Package 'multilevelcoda'

November 23, 2024

Type Package

Title Estimate Bayesian Multilevel Models for Compositional Data

Version 1.3.1

Date 2024-11-11

URL https://florale.github.io/multilevelcoda/,
  https://github.com/florale/multilevelcoda

BugReports https://github.com/florale/multilevelcoda/issues

Description Implement Bayesian Multilevel Modelling for compositional data
  in a multilevel framework. Compute multilevel compositional data and
  Isometric log ratio (ILR) at between and within-person levels,
  fit Bayesian multilevel models for compositional predictors and outcomes,
  and run post-hoc analyses such as isotemporal substitution models.
  References:
  Le, Stanford, Dumuid, and Wiley (2024) <doi:10.48550/arXiv.2405.03985>,
  Le, Dumuid, Stanford, and Wiley (2024) <doi:10.48550/arXiv.2411.12407>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 4.0.0)

Imports stats, data.table (>= 1.12.0), compositions, brms, bayestestR,
  extraoperators, ggplot2, foreach, future, doFuture, abind,
  graphics, shiny, shinystan, plotly, hrbrthemes, bslib, DT, loo,
  bayesplot, emmeans, posterior

Suggests testthat (>= 3.0.0), covr, withr, knitr, rmarkdown, lme4,
  cmdstanr (>= 0.5.0)

Config/testthat/edition 3

Config/testthat/parallel true

Additional_repositories https://mc-stan.org/r-packages/

VignetteBuilder knitr

Contents

as.data.frame.complr   *Extract Compositional Data from* complr *object.*

Description

Extract amounts and compositions in conventional formats as data.frames, matrices, or arrays.

Usage

```
## S3 method for class 'complr'
as.data.frame(x, row.names = NULL, optional = TRUE, ...)

## S3 method for class 'complr'
as.matrix(x, ...)
```

Arguments

x   An object of class complr.

row.names, optional

                Unused and only added for consistency with the as.data.frame generic.

...   generic argument, not in use.

---

bayes_factor.brmcoda   *Bayes Factors from Marginal Likelihoods*

Description

Compute Bayes factors from marginal likelihoods

Usage

```
## S3 method for class 'brmcoda'
bayes_factor(x1, x2, ...)
```

Arguments

x1	A brmcoda object.
x2	Another brmcoda object based on the same responses.
...	Other arguments passed to `bayes_factor.brmsfit`.

See Also

`bayes_factor.brmsfit`

---

brmcoda   *Fit Bayesian generalised (non-)linear multilevel compositional model via full Bayesian inference*

Description

Fit a `brm` model with multilevel ILR coordinates

Usage

```
brmcoda(complr, formula, ...)
```

Arguments

complr	A `complr` object containing data of composition, ILR coordinates, and other variables used in the model.
formula	A object of class formula, brmsformula: A symbolic description of the model to be fitted. Details of the model specification can be found in `brmsformula`.
...	Further arguments passed to `brm`.

Value

A [brmcoda](#) with two elements

complr	An object of class complr used in the brm model.
model	An object of class brmsfit, which contains the posterior draws along with many other useful information about the model.

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- complr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID")

  # inspects ILRs before passing to brmcoda
  names(cilr$between_logratio)
  names(cilr$within_logratio)
  names(cilr$logratio)

  # model with compositional predictor at between and within-person levels
  m1 <- brmcoda(complr = cilr,
                formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                   wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
                chain = 1, iter = 500,
                backend = "cmdstanr")

  # model with compositional outcome
  m2 <- brmcoda(complr = cilr,
                formula = mvbind(ilr1, ilr2, ilr3, ilr4) ~ Stress + Female + (1 | ID),
                chain = 1, iter = 500,
                backend = "cmdstanr")
}
```

---

bsub   *Between-person Simple Substitution*

Description

This function is an alias of [substitution](#) to estimates the the difference in an outcome when compositional parts are substituted for specific unit(s) at *between* level using a single reference composition (e.g., compositional mean at sample level). It is recommended that users run substitution model using the [substitution](#) function.

Usage

```
bsub(
  object,
  delta,
  basesub,
```

```
  summary = TRUE,
  ref = "grandmean",
  level = "between",
  weight = "equal",
  scale = c("response", "linear"),
  cores = NULL,
  ...
)
```

Arguments

`object`	A fitted [brmcoda](#) object.
`delta`	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
`basesub`	A base substitution. Can be a `data.frame` or `data.table` of the base possible substitution of compositional parts, which can be computed using function [build.basesub](#). If `"one-to-all"`, all possible one-to-remaining reallocations are estimated. If NULL, all possible one-to-one reallocations are estimated.
`summary`	A logical value. Should the estimate at each level of the reference grid (FALSE) or their average (TRUE) be returned? Default is TRUE. Only applicable for model with covariates in addition to the isometric log-ratio coordinates (i.e., adjusted model).
`ref`	Either a character value or vector or a dataset. Can be `"grandmean"` and/or `"clustermean"`, or a `data.frame` or `data.table` of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to `"grandmean"`.
`level`	A character string or vector. Should the estimate of multilevel models focus on the `"between"` and/or `"within"` or `"aggregate"` variance? Single-level models are default to `"aggregate"`.
`weight`	A character value specifying the weight to use in calculation of the reference composition. If `"equal"`, give equal weight to units (e.g., individuals). If `"proportional"`, weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default to `"equal"` for ref = `"grandmean"` and `"proportional"` for ref = `"clustermean"`.
`scale`	Either `"response"` or `"linear"`. If `"response"`, results are returned on the scale of the response variable. If `"linear"`, results are returned on the scale of the linear predictor term, that is without applying the inverse link function or other transformations.
`cores`	Number of cores to use when executing the chains in parallel, we recommend setting the `mc.cores` option to be as many processors as the hardware and RAM allow (up to the number of compositional parts). For non-Windows OS in non-interactive R sessions, forking is used instead of PSOCK clusters.
`...`	currently ignored.

Value

A list containing the results of multilevel compositional substitution model. The first six lists contain the results of the substitution estimation for a compositional part.

Mean	Posterior means.
CI_low and CI_high	
	95% credible intervals.
Delta	Amount substituted across compositional parts.
From	Compositional part that is substituted from.
To	Compositional parts that is substituted to.
Level	Level where changes in composition takes place.
Reference	Either grandmean, clustermean, or users.

See Also

substitution

Examples

```
if(requireNamespace("cmdstanr")){
cilr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID", total = 1440)

# model with compositional predictor at between and between-person levels
m <- brmcoda(complr = cilr,
             formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                wilr1 + wilr2 + wilr3 + wilr4 + Female + (1 | ID),
             chain = 1, iter = 500,
             backend = "cmdstanr")
subm <- bsub(object = m, basesub = psub, delta = 5)
}
```

---

bsubmargins	*Between-person Average Substitution*

Description

This function is an alias of substitution to estimates the the difference in an outcome when compositional parts are substituted for specific unit(s) at *between* level using cluster mean (e.g., compositional mean at individual level) as reference composition. It is recommended that users run substitution model using the substitution function.

Usage

```
bsubmargins(
  object,
  delta,
  basesub,
  ref = "clustermean",
  level = "between",
  weight = "proportional",
  scale = c("response", "linear"),
  cores = NULL,
  ...
)
```

Arguments

object	A fitted [brmcoda](#) object.
delta	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
basesub	A base substitution. Can be a data.frame or data.table of the base possible substitution of compositional parts, which can be computed using function [build.basesub](#). If "one-to-all", all possible one-to-remaining reallocations are estimated. If NULL, all possible one-to-one reallocations are estimated.
ref	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a data.frame or data.table of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to "grandmean".
level	A character string or vector. Should the estimate of multilevel models focus on the "between" and/or "within" or "aggregate" variance? Single-level models are default to "aggregate".
weight	A character value specifying the weight to use in calculation of the reference composition. If "equal", give equal weight to units (e.g., individuals). If "proportional", weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default to "equal" for ref = "grandmean" and "proportional" for ref = "clustermean".
scale	Either "response" or "linear". If "response", results are returned on the scale of the response variable. If "linear", results are returned on the scale of the linear predictor term, that is without applying the inverse link function or other transformations.
cores	Number of cores to use when executing the chains in parallel, we recommend setting the mc.cores option to be as many processors as the hardware and RAM allow (up to the number of compositional parts). For non-Windows OS in non-interactive R sessions, forking is used instead of PSOCK clusters.
...	currently ignored.

Value

A list containing the results of multilevel compositional substitution model. The first six lists contain the results of the substitution estimation for a compositional part.

`Mean`	Posterior means.
`CI_low` and `CI_high`	
	95% credible intervals.
`Delta`	Amount substituted across compositional parts.
`From`	Compositional part that is substituted from.
`To`	Compositional parts that is substituted to.
`Level`	Level where changes in composition takes place.
`Reference`	Either `grandmean`, `clustermean`, or `users`.

See Also

substitution

Examples

```
if(requireNamespace("cmdstanr")){
cilr <- complr(data = mcompd[ID %in% 1:10, .SD[1:3], by = ID], sbp = sbp,
                parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID", total = 1440)

m <- brmcoda(complr = cilr,
             formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 + wilr1 +
                                wilr2 + wilr3 + wilr4 + Female + (1 | ID),
             chains = 1, iter = 500,
             backend = "cmdstanr")

subm <- bsubmargins(object = m, basesub = psub, delta = 5)
}
```

---

build.basesub	*Build Base Pairwise Substitution*

Description

Make a data set of all possible pairwise substitution of a composition which can be used as the base for substitution models.

Usage

```
build.basesub(parts, comparison = NULL)
```

Arguments

`parts`	A character vector specifying the names of compositional variables to be used.
`comparison`	Either `"one-to-one"` or `"one-to-all"`. Default is `"one-to-one"`.

Value

A data table of all possible pairwise substitution.

Examples

```
ps1 <- build.basesub(parts = c("TST", "WAKE", "MVPA", "LPA", "SB"))
print(ps1)

ps2 <- build.basesub(c("WAKE", "MVPA", "LPA", "SB"), comparison = "one-to-all")
print(ps2)
```

---

build.rg   *Reference Grid for* substitution *model.*

Description

Build a dataset for `fitted.brmcoda` used in `substitution` model

Usage

```
build.rg(object, ref, level, weight, fill = FALSE)
```

Arguments

`object`	A fitted brmcoda object.
`ref`	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a `data.frame` or `data.table` of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to "grandmean".
`level`	A character string or vector. Should the estimate of multilevel models focus on the "between" and/or "within" or "aggregate" variance? Single-level models are default to "aggregate".
`weight`	A character value specifying the weight to use in calculation of the reference composition. If "equal", give equal weight to units (e.g., individuals). If "proportional", weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default to "equal" for ref = "grandmean" and "proportional" for ref = "clustermean".
`fill`	Logical value only relevant when `ref` is an user's specified reference grid in which information about some, but not all covariates is provided (e.g., models including age and sex as covariate but only age was provided in the reference grid). If TRUE, the unspecified covariates are filled with the default reference grid. If FALSE, users will be asked to provide a full reference grid. Currently only support the default to FALSE.

Value

A reference grid consisting of a combination of covariates in brmcoda

---

build.sbp   *Build Sequential Binary Partition*

Description

Build a default sequential binary partition for `complr` object. The default sequential binary partition is a pivot balance that allows the effect of this first balance coordinate to be interpreted as the change in the prediction for the dependent variable when that given part increases while all remaining parts decrease by a common proportion.

Usage

```
build.sbp(parts)
```

Arguments

parts   A character vector specifying the names of compositional variables to be used.

Value

A matrix sequential binary partition.

Examples

```
sbp1 <- build.sbp(parts = c("TST", "WAKE", "MVPA", "LPA", "SB"))
print(sbp1)

sbp2 <- build.sbp(c("WAKE", "MVPA", "LPA", "SB"))
print(sbp2)
```

---

coef.brmcoda   *Model Coefficients*

Description

Extract model coefficients, which are the sum of population-level effects and corresponding group-level effects of the `brmsfit` object in a `brmcoda` object.

Usage

```
## S3 method for class 'brmcoda'
coef(object, ...)
```

Arguments

object   An object of class brmcoda.

...   Further arguments passed to coef.brmsfit.

Value

A list of 3D arrays (one per grouping factor). If summary is TRUE, the 1st dimension contains the factor levels, the 2nd dimension contains the summary statistics (see [posterior_summary](#)), and the 3rd dimension contains the group-level effects. If summary is FALSE, the 1st dimension contains the posterior draws, the 2nd dimension contains the factor levels, and the 3rd dimension contains the group-level effects.

See Also

[coef.brmsfit](#)

Examples

```
## fit a model
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  ## extract population and group-level coefficients separately
  fixef(m)
  ranef(m)

  ## extract combined coefficients
  coef(m)
}
```

---

compilr   *Indices from a (dataset of) Multilevel Composition(s) (deprecated.)*

Description

Indices from a (dataset of) Multilevel Composition(s) (deprecated.)

Usage

```
compilr(...)
```

Arguments

...	arguments passed to [complr](#).

Value

A [complr](#) object with at least the following elements.

comp	A vector of class acomp representing one closed composition or a matrix of class acomp representing multiple closed compositions each in one row.
between_comp	A vector of class acomp representing one closed between-person composition or a matrix of class acomp representing multiple closed between-person compositions each in one row.
within_comp	A vector of class acomp representing one closed within-person composition or a matrix of class acomp representing multiple closed within-person compositions each in one row.
logratio	Log ratio transform of composition.
between_logratio	
	Log ratio transform of between-person composition.
within_logratio	
	Log ratio transform of within-person composition.
data	The user's dataset or imputed dataset if the input data contains zeros.
transform	Type of transform applied on compositional data.
parts	Names of compositional variables.
idvar	Name of the variable containing IDs.
total	Total amount to which the compositions is closed.

See Also

[complr](#)

---

complr	*Indices from a (dataset of) Multilevel Composition(s)*

Description

Compute sets of compositions and log ratio transformation for multilevel compositional data

Usage

```
complr(data, parts, sbp = NULL, total = 1, idvar = NULL, transform = "ilr")
```

Arguments

data	A data.frame or data.table containing data of all variables used in the analysis. It must include a composition and a ID variable. Required.
parts	A character vector specifying the names of compositional variables to be used.
sbp	A signary matrix indicating sequential binary partition.

total	A numeric value of the total amount to which the compositions should be closed. Default is 1.
idvar	Only for multilevel data, a character string specifying the name of the variable containing IDs.
transform	A character value naming a log ratio transformation to be applied on compositional data. Can be either "ilr" (isometric logratio), "alr" (additive logratio), or "clr" (centered logratio). Default is "ilr".

Details

The *ilr*-transform maps the D-part compositional data from the simplex into non-overlapping subgroups in the (D-1)-dimension Euclidean space isometrically by using an orthonormal basis, thereby preserving the compositional properties and yielding a full-rank covariance matrix. *ilr* transformation should be preferred. However, the *alr* and *clr* are alternatives. The *alr*-transform maps a D-part composition in the Aitchison-simplex non-isometrically to a (D-1)-dimension Euclidian vectors, commonly treating the last part as the common denominator of the others. *alr* transformation does not rely on distance which breaks the constraint of compositional data. *clr*-transform maps a D-part composition in the Aitchison-simplex isometrically to a D-dimensional Euclidian vector subspace. *clr* transformation is not injetive, resulting in singular covariance matrices.

Value

A [complr](#) object with at least the following elements.

comp	A vector of class acomp representing one closed composition or a matrix of class acomp representing multiple closed compositions each in one row.
between_comp	A vector of class acomp representing one closed between-person composition or a matrix of class acomp representing multiple closed between-person compositions each in one row.
within_comp	A vector of class acomp representing one closed within-person composition or a matrix of class acomp representing multiple closed within-person compositions each in one row.
logratio	Log ratio transform of composition.
between_logratio	
	Log ratio transform of between-person composition.
within_logratio	
	Log ratio transform of within-person composition.
data	The user's dataset or imputed dataset if the input data contains zeros.
transform	Type of transform applied on compositional data.
parts	Names of compositional variables.
idvar	Name of the variable containing IDs.
total	Total amount to which the compositions is closed.

Examples

```
cilr <- complr(data = mcompd,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
               idvar = "ID", total = 1440)
str(cilr)

calr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
               idvar = "ID",
               transform = "alr")
str(calr)

cclr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
               idvar = "ID",
                transform = "clr")
str(cclr)

cilr_wide <- complr(data = mcompd[!duplicated(ID)], sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"))
str(cilr_wide)
```

---

create_substitution   *Constructor function for* substitution *class.*

Description

Constructor function for substitution class.

Usage

```
create_substitution(
  between_simple_sub,
  within_simple_sub,
  simple_sub,
  between_avg_sub,
  within_avg_sub,
  avg_sub,
  delta,
  ref,
  level,
  weight,
  parts,
  summary
)
```

Arguments

`between_simple_sub`	
	A list of results from `bsub` or `NULL`
`within_simple_sub`	
	A list of results from `wsub` or `NULL`
`simple_sub`	A list of results from `sub` or `NULL`
`between_avg_sub`	
	A list of results from `bsubmargins` or `NULL`
`within_avg_sub`	A list of results from `wsubmargins` or `NULL`
`avg_sub`	A list of results from `submargins` or `NULL`
`delta`	A numeric vector of the amount of substitution
`ref`	A character value specifying the reference grid
`level`	A character value specifying the level of substitution
`weight`	The weight to use in calculation of the reference composition
`parts`	The parts of the composition
`summary`	A logical value specifying whether to summarize the results

Value

An object of class `substitution`

See Also

substitution

`diagnostic-quantities-brmcoda`

Extract Diagnostic Quantities from `brmsfit` *Models in* `brmcoda`

Description

Extract Diagnostic Quantities from `brmsfit` Models in `brmcoda`

Usage

```
## S3 method for class 'brmcoda'
log_posterior(object, ...)

## S3 method for class 'brmcoda'
nuts_params(object, ...)

## S3 method for class 'brmcoda'
rhat(x, ...)

## S3 method for class 'brmcoda'
neff_ratio(object, ...)
```

Arguments

...	Arguments passed to individual methods (if applicable).
x, object	A brmcoda object or another R object for which the methods are defined.

Value

The exact form of the output depends on the method.

See Also

[log_posterior.brmsfit](#)

[nuts_params.brmsfit](#)

[rhat.brmsfit](#)

[neff_ratio.brmsfit](#)

---

draws-index-brmcoda   *Index* brmcoda *objects*

Description

Index brmcoda objects

Usage

```
## S3 method for class 'brmcoda'
variables(x, ...)

## S3 method for class 'brmcoda'
nvariables(x, ...)

## S3 method for class 'brmcoda'
niterations(x)

## S3 method for class 'brmcoda'
nchains(x)

## S3 method for class 'brmcoda'
ndraws(x)
```

Arguments

x	An object of class brmcoda.
...	Arguments passed to individual methods.

See Also

variables.brmsfit

nvariables.brmsfit

niterations.brmsfit

nchains.brmsfit

ndraws.brmsfit

---

fitted.brmcoda   *Expected Values of the Posterior Predictive Distribution*

Description

Compute posterior draws of the expected value of the posterior predictive distribution of a brmsfit model in the brmcoda object. Can be performed for the data used to fit the model (posterior predictive checks) or for new data. By definition, these predictions have smaller variance than the posterior predictions performed by the predict.brmcoda method. This is because only the uncertainty in the expected value of the posterior predictive distribution is incorporated in the draws computed by fitted while the residual error is ignored there. However, the estimated means of both methods averaged across draws should be very similar.

Usage

```
## S3 method for class 'brmcoda'
fitted(object, scale = c("linear", "response"), summary = TRUE, ...)
```

Arguments

object	An object of class brmcoda.
scale	Specifically for models with compositional responses, either "response" or "linear". If "linear", results are returned on the log-ratio scale. If "response", results are returned on the compositional scale of the response variable.
summary	Should summary statistics be returned instead of the raw values? Default is TRUE.
...	Further arguments passed to fitted.brmsfit that control additional aspects of prediction.

Value

An array of predicted *mean* response values. If summary = FALSE the output resembles those of posterior_epred.brmsfit.

If summary = TRUE the output depends on the family: For categorical and ordinal families, the output is an N x E x C array, where N is the number of observations, E is the number of summary statistics, and C is the number of categories. For all other families, the output is an N x E matrix. The number of summary statistics E is equal to 2 + length(probs): The Estimate column contains point

estimates (either mean or median depending on argument `robust`), while the `Est.Error` column contains uncertainty estimates (either standard deviation or median absolute deviation depending on argument `robust`). The remaining columns starting with `Q` contain quantile estimates as specified via argument `probs`.

In multivariate models, an additional dimension is added to the output which indexes along the different response variables.

See Also

[fitted.brmsfit](#)

Examples

```
## fit a model
if(requireNamespace("cmdstanr")){
  ## compute composition and ilr coordinates
  cilr <- complr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                 idvar = "ID", total = 1440)

  ## fit a model
  m1 <- brmcoda(complr = cilr,
                formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                  wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
                chain = 1, iter = 500,
                backend = "cmdstanr")

  ## compute expected predictions
  epred <- fitted(m1)
  head(epred)

  ## fit a model with compositional outcome
  m2 <- brmcoda(complr = cilr,
                formula = mvbind(ilr1, ilr2, ilr3, ilr4) ~ Stress + Female + (1 | ID),
                chain = 1, iter = 500,
                backend = "cmdstanr")

  ## expected predictions on compositional scale
  epredcomp <- fitted(m2, scale = "response")
  head(epredcomp)
}
```

---

fixef.brmcoda   *Population-Level Estimates*

Description

Extract the population-level ('fixed') effects from the `brmsfit` object in a `brmcoda` object.

Usage

```
## S3 method for class 'brmcoda'
fixef(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Further arguments passed to `fixef.brmsfit`.

Value

If summary is TRUE, a matrix returned by `posterior_summary` for the population-level effects. If summary is FALSE, a matrix with one row per posterior draw and one column per population-level effect.

See Also

`fixef.brmsfit`

Examples

```
## fit a model
if(requireNamespace("cmdstanr")){
  ## fit a model
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  ## extract population-Level coefficients
  fixef(m)
}
```

---

get-substitution   *Helper functions used only internally to estimate substitution model*

Description

Helper functions used only internally to estimate substitution model

is.brmcoda	*Checks if argument is a* brmcoda *object*

Description

Checks if argument is a brmcoda object

Usage

```
is.brmcoda(x)
```

Arguments

x   An object of class brmcoda.

is.complr	*Checks if argument is a* complr *object*

Description

Checks if argument is a complr object

Usage

```
is.complr(x)
```

Arguments

x   An object of class complr.

is.substitution	*Checks if argument is a* substitution *object*

Description

Checks if argument is a substitution object

Usage

```
is.substitution(x)
```

Arguments

x   An object of class substitution.

---

launch_shinystan.brmcoda

Interface to **shinystan**

Description

Provide an interface to **shinystan** for models fitted with **brms**

Usage

```
## S3 method for class 'brmcoda'
launch_shinystan(object, ...)
```

Arguments

object	A fitted model object of class brmcoda.
...	Optional arguments to pass to launch_shinystan or runApp.

Value

An S4 shinystan object

See Also

launch_shinystan

---

loo.brmcoda   *Efficient approximate leave-one-out cross-validation (LOO)*

Description

Perform approximate leave-one-out cross-validation based on the posterior likelihood using the **loo** package. For more details see [loo](loo).

Usage

```
## S3 method for class 'brmcoda'
loo(x, ...)
```

Arguments

x	A brmcoda object.
...	More brmsfit objects or further arguments passed to the underlying post-processing functions. In particular, see prepare_predictions for further supported arguments.

Value

If just one object is provided, an object of class loo. If multiple objects are provided, an object of class loolist.

See Also

[loo.brmsfit](#)

---

mcmc_plot.brmcoda   *MCMC Plots Implemented in* **bayesplot**

Description

Call MCMC plotting functions implemented in the **bayesplot** package.

Usage

```
## S3 method for class 'brmcoda'
mcmc_plot(object, ...)
```

Arguments

object	A brmcoda class object.
...	Further arguments passed to [mcmc_plot.brmsfit](#).

Value

A [ggplot](#) object that can be further customized using the **ggplot2** package.

See Also

[mcmc_plot.brmsfit](#)

Examples

```
## Not run:
cilr <- complr(data = mcompd, sbp = sbp,
        parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID")

# model with compositional predictor at between and within-person levels
fit <- brmcoda(complr = cilr,
               formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                  wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
               chain = 1, iter = 500)
mcmc_plot(fit)

## End(Not run)
```

---

mcompd   *Multilevel Compositional Data*

Description

A simulated dataset containing multiple days of compositional data.

Usage

```
mcompd
```

Format

A data table containing 10 variables.

ID A unique identifier for each individual

Time Recurrence time of repeated measures by individual

Stress Self report stress measures on a 0 to 10 scale — repeated measure

TST Total Sleep Time (minutes) — repeated measure

WAKE Wake time while in bed, trying to sleep (minutes) — repeated measure

MVPA Moderate to Vigorous Physical Activity (minutes) — repeated measure

LPA Light Physical Activity (minutes) — repeated measure

SB Sedentary Behavior (minutes) — repeated measure

Age Age in years — baseline measure only

Female Binary: whether participants identified as female (1) or not (0) — baseline measure only

---

mean.complr   *Mean amounts and mean compositions presented in a* complr *object.*

Description

Mean amounts and mean compositions presented in a complr object.

Usage

```
## S3 method for class 'complr'
mean(x, weight = c("equal", "proportional"), ...)
```

Arguments

x	An object of class `complr`.
weight	A character value specifying the weight to use in calculation of the reference composition. If `"equal"`, give equal weight to units (e.g., individuals). If `"proportional"`, weights in proportion to the frequencies of units being averaged (e.g., observations across individuals) Default is equal.
...	generic argument, not in use.

Examples

```
cilr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
               idvar = "ID")
mean(cilr)
```

---

model.frame.brmcoda	*Extracting the Model Frame from a Formula or Fit from* **brmcoda** *object*

Description

Extracting the Model Frame from a Formula or Fit from **brmcoda** object

Usage

```
## S3 method for class 'brmcoda'
model.frame(formula, ...)
```

Arguments

formula	A brmcoda object.
...	Further arguments to be passed to methods.

---

multilevelcoda_sim	*multilevelcoda Simulation Study Results*

Description

Provide the full results for a simulation study testing the performance of **multilevelcoda**

Usage

```
multilevelcoda_sim()
```

Value

An S4 shiny object

nobs.brmcoda *Extract Number of Observations from* **brmcoda** *object*

Description

Extract Number of Observations from **brmcoda** object

Usage

```
## S3 method for class 'brmcoda'
nobs(object, ...)
```

Arguments

object   A brmcoda object.

...   Further arguments to be passed to methods.

pairs.brmcoda *Create a matrix of output plots from a* brmcoda*'s* brmsfit *object*

Description

A pairs method that is customized for MCMC output.

Usage

```
## S3 method for class 'brmcoda'
pairs(x, ...)
```

Arguments

x   A brmcoda class object.

...   Further arguments passed to pairs.brmsfit.

See Also

pairs.brmsfit

Examples

```
## Not run:
cilr <- complr(data = mcompd, sbp = sbp,
        parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID")

# model with compositional predictor at between and within-person levels
fit <- brmcoda(complr = cilr,
                formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
                chain = 1, iter = 500)
pairs(fit)

## End(Not run)
```

---

pivot_coord	*Estimate pivot balance coordinates*

Description

Estimate pivot balance coordinates

Usage

```
pivot_coord(object, summary = TRUE, method = c("rotate", "refit"), ...)
```

Arguments

object	An object of class brmcoda.
summary	Should summary statistics be returned instead of the raw values? Default is TRUE.
method	A character string. Should the pivot balance coordinates be estimated by "rotate" the sequential binary partition using the same brmcoda object or "refit" the brmcoda object? Default is "rotate".
...	currently ignored.

Value

A list of [brmcoda](#) for each pivot balance coordinate.

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- complr(data = mcompd, sbp = sbp,
                parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID",
                total = 1440)

  # inspects ILRs before passing to brmcoda
  names(cilr$between_logratio)
```

```
names(cilr$within_logratio)
names(cilr$logratio)

# model with compositional predictor at between and within-person levels
m <- brmcoda(complr = cilr,
             formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                   wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
             chain = 1, iter = 500,
             backend = "cmdstanr")

m_pivot_coord <- pivot_coord(m)
summary(m_pivot_coord)
}
```

pivot_coord_refit   *Estimate pivot balance coordinates by refitting model.*

Description

Estimate pivot balance coordinates by refitting model.

Usage

```
pivot_coord_refit(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Further arguments passed to [brm]().

Value

A list of [brmcoda]() for each pivot balance coordinate.

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- complr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID",
                 total = 1440)

  m <- brmcoda(complr = cilr,
               formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                     wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
               chain = 1, iter = 500,
               backend = "cmdstanr")

  m_pivot_coord_refit <- pivot_coord_refit(m)
  summary(m_pivot_coord_refit)
}
```

---

pivot_coord_rotate	*Estimate pivot balance coordinates by rotating sequential binary partition.*

Description

Estimate pivot balance coordinates by rotating sequential binary partition.

Usage

```
pivot_coord_rotate(object, summary = TRUE, ...)
```

Arguments

object	An object of class brmcoda.
summary	Should summary statistics be returned instead of the raw values? Default is TRUE.
...	currently ignored.

Value

A list of [brmcoda](#) for each pivot balance coordinate.

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- complr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID",
                 total = 1440)

  m <- brmcoda(complr = cilr,
               formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                  wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
               chain = 1, iter = 500,
               backend = "cmdstanr")

  m_pivot_coord_rotate <- pivot_coord_rotate(m)
  summary(m_pivot_coord_rotate)

  m_pivot_coord_raw <-  pivot_coord_rotate(m, summary = FALSE)
  posterior::summarise_draws(posterior::as_draws_array(m_pivot_coord_raw$output))
  }
```

plot.brmcoda *Trace and Density Plots for MCMC Draws plot*

Description

Make a plot of brmcoda model results.

Usage

```
## S3 method for class 'brmcoda'
plot(x, ...)
```

Arguments

x   A brmcoda class object.

...   Further arguments passed to plot.brmsfit.

Value

An invisible list of gtable objects.

See Also

plot.brmsfit

Examples

```
## Not run:
cilr <- complr(data = mcompd, sbp = sbp,
        parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID")

# model with compositional predictor at between and within-person levels
fit <- brmcoda(complr = cilr,
               formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                  wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
               chain = 1, iter = 500)
plot(fit)

## End(Not run)
```

---

plot.substitution   *Substitution Plot*

Description

Make a plot of [substitution](#) model results.

Usage

```
## S3 method for class 'substitution'
plot(x, to, ref, level, ...)
```

Arguments

x	A [substitution](#) class object.
to	A character value or vector specifying the names of the compositional parts that were reallocated to in the model.
ref	A character value of (("grandmean" or "clustermean" or "users"),
level	A character value of ("between", "within"), or "aggregate").
...	Further components to the plot, followed by a plus sign (+).

Value

A ggplot graph object showing the estimated difference in outcome when each pair of compositional variables are substituted for a specific time.

---

pp_check.brmcoda   *Posterior Predictive Checks for* brmcoda *Objects*

Description

Perform posterior predictive checks with the help of the **bayesplot** package.

Usage

```
## S3 method for class 'brmcoda'
pp_check(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Further arguments passed to [predict.brmsfit](#) as well as to the PPC function specified in type.

See Also

[pp_check.brmsfit](#)

---

predict.brmcoda   *Draws from the Posterior Predictive Distribution*

Description

Compute posterior draws of the posterior predictive distribution of a brmsfit model in the brmcoda object. Can be performed for the data used to fit the model (posterior predictive checks) or for new data. By definition, these draws have higher variance than draws of the expected value of the posterior predictive distribution computed by `fitted.brmcoda`. This is because the residual error is incorporated in posterior_predict. However, the estimated means of both methods averaged across draws should be very similar.

Usage

```
## S3 method for class 'brmcoda'
predict(object, scale = c("linear", "response"), summary = TRUE, ...)
```

Arguments

object	An object of class brmcoda.
scale	Specifically for models with compositional responses, either "response" or "linear". If "linear", results are returned on the log-ratio scale. If "response", results are returned on the compositional scale of the response variable.
summary	Should summary statistics be returned instead of the raw values? Default is TRUE.
...	Further arguments passed to `predict.brmsfit` that control additional aspects of prediction.

Value

An array of predicted response values. If summary = FALSE the output resembles those of `posterior_predict.brmsfit`.

If summary = TRUE the output depends on the family: For categorical and ordinal families, the output is an N x C matrix, where N is the number of observations, C is the number of categories, and the values are predicted category probabilities. For all other families, the output is a N x E matrix where E = 2 + length(probs) is the number of summary statistics: The Estimate column contains point estimates (either mean or median depending on argument robust), while the Est.Error column contains uncertainty estimates (either standard deviation or median absolute deviation depending on argument robust). The remaining columns starting with Q contain quantile estimates as specified via argument probs.

See Also

`predict.brmsfit`

Examples

```
if(requireNamespace("cmdstanr")){
  ## fit a model
  cilr <- complr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                 idvar = "ID", total = 1440)

  m1 <- brmcoda(complr = cilr,
                formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                  wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
                chain = 1, iter = 500,
                backend = "cmdstanr")

  ## predicted responses
  pred <- predict(m1)
  head(pred)

  ## fit a model with compositional outcome
  m2 <- brmcoda(complr = cilr,
                formula = mvbind(ilr1, ilr2, ilr3, ilr4) ~ Stress + Female + (1 | ID),
                chain = 1, iter = 500,
                backend = "cmdstanr")

  ## predicted responses on compositional scale
  predcomp <- predict(m2, scale = "linear")
  head(predcomp)
}
```

---

print.brmcoda   *Print a Summary for a fitted* brmsfit *model in a* brmcoda *object*

Description

Print a Summary for a fitted `brmsfit` model in a `brmcoda` object

Usage

```
## S3 method for class 'brmcoda'
print(x, ...)
```

Arguments

x	An object of class brmcoda.
...	Other arguments passed to summary.brmcoda.

See Also

summary.brmcoda

Examples

```
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

 print(m)
}
```

---

print.complr	*Print a Summary for a* complr *object*

Description

Print a Summary for a complr object

Usage

```
## S3 method for class 'complr'
print(x, ...)
```

Arguments

x   An object of class complr.

...   Other arguments passed to summary.complr.

See Also

summary.complr

Examples

```
cilr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
               idvar = "ID")
print(cilr)
```

---

print.substitution   *Print a Summary for a* substitution *object*

Description

Print a Summary for a substitution object

Usage

```
## S3 method for class 'substitution'
print(x, ...)
```

Arguments

x	A substitution object.
...	Additional arguments to be passed to to method summary of substitution.

See Also

summary.substitution

Examples

```
if(requireNamespace("cmdstanr")){
  ## fit a model with compositional predictor at between and between-person levels
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  subm <- substitution(object = m, delta = 5)
  print(subm)
}
```

---

print.summary.complr   *Summary for a* complr *object*

Description

Summary for a complr object

Usage

```
## S3 method for class 'summary.complr'
print(x, ...)
```

Arguments

x	An object of class summary.complr.
...	Other arguments passed to summary.complr.

See Also

summary.complr

Examples

```
cilr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
               idvar = "ID")
print(cilr)
```

---

prior_summary.brmcoda   *Extract Priors of a* brmsfit *from a* brmcoda *object*

Description

Compute Bayes factors from marginal likelihoods

Usage

```
## S3 method for class 'brmcoda'
prior_summary(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Further arguments passed to or from other methods.

See Also

prior_summary.brmsfit

---

psub   *Possible Pairwise Substitutions*

Description

A dataset containing possible pairwise subsitutions.

Usage

```
psub
```

Format

A data table containing 5 variables.

TST first compositional variable

WAKE second compositional variable

MVPA third compositional variable

LPA fourth compositional variable

SB fifth compositional variable

---

ranef.brmcoda   *Group-Level Estimates*

Description

Extract the group-level ('random') effects of each level of the brmsfit object in a brmcoda object.

Usage

```
## S3 method for class 'brmcoda'
ranef(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Further arguments passed to `ranef.brmsfit`.

Value

A list of 3D arrays (one per grouping factor). If summary is TRUE, the 1st dimension contains the factor levels, the 2nd dimension contains the summary statistics (see `posterior_summary`), and the 3rd dimension contains the group-level effects. If summary is FALSE, the 1st dimension contains the posterior draws, the 2nd dimension contains the factor levels, and the 3rd dimension contains the group-level effects.

See Also

ranef.brmsfit

Examples

```
## fit a model
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  ## extract group-level coefficients
  ranef(m)
}
```

---

residuals.brmcoda   *Posterior Draws of Residuals/Predictive Errors*

Description

Compute posterior draws of residuals/predictive errors

Usage

```
## S3 method for class 'brmcoda'
residuals(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Further arguments passed to residuals.brmsfit.

Value

An array of predictive error/residual draws. If summary = FALSE the output resembles those of predictive_error.brmsfit. If summary = TRUE the output is an N x E matrix, where N is the number of observations and E denotes the summary statistics computed from the draws.

See Also

residuals.brmsfit

Examples

```
## fit a model
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                              parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                              idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  ## extract residuals
  res <- residuals(m)
  head(res)
}
```

---

sbp   *Sequential Binary Partition*

Description

A matrix of sequential binary partition.

Usage

```
sbp
```

Format

A matrix with 5 columns and 4 rows.

TST first compositional variable

WAKE second compositional variable

MVPA third compositional variable

LPA fourth compositional variable

SB fifth compositional variable

---

sim   *multilevelcoda Simulation Study results*

Description

A list of 4 components

Usage

```
sim
```

Format

A list with 5 columns and 4 rows.

brmcoda_tab Simulation results for brmcoda() for tables

sub_tab Simulation results for substitution() for tables

brmcoda_plot Simulation results for brmcoda() for graphs

sub_plot Simulation results for substitution() for graphs

---

sub   *Simple Substitution*

Description

This function is an alias of [substitution](#) to estimates the the difference in an outcome when compositional parts are substituted for specific unit(s) using a aggregate reference composition (e.g., compositional mean at sample level, not seperated by between- and within effects). It is recommended that users run substitution model using the [substitution](#) function.

Usage

```
sub(
  object,
  delta,
  basesub,
  summary = TRUE,
  ref = "grandmean",
  level = "aggregate",
  weight = "equal",
  scale = c("response", "linear"),
  cores = NULL,
  ...
)
```

Arguments

object	A fitted [brmcoda](#) object.
delta	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
basesub	A base substitution. Can be a data.frame or data.table of the base possible substitution of compositional parts, which can be computed using function [build.basesub](#). If "one-to-all", all possible one-to-remaining reallocations are estimated. If NULL, all possible one-to-one reallocations are estimated.
summary	A logical value. Should the estimate at each level of the reference grid (FALSE) or their average (TRUE) be returned? Default is TRUE. Only applicable for model with covariates in addition to the isometric log-ratio coordinates (i.e., adjusted model).
ref	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a data.frame or data.table of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to "grandmean".
level	A character string or vector. Should the estimate of multilevel models focus on the "between" and/or "within" or "aggregate" variance? Single-level models are default to "aggregate".
weight	A character value specifying the weight to use in calculation of the reference composition. If "equal", give equal weight to units (e.g., individuals). If "proportional", weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default to "equal" for ref = "grandmean" and "proportional" for ref = "clustermean".
scale	Either "response" or "linear". If "response", results are returned on the scale of the response variable. If "linear", results are returned on the scale of the linear predictor term, that is without applying the inverse link function or other transformations.
cores	Number of cores to use when executing the chains in parallel, we recommend setting the mc.cores option to be as many processors as the hardware and RAM allow (up to the number of compositional parts). For non-Windows OS in non-interactive R sessions, forking is used instead of PSOCK clusters.
...	currently ignored.

Value

A list containing the results of multilevel compositional substitution model. The first six lists contain the results of the substitution estimation for a compositional part.

Mean	Posterior means.
CI_low and CI_high	
	95% credible intervals.
Delta	Amount substituted across compositional parts.
From	Compositional part that is substituted from.

To	Compositional parts that is substituted to.
Level	Level where changes in composition takes place.
Reference	Either grandmean, clustermean, or users.

See Also

[substitution](#)

Examples

```
if(requireNamespace("cmdstanr")){

cilr <- complr(data = mcompd, sbp = sbp,
                parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID", total = 1440)

# model with compositional predictor at between and within-person levels
m <- brmcoda(complr = cilr,
             formula = Stress ~ ilr1 + ilr2 + ilr3 + ilr4 + (1 | ID),
             chain = 1, iter = 500,
             backend = "cmdstanr")

subm <- sub(object = m, basesub = psub, delta = 5)
}
```

---

submargins   *Average Substitution*

Description

This function is an alias of [substitution](#) to estimates the the difference in an outcome when
compositional parts are substituted for specific unit(s) using cluster mean (e.g., compositional mean
at individual level) as reference composition. It is recommended that users run substitution model
using the [substitution](#) function.

Usage

```
submargins(
  object,
  delta,
  basesub,
  ref = "clustermean",
  level = "aggregate",
  weight = "proportional",
  scale = c("response", "linear"),
  cores = NULL,
  ...
)
```

Arguments

object	A fitted brmcoda object.
delta	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
basesub	A base substitution. Can be a data.frame or data.table of the base possible substitution of compositional parts, which can be computed using function build.basesub. If "one-to-all", all possible one-to-remaining reallocations are estimated. If NULL, all possible one-to-one reallocations are estimated.
ref	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a data.frame or data.table of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to "grandmean".
level	A character string or vector. Should the estimate of multilevel models focus on the "between" and/or "within" or "aggregate" variance? Single-level models are default to "aggregate".
weight	A character value specifying the weight to use in calculation of the reference composition. If "equal", give equal weight to units (e.g., individuals). If "proportional", weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default to "equal" for ref = "grandmean" and "proportional" for ref = "clustermean".
scale	Either "response" or "linear". If "response", results are returned on the scale of the response variable. If "linear", results are returned on the scale of the linear predictor term, that is without applying the inverse link function or other transformations.
cores	Number of cores to use when executing the chains in parallel, we recommend setting the mc.cores option to be as many processors as the hardware and RAM allow (up to the number of compositional parts). For non-Windows OS in non-interactive R sessions, forking is used instead of PSOCK clusters.
...	currently ignored.

Value

A list containing the results of multilevel compositional substitution model. The first six lists contain the results of the substitution estimation for a compositional part.

Mean	Posterior means.
CI_low and CI_high	
	95% credible intervals.
Delta	Amount substituted across compositional parts.
From	Compositional part that is substituted from.
To	Compositional parts that is substituted to.
Level	Level where changes in composition takes place.
Reference	Either grandmean, clustermean, or users.

See Also

[substitution](#)

Examples

```
if(requireNamespace("cmdstanr")){

cilr <- complr(data = mcompd, sbp = sbp,
                parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID", total = 1440)

# model with compositional predictor at between and within-person levels
m <- brmcoda(complr = cilr,
             formula = Stress ~ ilr1 + ilr2 + ilr3 + ilr4 + (1 | ID),
             chain = 1, iter = 500,
             backend = "cmdstanr")

subm <- submargins(object = m, basesub = psub, delta = 5)
}
```

---

substitution   *Multilevel Compositional Substitution Analysis*

Description

Estimate the difference in an outcome when compositional parts are substituted for specific unit(s). The substitution output encapsulates the substitution results for all compositional parts present in the [brmcoda](#) object.

Usage

```
substitution(
  object,
  delta,
  basesub,
  summary = TRUE,
  ref = c("grandmean", "clustermean"),
  level = c("between", "within", "aggregate"),
  weight = c("equal", "proportional"),
  scale = c("response", "linear"),
  cores = NULL,
  ...
)
```

Arguments

object	A fitted [brmcoda](#) object.
delta	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.

basesub	A base substitution. Can be a data.frame or data.table of the base possible substitution of compositional parts, which can be computed using function [build.basesub](). If "one-to-all", all possible one-to-remaining reallocations are estimated. If NULL, all possible one-to-one reallocations are estimated.
summary	A logical value. Should the estimate at each level of the reference grid (FALSE) or their average (TRUE) be returned? Default is TRUE. Only applicable for model with covariates in addition to the isometric log-ratio coordinates (i.e., adjusted model).
ref	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a data.frame or data.table of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to "grandmean".
level	A character string or vector. Should the estimate of multilevel models focus on the "between" and/or "within" or "aggregate" variance? Single-level models are default to "aggregate".
weight	A character value specifying the weight to use in calculation of the reference composition. If "equal", give equal weight to units (e.g., individuals). If "proportional", weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default to "equal" for ref = "grandmean" and "proportional" for ref = "clustermean".
scale	Either "response" or "linear". If "response", results are returned on the scale of the response variable. If "linear", results are returned on the scale of the linear predictor term, that is without applying the inverse link function or other transformations.
cores	Number of cores to use when executing the chains in parallel, we recommend setting the mc.cores option to be as many processors as the hardware and RAM allow (up to the number of compositional parts). For non-Windows OS in non-interactive R sessions, forking is used instead of PSOCK clusters.
...	currently ignored.

Value

A list containing the results of multilevel compositional substitution model. The first six lists contain the results of the substitution estimation for a compositional part.

Mean	Posterior means.
CI_low and CI_high	
	95% credible intervals.
Delta	Amount substituted across compositional parts.
From	Compositional part that is substituted from.
To	Compositional parts that is substituted to.
Level	Level where changes in composition takes place.
Reference	Either grandmean, clustermean, or users.

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- complr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                 idvar = "ID", total = 1440)

  # model with compositional predictor at between and within-person levels
  fit1 <- brmcoda(complr = cilr,
                  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                     wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
                  chain = 1, iter = 500, backend = "cmdstanr")

  # one to one reallocation at between and within-person levels
  sub1 <- substitution(object = fit1, delta = 5, level = c("between", "within"))
  summary(sub1)

  # one to all reallocation at between and within-person levels
  sub2 <- substitution(object = fit1, delta = 5, level = c("between", "within"),
                       basesub = "one-to-all")
  summary(sub2)

  # model with compositional predictor at aggregate level of variance
  fit2 <- brmcoda(complr = cilr,
                  formula = Stress ~ ilr1 + ilr2 + ilr3 + ilr4 + (1 | ID),
                  chain = 1, iter = 500, backend = "cmdstanr")
  sub3 <- substitution(object = fit2, delta = 5, level = c("aggregate"))

}
```

---

summary.brmcoda   *Create a Summary of a fitted* brmsfit *model in a* brmcoda *object*

Description

Create a Summary of a fitted brmsfit model in a brmcoda object

Usage

```
## S3 method for class 'brmcoda'
summary(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Other arguments passed to summary.brmsfit.

Examples

```
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  summary(m)
}
```

---

summary.complr	*Create a Summary of a* complr *object*

Description

Create a Summary of a complr object

Usage

```
## S3 method for class 'complr'
summary(object, weight = c("equal", "proportional"), ...)
```

Arguments

object	An object of class complr.
weight	A character value specifying the weight to use in calculation of the reference composition. If "equal", give equal weight to units (e.g., individuals). If "proportional", weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default is equal.
...	generic argument, not in use.

Examples

```
cilr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
               idvar = "ID")
summary(cilr)
cilr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"))
summary(cilr)
```

---

summary.pivot_coord   *Create a Summary of a fitted* brmsfit *model from a* pivot_coord
                           object

Description

Create a Summary of a fitted `brmsfit` model from a `pivot_coord` object

Usage

```
## S3 method for class 'pivot_coord'
summary(object, digits = 2, ...)
```

Arguments

object	An object of class `pivot_coord`.
digits	A integer value used for number formatting. Default is 2.
...	currently ignored.

Value

A data table of results.

Examples

```
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  m_pb <- pivot_coord(m)
  summary(m_pb)
}
```

---

summary.substitution   *Create a Summary of a Substitution Model represented by a* substitution *object*

Description

Create a Summary of a Substitution Model represented by a substitution object

Usage

```
## S3 method for class 'substitution'
summary(object, delta, to, from, ref, level, digits = 2, ...)
```

Arguments

object	A substitution class object.
delta	A integer, numeric value or vector indicating the desired delta at which substitution results should be summarised. Default to all delta available in the substitution object.
to	A character value or vector specifying the names of the compositional parts that were reallocated to in the model.
from	A character value or vector specifying the names of the compositional parts that were reallocated from in the model.
ref	Either a character value or vector (("grandmean" and/or "clustermean" or "users"), Default to all ref available in the substitution object.
level	A character string or vector ("between" and/or "within"). Default to all level available in the substitution object.
digits	A integer value used for number formatting. Default is 2.
...	generic argument, not in use.

Value

A summary of substitution object.

Mean	Posterior means.
CI_low and CI_high	
	95% credible intervals.
Delta	Amount substituted across compositional parts.
From	Compositional part that is substituted from.
To	Compositional parts that is substituted to.
Level	Level where changes in composition takes place. Either between or within.
Reference	Either grandmean, clustermean, or users.

Examples

```
if(requireNamespace("cmdstanr")){
  ## fit a model with compositional predictor at between and between-person levels
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  subm <- substitution(object = m, delta = 5)
  summary(subm)
}
```

---

update.brmcoda   *Update* brmcoda *models*

Description

This method allows for updating an existing brmcoda object.

Usage

```
## S3 method for class 'brmcoda'
update(object, formula. = NULL, newdata = NULL, newcomplr = NULL, ...)
```

Arguments

object	A fitted brmcoda object to be updated.
formula.	Changes to the formula; for details see update.formula and brmsformula.
newdata	A data.frame or data.table containing data of all variables used in the analysis. It must include a composition and the same ID variable as the existing complr object.
newcomplr	A complr object containing data of composition, ILR coordinates, and other variables used in the updated model.
...	Further arguments passed to brm.

Value

A brmcoda with two elements

complr	An object of class complr used in the brm model.
model	An object of class brmsfit, which contains the posterior draws along with many other useful information about the model.

See Also

[brmcoda](#)

Examples

```
if(requireNamespace("cmdstanr")){

# model with compositional predictor at between and within-person levels
fit <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID"),
               formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                   wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
               chain = 1, iter = 500,
               backend = "cmdstanr")

# removing the effect of wilr1
fit1 <- update(fit, formula. = ~ . - wilr1)

# using only a subset
fit2 <- update(fit, newdata = mcompd[ID != 1])
}
```

---

update.complr   *Update* [complr](#)

Description

This method allows for updating an existing [complr](#) object.

Usage

```
## S3 method for class 'complr'
update(object, newdata, ...)
```

Arguments

object	A [complr](#) class object to be updated.
newdata	A data.frame or data.table containing data of all variables used in the analysis. It must include a composition and the same ID variable as the existing [complr](#) object.
...	generic argument, not in use.

Value

A complr object with at least the following elements.

comp	A vector of class acomp representing one closed composition or a matrix of class acomp representing multiple closed compositions each in one row.
between_comp	A vector of class acomp representing one closed between-person composition or a matrix of class acomp representing multiple closed between-person compositions each in one row.
within_comp	A vector of class acomp representing one closed within-person composition or a matrix of class acomp representing multiple closed within-person compositions each in one row.
logratio	Log ratio transform of composition.
between_logratio	
	Log ratio transform of between-person composition.
within_logratio	
	Log ratio transform of within-person composition.
data	The user's dataset or imputed dataset if the input data contains zeros.
transform	Type of transform applied on compositional data.
parts	Names of compositional variables.
idvar	Name of the variable containing IDs.
total	Total amount to which the compositions is closed.

See Also

complr

Examples

```
cilr <- complr(data = mcompd, sbp = sbp,
                parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID")

# update with new data
newdat <- mcompd[ID != 1] # excluding ID 1
cilr1 <- update(object = cilr, newdata = newdat)
```

---

var.complr	*Variance of compositions presented in a* complr *object.*

Description

Variance of compositions presented in a complr object.

Usage

```
## S3 method for class 'complr'
var(x, weight = c("equal", "proportional"), ...)
```

Arguments

x	An object of class `complr`.
weight	A character value specifying the weight to use in calculation of the reference composition. If `"equal"`, give equal weight to units (e.g., individuals). If `"proportional"`, weights in proportion to the frequencies of units being averaged (e.g., observations across individuals) Default is `equal`.
...	generic argument, not in use.

VarCorr.brmcoda   *Extract Variance and Correlation Components*

Description

Calculates the estimated standard deviations, correlations and covariances of the group-level terms of the `brmsfit` object in a `brmcoda` object.

Usage

```
## S3 method for class 'brmcoda'
VarCorr(x, ...)
```

Arguments

x	An object of class brmcoda.
...	Further arguments passed to `VarCorr.brmsfit`.

Value

A list of lists (one per grouping factor), each with three elements: a matrix containing the standard deviations, an array containing the correlation matrix, and an array containing the covariance matrix with variances on the diagonal.

See Also

`VarCorr.brmsfit`

Examples

```
## fit a model
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")
```

```
  VarCorr(m)
}
```

---

vcov.brmcoda   *Covariance and Correlation Matrix of Population-Level Effects*

Description

Get a point estimate of the covariance or correlation matrix of population-level parameters of the
brmsfit object in a brmcoda object.

Usage

```
## S3 method for class 'brmcoda'
vcov(object, ...)
```

Arguments

object	An object of class brmcoda.
...	Further arguments passed to vcov.brmsfit.

Value

covariance or correlation matrix of population-level parameters

See Also

vcov.brmsfit

Examples

```
## fit a model
if(requireNamespace("cmdstanr")){
  m <- brmcoda(complr = complr(data = mcompd, sbp = sbp,
                               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                               idvar = "ID", total = 1440),
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500,
  backend = "cmdstanr")

  vcov(m)
}
```

Description

This function is an alias of [substitution](#) to estimates the the difference in an outcome when compositional parts are substituted for specific unit(s) at *within* level using a single reference composition (e.g., compositional mean at sample level). It is recommended that users run substitution model using the [substitution](#) function.

Usage

```
wsub(
  object,
  basesub,
  delta,
  summary = TRUE,
  ref = "grandmean",
  level = "within",
  weight = "equal",
  scale = c("response", "linear"),
  cores = NULL,
  ...
)
```

Arguments

object	A fitted [brmcoda](#) object.
basesub	A base substitution. Can be a data.frame or data.table of the base possible substitution of compositional parts, which can be computed using function [build.basesub](#). If "one-to-all", all possible one-to-remaining reallocations are estimated. If NULL, all possible one-to-one reallocations are estimated.
delta	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
summary	A logical value. Should the estimate at each level of the reference grid (FALSE) or their average (TRUE) be returned? Default is TRUE. Only applicable for model with covariates in addition to the isometric log-ratio coordinates (i.e., adjusted model).
ref	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a data.frame or data.table of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to "grandmean".
level	A character string or vector. Should the estimate of multilevel models focus on the "between" and/or "within" or "aggregate" variance? Single-level models are default to "aggregate".

weight   A character value specifying the weight to use in calculation of the reference
                composition. If "equal", give equal weight to units (e.g., individuals). If
                "proportional", weights in proportion to the frequencies of units being av-
                eraged (e.g., observations across individuals). Default to "equal" for ref =
                "grandmean" and "proportional" for ref = "clustermean".

scale   Either "response" or "linear". If "response", results are returned on the
                scale of the response variable. If "linear", results are returned on the scale of
                the linear predictor term, that is without applying the inverse link function or
                other transformations.

cores   Number of cores to use when executing the chains in parallel, we recommend
                setting the mc.cores option to be as many processors as the hardware and RAM
                allow (up to the number of compositional parts). For non-Windows OS in non-
                interactive R sessions, forking is used instead of PSOCK clusters.

...   currently ignored.

Value

A list containing the results of multilevel compositional substitution model. The first six lists con-
tain the results of the substitution estimation for a compositional part.

Mean	Posterior means.
CI_low and CI_high	
	95% credible intervals.
Delta	Amount substituted across compositional parts.
From	Compositional part that is substituted from.
To	Compositional parts that is substituted to.
Level	Level where changes in composition takes place.
Reference	Either grandmean, clustermean, or users.

See Also

[substitution](#)

Examples

```
if(requireNamespace("cmdstanr")){

cilr <- complr(data = mcompd, sbp = sbp,
               parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID", total = 1440)

# model with compositional predictor at between and within-person levels
m <- brmcoda(complr = cilr,
             formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                               wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
             chain = 1, iter = 500,
             backend = "cmdstanr")

subm <- wsub(object = m, basesub = psub, delta = 60)
}
```

---

wsubmargins   *Within-person Average Substitution*

Description

This function is an alias of [substitution](#) to estimates the the difference in an outcome when compositional parts are substituted for specific unit(s) at *within* level using cluster mean (e.g., compositional mean at individual level) as reference composition. It is recommended that users run substitution model using the [substitution](#) function.

Usage

```
wsubmargins(
  object,
  delta,
  basesub,
  ref = "clustermean",
  level = "within",
  weight = "proportional",
  scale = c("response", "linear"),
  cores = NULL,
  ...
)
```

Arguments

object	A fitted [brmcoda](#) object.
delta	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
basesub	A base substitution. Can be a data.frame or data.table of the base possible substitution of compositional parts, which can be computed using function [build.basesub](#). If "one-to-all", all possible one-to-remaining reallocations are estimated. If NULL, all possible one-to-one reallocations are estimated.
ref	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a data.frame or data.table of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid is only possible for simple substitution. Single level models are default to "grandmean".
level	A character string or vector. Should the estimate of multilevel models focus on the "between" and/or "within" or "aggregate" variance? Single-level models are default to "aggregate".
weight	A character value specifying the weight to use in calculation of the reference composition. If "equal", give equal weight to units (e.g., individuals). If "proportional", weights in proportion to the frequencies of units being averaged (e.g., observations across individuals). Default to "equal" for ref = "grandmean" and "proportional" for ref = "clustermean".

scale   Either "response" or "linear". If "response", results are returned on the
                    scale of the response variable. If "linear", results are returned on the scale of
                    the linear predictor term, that is without applying the inverse link function or
                    other transformations.

cores   Number of cores to use when executing the chains in parallel, we recommend
                    setting the mc.cores option to be as many processors as the hardware and RAM
                    allow (up to the number of compositional parts). For non-Windows OS in non-
                    interactive R sessions, forking is used instead of PSOCK clusters.

...   currently ignored.

Value

A list containing the results of multilevel compositional substitution model. The first six lists con-
tain the results of the substitution estimation for a compositional part.

Mean	Posterior means.
CI_low and CI_high	
	95% credible intervals.
Delta	Amount substituted across compositional parts.
From	Compositional part that is substituted from.
To	Compositional parts that is substituted to.
Level	Level where changes in composition takes place.
Reference	Either grandmean, clustermean, or users.

See Also

substitution

Examples

```
if(requireNamespace("cmdstanr")){

cilr <- complr(data = mcompd, sbp = sbp,
                parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID", total = 1440)

# model with compositional predictor at between and within-person levels
m <- brmcoda(complr = cilr,
             formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
                                wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
             chain = 1, iter = 500,
             backend = "cmdstanr")

subm <- wsubmargins(object = m, basesub = psub, delta = 5)
}
```

Index

59