study. The suffixes of these column names are integer study indexes. Call dplyr::distinct(hb_data(your_data), study, study_label) to see which study labels correspond to these integer indexes. Note: the combined dataset for the mixture model is just the current study. If all the data_n_study_* results across all studies are desired, then call hb_summary() on a different model (e.g. pooled).
- data_N_study_*: same as data_n_study_* except both missing and non-missing observations are counted (total number of observations).
- response_mean: Estimated posterior mean of the response from the model specific to the current study. Typically, the raw response is change from baseline, in which case response_mean is estimating change from baseline.
- response_sd: Estimated posterior standard deviation of the mean response from the model specific to the current study.
- response_variance: Estimated posterior variance of the mean response from the model specific to the current study.
- response_lower: Lower bound of a 95% posterior interval on the mean response from the model specific to the current study.
- response_upper: Upper bound of a 95% posterior interval on the mean response from the model specific to the current study.
- response_mean_mcse: Monte Carlo standard error of response_mean.
- response_sd_mcse: Monte Carlo standard error of response_sd.
- response_lower_mcse: Monte Carlo standard error of response_lower.
- response_upper_mcse: Monte Carlo standard error of response_upper.
- diff_mean: Estimated treatment effect from the model specific to the current study.
- diff_lower: Lower bound of a 95% posterior interval on the treatment effect from the model specific to the current study..
- diff_upper: Upper bound of a 95% posterior interval on the treatment effect from the model specific to the current study..
- diff_mean_mcse: Monte Carlo standard error of diff_mean.
- diff_lower_mcse: Monte Carlo standard error of diff_lower.
- diff_upper_mcse: Monte Carlo standard error of diff_upper.
- P(diff > EOI), P(diff < EOI): CSF probabilities on the treatment effect specified with the eoi and direction arguments. Specific to the current study.

- `effect_mean`: Estimated posterior mean of effect size (treatment difference divided by residual standard deviation). Specific to the current study.
- `effect_lower`: Lower bound of a 95% posterior interval of effect size from the model. Specific to the current study.
- `effect_upper`: Upper bound of a 95% posterior interval of effect size from the model. Specific to the current study.
- `precision_ratio`: For the hierarchical model only, a model-based mean of the precision ratio. Specific to the current study.
- `precision_ratio_lower`: For the hierarchical model only, lower bound of a model-based 95% posterior interval of the precision ratio. Specific to the current study.
- `precision_ratio_upper`: For the hierarchical model only, upper bound of a model-based 95% posterior interval of the precision ratio. Specific to the current study.
- `mix_prop_*`: For the mixture model only, posterior mixture proportions of each of the mixture components. The last one is for the current study and the first ones are for the historical studies. The suffixes of these column names are the integer study indexes. Call `dplyr::distinct(hb_data(your_data), study, study_label)` to see which study labels correspond to these integer indexes.

## See Also

Other summary: `hb_ess()`

## Examples

```
if (!identical(Sys.getenv("HB_TEST", unset = ""), "")) {
data <- hb_sim_pool(n_continuous = 2)$data
data$group <- sprintf("group%s", data$group)
mcmc <- hb_mcmc_pool(
  data,
  n_chains = 1,
  n_adapt = 100,
  n_warmup = 50,
  n_iterations = 50
)
hb_summary(mcmc, data)
}
```

# Index