Package: bmco (via r-universe)

October 19, 2026

Title Bayesian Analysis for Multivariate Categorical Outcomes

Version 0.1.0

Description Provides Bayesian methods for comparing groups on multiple binary outcomes. Includes basic tests using multivariate Bernoulli distributions, subgroup analysis via generalized linear models, and multilevel models for clustered data. For statistical underpinnings, see Kavelaars, Mulder, and Kaptein (2020) <doi:10.1177/0962280220922256>, Kavelaars, Mulder, and Kaptein (2024) <doi:10.1080/00273171.2024.2337340>, and Kavelaars, Mulder, and Kaptein (2023) <doi:10.1186/s12874-023-02034-z>. An interactive shiny app to perform sample size computations is available.

License GPL (>= 3)

Encoding UTF-8

LazyData TRUE

LazyDataCompression xz

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports abind, coda, MCMCpack, msm, pgdraw, tcltk, Rdpack, stats

Suggests knitr, rmarkdown, testthat, RefManageR

Config/testthat/edition 3

VignetteBuilder knitr

RdMacros Rdpack

URL https://github.com/XynthiaKavelaars/bmco, https://xynthia-kavelaars.shinyapps.io/bmco-pwr/, https://xynthiakavelaars.github.io/bmco/

BugReports https://github.com/XynthiaKavelaars/bmco/issues

Depends R (>= 3.5)

Repository https://xynthiakavelaars.r-universe.dev

Date/Publication 2026-04-22 11:54:15 UTC

RemoteUrl https://github.com/xynthiakavelaars/bmco

RemoteRef HEAD

RemoteSha 9eabe8712309c8c233cb29ef13b95cb0ee2854e4

Contents

bmco-package 2
bglm 3
bglm_data 6
bglm_fit 7
bglmm 8
bglmm_data 11
bglmm_fit 12
bmvb 13
plot.bglm 15
plot.bglmm 16
print.bglm 16
print.bglmm 17
print.bmvb 18
print.summary.bglm 19
print.summary.bglmm 19
print.summary.bmvb 20
summary.bglm 21
summary.bglmm 22
summary.bmvb 23

Index **25**

`bmco-package` *bmco: Bayesian Analysis for Multivariate Categorical Outcomes*

Description

Provides Bayesian methods for comparing groups on multiple binary outcomes, including basic tests, regression adjustment, and multilevel models.

Main functions

- `bmvb`: Bayesian test using multivariate Bernoulli
- `bglm`: Subgroup analysis using Bayesian logistic regression analysis
- `bglmm`: Multilevel data using Bayesian multilevel logistic regression analysis

Author(s)

Maintainer: Xynthia Kavelaars <xynthia.kavelaars@ou.nl> (ORCID)

Other contributors:

- Joris Mulder (ORCID) [thesis advisor]
- Maurits Kaptein (ORCID) [thesis advisor]
- Dutch Research Council (Grant no. 406.18.505) [funder]

References

Kavelaars X, Mulder J, Kaptein M (2020). "Decision-making with multiple correlated binary outcomes in clinical trials." *Statistical Methods in Medical Research*, **29**(11), 3265–3277. doi:10.1177/0962280220922256.

Kavelaars X, Mulder J, Kaptein M (2024). "Bayesian Multivariate Logistic Regression for Superiority and Inferiority Decision-Making under Observable Treatment Heterogeneity." *Multivariate Behavioral Research*, **59**(4), 859–882. doi:10.1080/00273171.2024.2337340.

Kavelaars X, Mulder J, Kaptein M (2023). "Bayesian multilevel multivariate logistic regression for superiority decision-making under observable treatment heterogeneity." *BMC Medical Research Methodology*, **23**(1). doi:10.1186/s1287402302034z.

See Also

Useful links:

- https://github.com/XynthiaKavelaars/bmco
- https://xynthia-kavelaars.shinyapps.io/bmco-pwr/
- Report bugs at https://github.com/XynthiaKavelaars/bmco/issues

---

`bglm` *Bayesian Generalized Linear Model*

Description

Perform a Bayesian test for differences between two (sub)groups on multiple binary outcomes using multinomial logistic regression as described in Kavelaars et al. (2024).

Usage

```
bglm(
  data,
  grp,
  grp_a,
  grp_b,
  x_var,
  y_vars,
```

```
  x_method = c("Empirical", "Analytical", "Value"),
  x_def = c(-Inf, Inf),
  test = c("right_sided", "left_sided"),
  rule = c("All", "Any", "Comp"),
  w = NULL,
  b_mu0 = NULL,
  b_sigma0 = NULL,
  n_burn = 10000,
  n_it = 20000,
  n_thin = 1,
  n_chain = 2,
  start = c(0.5, 1),
  return_diagnostics = TRUE,
  return_diagnostic_plots = FALSE,
  return_samples = FALSE
)
```

Arguments

data	Data frame containing the data.
grp	Character string. Name of the grouping variable (will be treated as factor).
grp_a	Value of grp indicating first group.
grp_b	Value of grp indicating second group.
x_var	Character string. Name of covariate variable (currently supports single continuous or binary covariate)
y_vars	Character vector. Names of outcome variables (currently supports 2 outcomes).
x_method	Character. Method for handling covariate: "Analytical" (numerical integration), "Empirical" (empirical marginalization), or "Value" (specific value). Default is "Empirical".
x_def	Numeric vector. Defines subpopulation: length-2 vector c(lower, upper) for "Analytical"/"Empirical", or scalar for "Value". Default is c(-Inf, Inf).
test	Character. Direction of test: "left_sided" for P(A>B) or "right_sided" for P(B>A). Default is "right_sided".
rule	Character. Decision rule: "All" (all outcomes favor hypothesis), "Any" (at least one outcome favors hypothesis), or "Comp" (weighted combination). Default is "All".
w	Numeric vector. Weights for compensatory rule. Only used if rule = "Comp". If NULL and rule = "Comp", equal weights are used. Default is NULL.
b_mu0	Vector of prior means of fixed regression coefficients. Default is `rep(0, P)`, where P refers to the number of columns in the model matrix.
b_sigma0	Prior covariance matrix (PxP) of regression coefficients. Default is `diag(1e-2, P)`, where P refers to the number of columns in the model matrix.
n_burn	Integer. Number of burn-in iterations. Default is 10000.
n_it	Integer. Number of MCMC iterations. Default is 20000.

`n_thin` Integer. Thinning interval. Default is 1.

`n_chain` Integer. Number of MCMC chains to be sampled. Default is 2.

`start` Numeric vector. Starting values for chains. Should have length `n_chain`. Default is c(0.5, 1).

`return_diagnostics` Logical. Return MCMC diagnostics? Default is TRUE.

`return_diagnostic_plots` Logical. Should MCMC chains for diagnostic plots (traceplots, autocorrelation, density) be returned? Default is `FALSE`. If `TRUE`, diagnostics are returned by default.

`return_samples` Logical. Should posterior samples be returned? Default is FALSE.

Value

An object of class `bglm`, a list containing:

estimates A list with posterior means and standard deviations of group probabilities (`mean_a`, `mean_b`, `sd_a`, `sd_b`), as well as posterior means (`b`) and standard deviations (`b_sd`) of the regression coefficients.

sample_sizes A list with group sample sizes (`n_a`, `n_b`).

delta A list with posterior mean differences (`mean_delta`), posterior standard errors (`se_delta`), posterior probability of the hypothesis (`pop`), and, if `rule = "Comp"`, the weighted difference (`w_delta`).

info A list with prior specifications, test settings, group labels, covariate handling method, and subpopulation definition.

diags If diagnostics are requested, a list with MCMC diagnostic results for the regression coefficients.

samples If `return_samples = TRUE`, a list containing posterior draws of `theta_a`, `theta_b`, `delta`, and regression coefficients.

References

Kavelaars X, Mulder J, Kaptein M (2024). "Bayesian Multivariate Logistic Regression for Superiority and Inferiority Decision-Making under Observable Treatment Heterogeneity." *Multivariate Behavioral Research*, **59**(4), 859–882. doi:10.1080/00273171.2024.2337340.

Examples

```
# Example with simulated data
# Generate data
set.seed(123)
n <- 100

data <- data.frame(
 group = rep(c("A", "B"), each = n/2),
 x = rnorm(n),
 stringsAsFactors = FALSE
```

```
)

p1 <- p2 <- rep(NA, n)

for (i in 1:n) {
 grpB <- ifelse(data$group[i] == "B", 1, 0)

 p1[i] <- plogis(-0.50 + 0.75 * grpB + 0.10 * data$x[i] + 0.20 * grpB * data$x[i])
 p2[i] <- plogis(-0.50 + 0.80 * grpB + 0.05 * data$x[i] + 0.15 * grpB * data$x[i])

 data$y1[i] <- rbinom(1, 1, p1[i])
 data$y2[i] <- rbinom(1, 1, p2[i])
}

# Analyze
result <- bglm(
 data = data,
 grp = "group",
 grp_a = "A",
 grp_b = "B",
 x_var = "x",
 y_vars = c("y1", "y2"),
 x_method = "Empirical",
 x_def = c(-Inf, Inf),
 test = "right_sided",
 rule = "All",
 n_burn = 100, # Too low for proper MCMC sampling
 n_it = 500 # Too low for proper MCMC sampling
)

print(result)
```

`bglm_data` *Simulated Single-Level Clinical Trial Data*

Description

A simulated dataset representing a two-arm clinical trial with 200 subjects, one continuous covariate, and two binary outcomes. It serves as the underlying data for `bglm_fit` and can be used to illustrate `bglm`.

Usage

```
bglm_data
```

Format

A data frame with 200 rows and 4 columns:

group Character. Treatment arm: `"placebo"` (n = 100) or `"drug"` (n = 100).

age Numeric. Continuous covariate drawn from $N(50,\ 10^2)$.

y1 Integer (0/1). First binary outcome.

y2 Integer (0/1). Second binary outcome.

Details

Data were generated with `set.seed(2024)` using logistic models:

$$P(y_1 = 1) = \text{logit}^{-1}(-0.50 + 0.75\,\text{drug} + 0.10\,\text{age}/10)$$

$$P(y_2 = 1) = \text{logit}^{-1}(-0.50 + 0.80\,\text{drug} + 0.05\,\text{age}/10)$$

where drug is 1 for the drug arm and 0 for placebo. See `data-raw/generate_examples.R` for the full script.

See Also

`bglm`, `bglm_fit`, `bglmm_data`

Examples

```
head(bglm_data)
table(bglm_data$group, bglm_data$y1)
```

---

bglm_fit *Pre-computed bglm Example Fit*

Description

A fitted `bglm` object estimated on `bglm_data`. Used in package examples and tests so that `print`, `summary`, and `plot` examples run in well under 5 seconds without re-running the MCMC sampler.

Usage

```
bglm_fit
```

Format

An object of class `bglm` as returned by `bglm`. See the **Value** section of `bglm` for a full description of the list components. Key settings: `n_burn` = `10000`, `n_it` = `20000`, `n_chain` = `2`, `return_diagnostics` = `TRUE`, `return_samples` = `TRUE`.

Details

Generated with `set.seed(2024)`. See `data-raw/generate_examples.R` for the full reproducible script.

See Also

bglm, bglm_data, bglmm_fit

Examples

```
print(bglm_fit)
summary(bglm_fit)
```

---

bglmm *Bayesian Generalized Linear Mixed Model*

Description

Perform a Bayesian test for differences between two (sub)groups on multiple binary outcomes using multilevel multinomial logistic regression, as described in Kavelaars et al. (2023).

Usage

```
bglmm(
  data,
  grp,
  grp_a = NULL,
  grp_b = NULL,
  id_var,
  x_var,
  y_vars,
  x_method = c("Empirical", "Analytical", "Value"),
  x_def = c(-Inf, Inf),
  test = c("right_sided", "left_sided"),
  rule = c("All", "Any", "Comp"),
  w = NULL,
  n_burn = 10000,
  n_it = 50000,
  start = c(0.5, 1),
  fixed = NULL,
  random = NULL,
  b_mu0 = NULL,
  b_sigma0 = NULL,
  g_mu0 = NULL,
  g_sigma0 = NULL,
  nu0 = NULL,
  tau0 = NULL,
  n_chain = 2,
  return_thinned = TRUE,
  n_thin = 10,
  return_diagnostics = TRUE,
```

```
  return_diagnostic_plots = FALSE,
  return_samples = FALSE
)
```

Arguments

data	Data frame containing the data.
grp	Character string. Name of the grouping variable.
grp_a	Value of grp indicating first group (will be determined from factor levels if NULL).
grp_b	Value of grp indicating second group (will be determined from factor levels if NULL).
id_var	Character string. Name of cluster/ID variable.
x_var	Character string. Name of covariate variable.
y_vars	Character vector. Names of outcome variables (currently supports 2 outcomes).
x_method	Character. Method for handling covariate. Default is "Empirical".
x_def	Numeric. Defines subpopulation. Default is c(-Inf, Inf).
test	Character. Direction of test: "left_sided" for P(A>B) or "right_sided" for P(B>A). Default is "right_sided".
rule	Character. Decision rule: "All" (all outcomes favor hypothesis), "Any" (at least one outcome favors hypothesis), or "Comp" (weighted combination). Default is "All".
w	Numeric vector. Weights for compensatory rule. Only used if rule = "Comp". If NULL and rule = "Comp", equal weights are used. Default is NULL.
n_burn	Integer. Number of burn-in iterations. Default is 10000.
n_it	Integer. Number of MCMC iterations. Default is 50000 (takes long running time!).
start	Numeric vector. Starting values for chains. Default is c(0.5, 1).
fixed	Character vector. Names of fixed effect variables. Default is c(x_var, grp_x_var).
random	Character vector. Names of random effect variables. Default is c("Intercept", grp).
b_mu0	Numeric vector. Prior means for fixed effects. Default is rep(0, length(fixed)).
b_sigma0	Matrix. Prior covariance for fixed effects. Default is diag(0.1, length(fixed)).
g_mu0	Numeric vector. Prior means for random effects. Default is rep(0, length(random)).
g_sigma0	Matrix. Prior covariance for random effects. Default is diag(0.1, length(random)).
nu0	Numeric. Prior degrees of freedom for inverse-Wishart. Default is length(random).
tau0	Matrix. Prior scale matrix of dimension length(random) x length(random) for inverse-Wishart. Default is diag(1e-1, length(random)).
n_chain	Integer. Number of MCMC chains. Default is 2.
return_thinned	Logical. Return thinned chains? Default is TRUE.
n_thin	Integer. Thinning interval. Default is 10.

`return_diagnostics`
: Logical. Return MCMC diagnostics? Default is TRUE.

`return_diagnostic_plots`
: Logical. Should MCMC chains for diagnostic plots (traceplots, autocorrelation, density) be returned? Default is `FALSE`. If `TRUE`, diagnostics are returned by default.

`return_samples`
: Logical. Return posterior samples? Default is FALSE.

Value

An object of class `bglmm`, a list containing:

estimates A list with posterior means and standard deviations of group probabilities (`mean_a`, `mean_b`, `sd_a`, `sd_b`). If estimated, posterior means and standard deviations of fixed effects (`b`, `b_sd`) and random effects and variance components (`g`, `g_sd`, `tau`, `tau_sd`) are included.

sample_sizes A list with group sample sizes (`n_a`, `n_b`) and the number of clusters (`J`).

delta A list with posterior mean differences (`mean_delta`), posterior standard errors (`se_delta`), posterior probability of the hypothesis (`pop`), and, if `rule = "Comp"`, the weighted difference (`w_delta`).

info A list with prior specifications, model structure (fixed and random effects), test settings, group labels, covariate handling method, and subpopulation definition.

diags If diagnostics are requested, a list with MCMC diagnostic results for fixed effects, random effects, and variance components.

samples If `return_samples = TRUE`, a list containing posterior draws of group probabilities, differences, fixed effects, random effects, and variance components (if applicable).

References

Kavelaars X, Mulder J, Kaptein M (2023). "Bayesian multilevel multivariate logistic regression for superiority decision-making under observable treatment heterogeneity." *BMC Medical Research Methodology*, **23**(1). doi:10.1186/s1287402302034z.

Examples

```
# Example with simulated data
# Generate data
set.seed(123)
J <- 20 # No. clusters
nJ <- 15 # Sample size per cluster

# Generate random intercepts
uj_1 <- rnorm(J)
uj_2 <- rnorm(J)
data <- data.frame(
 id = factor(rep(1:J, each = nJ)),
 group = rep(rep(c("A", "B"), each = J/2), each = nJ),
 x = rnorm(J * nJ),
 stringsAsFactors = FALSE
)
```

```
p1 <- p2 <- rep(NA, J * nJ)

for (i in 1:(J * nJ)) {
 j <- as.numeric(data$id[i])
 grpB <- ifelse(data$group[i] == "B", 1, 0)

 p1[i] <- plogis(-0.50 + 0.75 * grpB + 0.10 * data$x[i] + 0.20 * grpB * data$x[i] + uj_1[j])
 p2[i] <- plogis(-0.50 + 0.80 * grpB + 0.05 * data$x[i] + 0.15 * grpB * data$x[i] + uj_2[j])

 data$y1[i] <- rbinom(1, 1, p1[i])
 data$y2[i] <- rbinom(1, 1, p2[i])
}

# Analyze
result <- bglmm(
 data = data,
 grp = "group",
 grp_a = "A",
 grp_b = "B",
 id_var = "id",
 x_var = "x",
 y_vars = c("y1", "y2"),
 x_method = "Empirical",
 x_def = c(-Inf, Inf),
 fixed = c("group", "x", "group_x"),
 random = c("Intercept"), # Random intercept model
 test = "right_sided",
 rule = "All",
 n_burn = 100, # Too low for proper MCMC sampling
 n_it = 500 # Too low for proper MCMC sampling
 )

print(result) # Warnings due to low number of MCMC iterations (n_burn and n_it)
```

`bglmm_data` *Simulated Multilevel Clinical Trial Data*

Description

A simulated dataset representing a two-arm clinical trial with 300 subjects nested within 20 clusters (e.g., hospitals), one continuous covariate, and two binary outcomes. It serves as the underlying data for `bglmm_fit` and can be used to illustrate `bglmm`.

Usage

```
bglmm_data
```

Format

A data frame with 300 rows and 5 columns:

id Factor with 20 levels (1–20). Cluster identifier (e.g., hospital). Each cluster contains 15 subjects.

group Character. Treatment arm: `"placebo"` (clusters 1–10) or `"drug"` (clusters 11–20).

age Numeric. Continuous covariate drawn from $N(50,\ 10^2)$.

y1 Integer (0/1). First binary outcome.

y2 Integer (0/1). Second binary outcome.

Details

Data were generated with `set.seed(2024)` using logistic models with cluster-specific random intercepts $u_{j1},\ u_{j2} \sim N(0,\ 0.25)$:

$$P(y_1 = 1) = \text{logit}^{-1}(-0.50 + 0.75\,\text{drug} + 0.10\,\text{age}/10 + u_{j1})$$

$$P(y_2 = 1) = \text{logit}^{-1}(-0.50 + 0.80\,\text{drug} + 0.05\,\text{age}/10 + u_{j2})$$

where drug is 1 for the drug arm and 0 for placebo. See `data-raw/generate_examples.R` for the full script.

See Also

bglmm, bglmm_fit, bglm_data

Examples

```
head(bglmm_data)
table(bglmm_data$group, bglmm_data$id)
```

---

bglmm_fit *Pre-computed bglmm Example Fit*

Description

A fitted bglmm object estimated on bglmm_data. Used in package examples and tests so that `print`, `summary`, and `plot` examples run in well under 5 seconds without re-running the MCMC sampler.

Usage

```
bglmm_fit
```

Format

An object of class `bglmm` as returned by bglmm. See the **Value** section of bglmm for a full description of the list components. Key settings: `n_burn` = `10000`, `n_it` = `50000`, `n_thin` = `10`, `n_chain` = `2`, `return_diagnostics` = `TRUE`, `return_samples` = `TRUE`.

Details

Generated with `set.seed(2024)`. See `data-raw/generate_examples.R` for the full reproducible script.

See Also

`bglmm`, `bglmm_data`, `bglm_fit`

Examples

```
print(bglmm_fit)
summary(bglmm_fit)
```

---

bmvb *Bayesian Multivariate Bernoulli Test*

Description

Perform a Bayesian test for differences between two groups on multiple binary outcomes using a Multivariate Bernoulli distribution, as described in Kavelaars et al. (2020).

Usage

```
bmvb(
  data,
  grp,
  grp_a,
  grp_b,
  y_vars,
  test = c("right_sided", "left_sided"),
  rule = c("All", "Any", "Comp"),
  w = NULL,
  prior_a = 0.5,
  prior_b = 0.5,
  n_it = 10000,
  return_samples = FALSE
)
```

Arguments

`data`	Data frame containing the data.
`grp`	Character string. Name of the grouping variable.
`grp_a`	Value of grp indicating first group.
`grp_b`	Value of grp indicating second group.
`y_vars`	Character vector. Names of outcome variables (currently supports 2 outcomes).

test	Character. Direction of test: "left_sided" for P(A>B) or "right_sided" for P(B>A). Default is "right_sided".
rule	Character. Decision rule: "All" (all outcomes favor hypothesis), "Any" (at least one outcome favors hypothesis), or "Comp" (weighted combination). Default is "All".
w	Numeric vector. Weights for compensatory rule. Only used if rule = "Comp". If NULL and rule = "Comp", equal weights are used. Default is NULL.
prior_a	Numeric. Prior hyperparameter (Dirichlet) for group A. Default is 0.5 (Jeffreys' prior)
prior_b	Numeric. Prior hyperparameter (Dirichlet) for group B. Default is 0.5 (Jeffreys' prior).
n_it	Integer. Number of MCMC iterations. Default is 10000.
return_samples	Logical. Should posterior samples be returned? Default is FALSE.

Value

An object of class bmvb, a list containing:

estimates A list with posterior means (mean_a, mean_b) and standard deviations (sd_a, sd_b) of the category probabilities for both groups.

sample_sizes A list with group sample sizes (n_a, n_b).

delta A list with posterior mean differences (mean_delta), posterior standard errors (se_delta), posterior probability of the hypothesis (pop), and, if rule = "Comp", the weighted difference (w_delta).

info A list with test specifications, including the decision rule, test direction, group labels, and weights (if applicable).

samples If return_samples = TRUE, a list containing posterior draws of theta_a, theta_b, and delta.

References

Kavelaars X, Mulder J, Kaptein M (2020). "Decision-making with multiple correlated binary outcomes in clinical trials." *Statistical Methods in Medical Research*, **29**(11), 3265–3277. doi:10.1177/0962280220922256.

Examples

```
# Example with simulated data
# Generate data
set.seed(123)
data <- data.frame(
treatment = rep(c("control", "drug"), each = 50),
 outcome1 = rbinom(100, 1, 0.5),
 outcome2 = rbinom(100, 1, 0.5)
)

# Analyze
```

```
result <- bmvb(
 data = data,
 grp = "treatment",
 grp_a = "control",
 grp_b = "drug",
 y_vars = c("outcome1", "outcome2"),
 n_it = 10000
)

print(result)
```

---

plot.bglm *Plot Method for bglm Objects*

Description

Plot Method for bglm Objects

Usage

```
## S3 method for class 'bglm'
plot(x, type = "all", parameters = NULL, ...)
```

Arguments

x	A bglm object returned by bglm().
type	Character. Type of plot: "trace", "density", "autocorr", or "all". Default is "all".
parameters	Character vector. Which parameters to plot. Default is NULL (all parameters).
...	Additional arguments passed to plotting functions.

Value

Invisibly returns NULL (plots are displayed).

Examples

```
# Uses the pre-computed example object shipped with the package.
# Plot trace plots for the fixed-effect regression coefficients:
plot(bglm_fit, type = "trace")
```

plot.bglmm *Plot Method for bglmm Objects*

Description

Plot Method for bglmm Objects

Usage

```
## S3 method for class 'bglmm'
plot(x, type = "all", which = "fixed", parameters = NULL, ...)
```

Arguments

x	A bglmm object returned by bglmm().
type	Character. Type of plot: "trace", "density", "autocorr", or "all". Default is "all".
which	Character. Which component to plot: "fixed" (b), "random" (g), "variance" (tau), or "all". Default is "fixed".
parameters	Character vector. Which parameters to plot. Default is NULL (all parameters).
...	Additional arguments passed to plotting functions.

Value

Invisibly returns NULL (plots are displayed).

Examples

```
# Uses the pre-computed example object shipped with the package.
# Trace plots for the fixed-effect regression coefficients:
plot(bglmm_fit, type = "trace", which = "fixed")

# Trace plots for the random-effect variance components:
plot(bglmm_fit, type = "trace", which = "variance")
```

print.bglm *Print Method for bglm Objects*

Description

Print Method for bglm Objects

Usage

```
## S3 method for class 'bglm'
print(x, digits = 3, ...)
```

Arguments

x	A bglm object.
digits	Number of digits to display. Default is 3.
...	Additional arguments (not used).

Value

Invisibly returns the input object.

See Also

summary.bglm

Examples

```
# Uses the pre-computed example object shipped with the package:
print(bglm_fit)
```

---

print.bglmm *Print Method for bglmm Objects*

Description

Print Method for bglmm Objects

Usage

```
## S3 method for class 'bglmm'
print(x, digits = 3, ...)
```

Arguments

x	A bglmm object.
digits	Number of digits to display. Default is 3.
...	Additional arguments (not used).

Value

Invisibly returns the input object.

See Also

summary.bglmm

Examples

```
# Uses the pre-computed example object shipped with the package:
print(bglmm_fit)
```

`print.bmvb` *Print Method for bmvb Objects*

Description

Print Method for bmvb Objects

Usage

```
## S3 method for class 'bmvb'
print(x, digits = 3, ...)
```

Arguments

`x`	A bmvb object.
`digits`	Number of digits to display. Default is 3.
`...`	Additional arguments (not used).

Value

Invisibly returns the input object.

See Also

`summary.bmvb`

Examples

```
set.seed(2024)
trial_data <- data.frame(
  treatment = rep(c("placebo", "drug"), each = 50),
  y1 = rbinom(100, 1, rep(c(0.40, 0.60), each = 50)),
  y2 = rbinom(100, 1, rep(c(0.50, 0.70), each = 50))
)
fit <- bmvb(
  data = trial_data, grp = "treatment",
  grp_a = "placebo", grp_b = "drug",
  y_vars = c("y1", "y2"), n_it = 1000
)
print(fit)
```

print.summary.bglm *Print Method for summary.bglm Objects*

Description

Print Method for summary.bglm Objects

Usage

```
## S3 method for class 'summary.bglm'
print(x, digits = 3, ...)
```

Arguments

x	A summary.bglm object returned by summary.bglm.
digits	Number of digits to display. Default is 3.
...	Additional arguments (not used).

Value

Invisibly returns x.

See Also

summary.bglm, print.bglm

Examples

```
# Uses the pre-computed example object shipped with the package:
print(summary(bglm_fit))
```

print.summary.bglmm *Print Method for summary.bglmm Objects*

Description

Print Method for summary.bglmm Objects

Usage

```
## S3 method for class 'summary.bglmm'
print(x, digits = 3, ...)
```

Arguments

x	A summary.bglmm object returned by summary.bglmm.
digits	Number of digits to display. Default is 3.
...	Additional arguments (not used).

Value

Invisibly returns x.

See Also

summary.bglmm, print.bglmm

Examples

```
# Uses the pre-computed example object shipped with the package:
print(summary(bglmm_fit))
```

---

print.summary.bmvb *Print Method for summary.bmvb Objects*

Description

Print Method for summary.bmvb Objects

Usage

```
## S3 method for class 'summary.bmvb'
print(x, digits = 3, ...)
```

Arguments

x	A summary.bmvb object returned by summary.bmvb.
digits	Number of digits to display. Default is 3.
...	Additional arguments (not used).

Value

Invisibly returns x.

See Also

summary.bmvb, print.bmvb

Examples

```
set.seed(2024)
trial_data <- data.frame(
  treatment = rep(c("placebo", "drug"), each = 50),
  y1 = rbinom(100, 1, rep(c(0.40, 0.60), each = 50)),
  y2 = rbinom(100, 1, rep(c(0.50, 0.70), each = 50))
)
fit <- bmvb(
  data = trial_data, grp = "treatment",
  grp_a = "placebo", grp_b = "drug",
  y_vars = c("y1", "y2"), n_it = 1000,
  return_samples = TRUE
)
print(summary(fit))
```

---

summary.bglm *Summary Method for bglm Objects*

Description

Provides a comprehensive summary of a `bglm` analysis, including the regression coefficient table, prior specification, MCMC diagnostics (effective sample sizes and $\hat{R}$ per parameter), and, when the model was fitted with `return_samples = TRUE`, credible intervals.

Usage

```
## S3 method for class 'bglm'
summary(object, prob = 0.95, ...)
```

Arguments

`object`	A `bglm` object returned by `bglm`.
`prob`	Numeric. Coverage probability for credible intervals. Default is `0.95`.
`...`	Additional arguments (not used).

Value

An object of class `summary.bglm`, a list containing:

estimates Posterior means and SDs of group probabilities and regression coefficients.

sample_sizes Group sample sizes.

delta Posterior mean differences, SEs, and posterior probability.

info Prior specification, test settings, and marginalization details.

credible_intervals If posterior samples are available: credible interval matrices for `theta_a`, `theta_b`, and `delta`.

effective_n If posterior samples are available: effective sample sizes for `theta_a`, `theta_b`, and `delta`.

mcmc_diags MCMC diagnostics for the regression coefficients (effective sample sizes and $\hat{R}$).

See Also

bglm, print.bglm

Examples

```
# Uses the pre-computed example object shipped with the package:
summary(bglm_fit)
```

---

summary.bglmm *Summary Method for bglmm Objects*

Description

Provides a comprehensive summary of a bglmm analysis, including fixed and random effect tables, variance component estimates, multilevel structure, and MCMC convergence diagnostics.

Usage

```
## S3 method for class 'bglmm'
summary(object, prob = 0.95, ...)
```

Arguments

object	A bglmm object returned by bglmm.
prob	Numeric. Coverage probability for credible intervals. Default is 0.95.
...	Additional arguments (not used).

Value

An object of class summary.bglmm, a list containing:

estimates Posterior means and SDs of group probabilities, fixed effects, random effects, and variance components.

sample_sizes Group sample sizes and number of clusters.

delta Posterior mean differences, SEs, and posterior probability.

info Prior specification, model structure, test settings, and marginalization details.

credible_intervals If posterior samples are available: credible interval matrices for theta_a, theta_b, and delta.

effective_n If posterior samples are available: effective sample sizes for theta_a, theta_b, and delta.

mcmc_diags MCMC convergence diagnostics (ESS, $\hat{R}$, MPSRF) for fixed effects, random effects, and variance components.

See Also

bglmm, print.bglmm

Examples

```
# Uses the pre-computed example object shipped with the package:
summary(bglmm_fit)
```

---

summary.bmvb *Summary Method for bmvb Objects*

Description

Provides a comprehensive summary of a bmvb analysis. When the model was fitted with `return_samples = TRUE`, credible intervals and effective sample sizes are included.

Usage

```
## S3 method for class 'bmvb'
summary(object, prob = 0.95, ...)
```

Arguments

object	A bmvb object returned by bmvb.
prob	Numeric. Coverage probability for credible intervals. Default is 0.95.
...	Additional arguments (not used).

Value

An object of class summary.bmvb, a list containing all fields of object plus:

credible_intervals If posterior samples are available: a list with prob and credible interval matrices for theta_a, theta_b, and delta.

effective_n If posterior samples are available: a list with effective sample sizes for theta_a, theta_b, and delta.

See Also

bmvb, print.bmvb

Examples

```
set.seed(2024)
trial_data <- data.frame(
  treatment = rep(c("placebo", "drug"), each = 50),
  y1 = rbinom(100, 1, rep(c(0.40, 0.60), each = 50)),
  y2 = rbinom(100, 1, rep(c(0.50, 0.70), each = 50))
)
fit <- bmvb(
  data = trial_data, grp = "treatment",
  grp_a = "placebo", grp_b = "drug",
  y_vars = c("y1", "y2"), n_it = 1000,
  return_samples = TRUE
)
summary(fit)
```

Index

∗ **datasets**
bglm_data, 6
bglm_fit, 7
bglmm_data, 11
bglmm_fit, 12

bglm, *2*, 3, *6–8*, *21, 22*
bglm_data, 6, *7, 8*, *12*
bglm_fit, *6*, 7, *7*, *13*
bglmm, *2*, 8, *11–13*, *22, 23*
bglmm_data, *7*, 11, *12, 13*
bglmm_fit, *8*, *11*, 12, *12*
bmco *(*bmco-package*)*, 2
bmco-package, 2
bmvb, *2*, 13, *23*

plot.bglm, 15
plot.bglmm, 16
print.bglm, 16, *19*, *22*
print.bglmm, 17, *20*, *23*
print.bmvb, 18, *20*, *23*
print.summary.bglm, 19
print.summary.bglmm, 19
print.summary.bmvb, 20

summary.bglm, *17*, *19*, 21
summary.bglmm, *17*, *20*, 22
summary.bmvb, *18*, *20*, 23