

Package ‘multilevelcoda’

August 13, 2023

Type Package

Title Estimate Bayesian Multilevel Models for Compositional Data

Version 1.2.1

Date 2023-8-13

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.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 4.0.0)

Imports stats, data.table (>= 1.12.0), compositions, brms, bayestestR, extraoperators, ggplot2, emmeans, insight, foreach, doFuture, abind, graphics

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

NeedsCompilation no

Author Flora Le [aut, cre] (<<https://orcid.org/0000-0003-0089-8167>>),
Joshua F. Wiley [aut] (<<https://orcid.org/0000-0002-0271-6702>>)

Maintainer Flora Le <13florale@gmail.com>

Repository CRAN

Date/Publication 2023-08-13 12:40:07 UTC

R topics documented:

as.data.frame.compilr	3
basesub	4
bayes_factor.brmcoda	4
brmcoda	5
bsub	6
bsubmargins	8
build.rg	9
coef.brmcoda	10
compilr	11
diagnostic-quantities-brmcoda	12
diagnostic_posterior.brmcoda	13
draws-index-brmcoda	14
fitted.brmcoda	15
fixef.brmcoda	16
get-substitution	17
is.brmcoda	17
is.compilr	18
is.substitution	18
launch_shinystan.brmcoda	18
mcmc_plot.brmcoda	19
mcompd	20
mean.compilr	20
model.frame.brmcoda	21
nobs.brmcoda	22
pairs.brmcoda	22
plot.brmcoda	23
plot.substitution	24
predict.brmcoda	24
print.brmcoda	26
print.compilr	27
print.substitution	27
prior_summary.brmcoda	28
psub	29
ranef.brmcoda	29
residuals.brmcoda	30
sbp	31
submargins	32
substitution	33
summary.brmcoda	34
summary.compilr	35
summary.substitution	36

update.brmcoda	37
update.compilr	38
VarCorr.brmcoda	39
vcov.brmcoda	40
wsub	41
wsubmargins	43

Index	45
--------------	-----------

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

Description

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

Usage

```
## S3 method for class 'compilr'
as.data.frame(
  x,
  row.names = NULL,
  optional = TRUE,
  class = c("composition", "logratio"),
  level = c("between", "within", "total"),
  ...
)

## S3 method for class 'compilr'
as.matrix(
  x,
  class = c("composition", "logratio"),
  level = c("between", "within", "total"),
  ...
)
```

Arguments

x	An object of class compilr.
row.names, optional	Unused and only added for consistency with the <code>as.data.frame</code> generic.
class	Optional. Can be "composition" and/or "logratio" to specify the geometry of the composition.
level	Optional. Can be "between", "within", and/or total indicating the level of the geometry.
...	generic argument, not in use.

basesub *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

```
basesub(parts)
```

Arguments

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

Value

A data table of all possible pairwise substitution.

Examples

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

ps2 <- basesub(c("WAKE", "MVPA", "LPA", "SB"))
print(ps2)
```

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	<i>Fit Bayesian generalised (non-)linear multilevel compositional model via full Bayesian inference</i>
---------	---

Description

Fit a brm model with multilevel ILR coordinates

Usage

```
brmcoda(compilr, formula, ...)
```

Arguments

compilr	A compilr 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

- [CompILR](#) An object of class [compilr](#) 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 <- compilr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"), idvar = "ID")

  # inspects ILRs before passing to brmcoda
  names(cilr$BetweenILR)
  names(cilr$WithinILR)
  names(cilr$TotalILR)

  # model with compositional predictor at between and within-person levels
  m1 <- brmcoda(compilr = 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(compilr = 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 = 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 <code>data.frame</code> or <code>data.table</code> of the base possible substitution of compositional parts. This data set can be computed using function basesub . If <code>NULL</code> , all possible pairwise substitution of compositional parts are used.
summary	A logical value. Should the estimate at each level of the reference grid (<code>FALSE</code>) or their average (<code>TRUE</code>) be returned? Default is <code>TRUE</code> . 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 <code>data.frame</code> or <code>data.table</code> 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.

level	A character string or vector. Should the estimate be at the "between" and/or "within" level?
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.
...	Additional arguments passed to describe_posterior .

Value

A list containing the results of multilevel compositional substitution model. The first four 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. Either between or within.
- Reference Either grandmean, clustermean, or users.

See Also

[substitution](#)

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- compilr(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(compilr = 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 = 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 <code>data.frame</code> or <code>data.table</code> of the base possible substitution of compositional parts. This data set can be computed using function basesub . If <code>NULL</code> , all possible pairwise substitution of compositional parts are used.
ref	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a <code>data.frame</code> or <code>data.table</code> 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.
level	A character string or vector. Should the estimate be at the "between" and/or "within" level?
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.
...	Additional arguments passed to describe_posterior .

Value

A list containing the results of multilevel compositional substitution model. The first four 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. Either between or within.
- Reference Either grandmean, clustermean, or users.

See Also

[substitution](#)

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- compilr(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(compilr = 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.rg

Reference Grid for substitution model.

Description

Build a dataset for fitted.brmcoda used in substitution model

Usage

```
build.rg(object, ref, 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 <code>data.frame</code> or <code>data.table</code> of user's specified reference grid consisting of combinations of covariates over which predictions are made. User's specified reference grid only applicable to substitution model using a single reference composition value.
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.
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. Default is FALSE.

Value

A reference grid consisting of a combination of covariates in `brmcoda`.

coef.brmcoda	<i>Model Coefficients</i>
--------------	---------------------------

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 <code>brmcoda</code> .
...	Further arguments passed to <code>coef.brmsfit</code> .

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(compilr = compilr(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)*

Description

Compute sets of compositions and IRLs for multilevel compositional data

Usage

```
compilr(data, sbp, parts, total = 1, idvar = "ID")
```

Arguments

<code>data</code>	A <code>data.frame</code> or <code>data.table</code> containing data of all variables used in the analysis. It must include a composition and a ID variable. Required.
<code>sbp</code>	A signary matrix indicating sequential binary partition. Required.
<code>parts</code>	A character vector specifying the names of compositional variables to be used.
<code>total</code>	A numeric value of the total amount to which the compositions should be closed. Default is 1.
<code>idvar</code>	A character string specifying the name of the variable containing IDs. Default is "ID".

Value

A `compilr` object with twelve elements.

- `BetweenComp` A vector of class `acom` representing one closed between-person composition or a matrix of class `acom` representing multiple closed between-person compositions each in one row.
- `WithinComp` A vector of class `acom` representing one closed within-person composition or a matrix of class `acom` representing multiple closed within-person compositions each in one row.
- `TotalComp` A vector of class `acom` representing one closed total composition or a matrix of class `acom` representing multiple closed total compositions each in one row.
- `BetweenILR` Isometric log ratio transform of between-person composition.
- `WithinILR` Isometric log ratio transform of within-person composition.
- `TotalILR` Isometric log ratio transform of total composition.
- `data` The user's dataset or imputed dataset if the input data contains zeros.
- `psi` A ILR matrix associated with user-defined partition structure.
- `sbp` The user-defined sequential binary partition matrix.
- `parts` Names of compositional variables.
- `idvar` Name of the variable containing IDs.
- `total` Total amount to which the compositions is closed.

Examples

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

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](#)

diagnostic_posterior.brmcoda

Posteriors Sampling Diagnostic

Description

Extract diagnostic metrics (Effective Sample Size (ESS), Rhat and Monte Carlo Standard Error MCSE).

Usage

```
## S3 method for class 'brmcoda'
diagnostic_posterior(posterior, diagnostic = c("ESS", "Rhat"), ...)
```

Arguments

posterior An object of class brmcoda.
 diagnostic Diagnostic metrics to compute. Character (vector) or list with one or more of these options: "ESS", "Rhat", "MCSE" or "all".
 ... Other arguments passed to [diagnostic_posterior](#).

See Also

[diagnostic_posterior](#)

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	<i>Expected Values of the Posterior Predictive Distribution</i>
----------------	---

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, acomp = FALSE, summary = TRUE, ...)
```

Arguments

<code>object</code>	An object of class <code>brmcoda</code> .
<code>acom</code>	Should the results be returned on the compositional scale of the response variable? Only applicable for models with compositional response
<code>summary</code>	Should summary statistics be returned instead of the raw values? Default is <code>TRUE</code> .
<code>...</code>	Further arguments passed to <code>fitted.brmsfit</code> 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 $N \times E \times C$ 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 $N \times E$ matrix. The number of summary statistics E is equal to $2 + \text{length}(\text{probs})$: 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 <- compilr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                 idvar = "ID", total = 1440)

  ## fit a model
  m1 <- brmcoda(compilr = 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(compilr = 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, acomp = TRUE)
  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(compilr = compilr(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)
}
```

<code>get-substitution</code>	<i>Helper functions used only internally to estimate substitution model</i>
-------------------------------	---

Description

Helper functions used only internally to estimate substitution model

<code>is.brmcoda</code>	<i>Checks if argument is a brmcoda object</i>
-------------------------	---

Description

Checks if argument is a brmcoda object

Usage

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

Arguments

`x` An object of class `brmcoda`.

`is.compilr` *Checks if argument is a `compilr` object*

Description

Checks if argument is a `compilr` object

Usage

`is.compilr(x)`

Arguments

`x` An object of class `compilr`.

`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.brmsfit](#) or `runApp`.

Value

An S4 shinystan object

See Also

[launch_shinystan](#)

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 <- compilr(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(compilr = 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.compilr

Mean amounts and mean compositions presented in a compilr object.

Description

Mean amounts and mean compositions presented in a compilr object.

Usage

```
## S3 method for class 'compilr'
mean(
  x,
  ...,
  class = c("composition", "logratio"),
  level = c("between", "within", "total"),
  weight = c("equal", "proportional"),
  digits = 3
)
```

Arguments

x	An object of class <code>compilr</code> .
...	generic argument, not in use.
class	Optional. Can be "composition" and/or "logratio" to specify the geometry of the composition.
level	Optional. Can be "between", "within", and/or total indicating the level of the geometry.
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.
digits	A integer value used for number formatting. Default is 3.

Examples

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

model.frame.brmcoda	<i>Extracting the Model Frame from a Formula or Fit from brmcoda object</i>
---------------------	--

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.

nobs.brmcoda	<i>Extract Number of Observations from brmcoda object</i>
--------------	--

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	<i>Create a matrix of output plots from a brmcoda's brmsfit object</i>
---------------	---

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 <- compilr(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(compilr = cilr,
  formula = Stress ~ bilr1 + bilr2 + bilr3 + bilr4 +
    wilr1 + wilr2 + wilr3 + wilr4 + (1 | ID),
  chain = 1, iter = 500)
pairs(fit)

## End(Not run)
```

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 <- compilr(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(compilr = 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" or "within").
...	Further arguments passed to ggplot .

Value

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

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, acomp = FALSE, summary = TRUE, ...)
```

Arguments

object	An object of class brmcoda.
acomp	Should the results be returned on the compositional scale of the response variable? Only applicable for models with compositional response
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 $N \times C$ 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 $N \times E$ matrix where $E = 2 + \text{length}(\text{probs})$ 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 <- compilr(data = mcompd, sbp = sbp,
                 parts = c("TST", "WAKE", "MVPA", "LPA", "SB"),
                 idvar = "ID", total = 1440)

  m1 <- brmcoda(compilr = 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(compilr = 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, acomp = TRUE)
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(compilr = compilr(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.compilr *Print a Summary for a compilr object*

Description

Print a Summary for a compilr object

Usage

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

Arguments

x An object of class compilr.
... Other arguments passed to [summary.compilr](#).

See Also

[summary.compilr](#)

Examples

```
cilr <- compilr(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(compilr = compilr(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)
}

```

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	<i>Possible Pairwise Substitutions</i>
------	--

Description

A dataset containing possible pairwise substitutions.

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	<i>Group-Level Estimates</i>
---------------	------------------------------

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(compilr = compilr(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 $N \times E$ 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(compilr = compilr(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

submargins

Average Marginal Substitution

Description

Using a fitted model object, estimates the the average marginal difference when compositional parts are substituted for specific unit(s). The submargins output encapsulates the substitution results for all compositional parts present in the [brmcoda](#) object.

Usage

```
submargins(object, delta, basesub, level = "total", type = "marginal", ...)
```

Arguments

object	A fitted brmcoda object.
delta	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
basesub	A <code>data.frame</code> or <code>data.table</code> of the base possible substitution of compositional parts. This data set can be computed using function basesub . If NULL, all possible pairwise substitution of compositional parts are used.
level	A character string or vector. Default to total.
type	A character string or vector. Should the estimate be conditional mean or average marginal mean?
...	Additional arguments to be passed to describe_posterior .

Value

A list containing the result of multilevel compositional substitution model. Each element of the list is the estimation for a compositional part and include at least six elements.

- 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.
- EffectType Either estimated conditional or average marginal changes.

substitution *Multilevel Compositional Substitution Model*

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 = NULL,
  summary = TRUE,
  ref = c("grandmean", "clustermean"),
  level = c("between", "within"),
  weight = c("equal", "proportional"),
  ...
)
```

Arguments

<code>object</code>	A fitted <code>brmcoda</code> object.
<code>delta</code>	A integer, numeric value or vector indicating the amount of substituted change between compositional parts.
<code>basesub</code>	A <code>data.frame</code> or <code>data.table</code> of the base possible substitution of compositional parts. This data set can be computed using function <code>basesub</code> . If <code>NULL</code> , all possible pairwise substitution of compositional parts are used.
<code>summary</code>	A logical value. Should the estimate at each level of the reference grid (<code>FALSE</code>) or their average (<code>TRUE</code>) be returned? Default is <code>TRUE</code> . Only applicable for model with covariates in addition to the isometric log-ratio coordinates (i.e., adjusted model).
<code>ref</code>	Either a character value or vector or a dataset. Can be "grandmean" and/or "clustermean", or a <code>data.frame</code> or <code>data.table</code> 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.
<code>level</code>	A character string or vector. Should the estimate be at the "between" and/or "within" level?
<code>weight</code>	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.
<code>...</code>	Additional arguments passed to <code>describe_posterior</code> .

Value

A list containing the results of multilevel compositional substitution model. The first four 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. Either between or within.
- Reference Either grandmean, clustermean, or users.

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- compilr(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(compilr = cilr,
              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.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(compile = compile(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.compile	<i>Create a Summary of a compile object</i>
-----------------	---

Description

Create a Summary of a compile object

Usage

```

## S3 method for class 'compile'
summary(
  object,
  class = c("composition", "logratio"),
  level = c("between", "within", "total"),
  weight = c("equal", "proportional"),
  digits = 3,
  ...
)

```

Arguments

object	An object of class compile.
class	Optional. Can be "composition" and/or "logratio" to specify the geometry of the composition.
level	Optional. Can be "between", "within", and/or "total" indicating the level of the geometry.
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.
digits	A integer value used for number formatting. Default is 3.
...	generic argument, not in use.

Examples

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

```
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(compilr = compilr(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, newcilr = NULL, ...)

```

Arguments

object	A fitted brmcoda object to be updated.
formula.	Changes to the formula; for details see update.formula and brmsformula .
newdata	A <code>data.frame</code> or <code>data.table</code> containing data of all variables used in the analysis. It must include a composition and the same ID variable as the existing compilr object.
newcilr	A compilr 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

- `CompILR` An object of class `compilr` 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(compilr = compilr(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.compilr

Update [compilr](#)**Description**

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

Usage

```

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

```

Arguments

object	A compilr class object to be updated.
newdata	A <code>data.frame</code> or <code>data.table</code> containing data of all variables used in the analysis. It must include a composition and the same ID variable as the existing compilr object.
...	generic argument, not in use.

Value

A `compilr` object with twelve elements.

- `BetweenComp` 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.
- `WithinComp` 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.
- `TotalComp` A vector of class `acomp` representing one closed total composition or a matrix of class `acomp` representing multiple closed total compositions each in one row.
- `BetweenILR` Isometric log ratio transform of between-person composition.
- `WithinILR` Isometric log ratio transform of within-person composition.
- `TotalILR` Isometric log ratio transform of total composition.
- `data` The user's dataset or imputed dataset if the input data contains zeros.
- `psi` A ILR matrix associated with user-defined partition structure.
- `sbp` The user-defined sequential binary partition matrix.
- `parts` Names of compositional variables.
- `idvar` Name of the variable containing IDs.
- `total` Total amount to which the compositions is closed.

See Also

`compilr`

Examples

```
cilr <- compilr(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)
```

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(compilr = compilr(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(compilr = compilr(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)
}
```

wsub

Within-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 *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 = NULL,
  ...
)
```

Arguments

object	A fitted brmcoda object.
basesub	A <code>data.frame</code> or <code>data.table</code> of the base possible substitution of compositional parts. This data set can be computed using function basesub . If <code>NULL</code> , all possible pairwise substitution of compositional parts are used.
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 (<code>FALSE</code>) or their average (<code>TRUE</code>) be returned? Default is <code>TRUE</code> . 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 <code>data.frame</code> or <code>data.table</code> 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.
level	A character string or vector. Should the estimate be at the "between" and/or "within" level?
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.
...	Additional arguments passed to describe_posterior .

Value

A list containing the results of multilevel compositional substitution model. The first four 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. Either between or within.
- Reference Either grandmean, clustermean, or users.

See Also

[substitution](#)

Examples

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

```

cilr <- compilr(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(compilr = 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 = 5)
}

```

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 = 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 data.frame or data.table of the base possible substitution of compositional parts. This data set can be computed using function basesub . If NULL, all possible pairwise substitution of compositional parts are used.
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.

level	A character string or vector. Should the estimate be at the "between" and/or "within" level?
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.
...	Additional arguments passed to describe_posterior .

Value

A list containing the results of multilevel compositional substitution model. The first four 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. Either between or within.
- Reference Either grandmean, clustermean, or users.

See Also

[substitution](#)

Examples

```
if(requireNamespace("cmdstanr")){
  cilr <- compilr(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(compilr = 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

- * **datasets**
 - mcompd, 20
 - psub, 29
 - sbp, 31
- as.data.frame, 3
- as.data.frame.compilr, 3
- as.matrix.compilr
 - (as.data.frame.compilr), 3
- basesub, 4, 6, 8, 32, 33, 42, 43
- bayes_factor.brmcoda, 4
- bayes_factor.brmsfit, 4, 5
- brm, 5, 37
- brmcoda, 5, 5, 6, 8, 10, 22, 23, 32, 33, 37, 38, 42, 43
- brmsfit, 22
- brmsformula, 5, 37
- bsub, 6
- bsubmargins, 8
- build.rg, 9

- coef (coef.brmcoda), 10
- coef.brmcoda, 10
- coef.brmsfit, 10, 11
- compilr, 5, 11, 12, 37–39

- describe_posterior, 7, 8, 32, 33, 42, 44
- diagnostic-quantities-brmcoda, 12
- diagnostic_posterior, 13
- diagnostic_posterior.brmcoda, 13
- draws-index-brmcoda, 14

- fitted (fitted.brmcoda), 15
- fitted.brmcoda, 15, 24
- fitted.brmsfit, 15
- fixef (fixef.brmcoda), 16
- fixef.brmcoda, 16
- fixef.brmsfit, 16, 17

- get-substitution, 17

- ggplot, 19, 24
- gtable, 23

- is.brmcoda, 17
- is.compilr, 18
- is.substitution, 18

- launch_shinystan, 19
- launch_shinystan
 - (launch_shinystan.brmcoda), 18
- launch_shinystan.brmcoda, 18
- launch_shinystan.brmsfit, 19
- log_posterior
 - (diagnostic-quantities-brmcoda), 12
- log_posterior.brmsfit, 13

- mcmc_plot.brmcoda, 19
- mcmc_plot.brmsfit, 19
- mcompd, 20
- mean.compilr, 20
- model.frame.brmcoda, 21

- nchains (draws-index-brmcoda), 14
- nchains.brmsfit, 14
- ndraws (draws-index-brmcoda), 14
- ndraws.brmsfit, 14
- neff_ratio
 - (diagnostic-quantities-brmcoda), 12
- neff_ratio.brmsfit, 13
- niterations (draws-index-brmcoda), 14
- niterations.brmsfit, 14
- nobs.brmcoda, 22
- nuts_params
 - (diagnostic-quantities-brmcoda), 12
- nuts_params.brmsfit, 13
- nvariables (draws-index-brmcoda), 14
- nvariables.brmsfit, 14

pairs, 22
pairs.brmcoda, 22
pairs.brmsfit, 22
plot.brmcoda, 23
plot.brmsfit, 23
plot.substitution, 24
posterior_epred.brmsfit, 15
posterior_predict.brmsfit, 25
posterior_summary, 10, 17, 29
predict (predict.brmcoda), 24
predict.brmcoda, 15, 24
predict.brmsfit, 25
predictive_error.brmsfit, 30
print.brmcoda, 26
print.compilr, 27
print.substitution, 27
prior_summary.brmcoda, 28
prior_summary.brmsfit, 28
psub, 29

ranef (ranef.brmcoda), 29
ranef.brmcoda, 29
ranef.brmsfit, 29, 30
residuals.brmcoda, 30
residuals.brmsfit, 30
rhat (diagnostic-quantities-brmcoda), 12
rhat.brmsfit, 13
runApp, 19

sbp, 31
submargins, 32
substitution, 6–9, 24, 33, 41–44
summary.brmcoda, 26, 34
summary.brmsfit, 34
summary.compilr, 27, 35
summary.substitution, 28, 36

update.brmcoda, 37
update.compilr, 38
update.formula, 37

VarCorr (VarCorr.brmcoda), 39
VarCorr.brmcoda, 39
VarCorr.brmsfit, 40
variables (draws-index-brmcoda), 14
variables.brmsfit, 14
vcov.brmcoda, 40
vcov.brmsfit, 40, 41

wsub, 41
wsubmargins, 43