

Package ‘coda.base’

May 8, 2026

Type Package

Title A Basic Set of Functions for Compositional Data Analysis

Version 1.0.6

Description A minimum set of functions to perform compositional data analysis using the log-ratio approach introduced by John Aitchison (1982). Main functions have been implemented in c++ for better performance.

URL <https://mcomas.net/coda.base/>, <https://github.com/mcomas/coda.base>

Depends R (>= 3.5)

Imports Rcpp (>= 0.12.12), stats, Matrix

LinkingTo Rcpp, RcppArmadillo

License GPL

Encoding UTF-8

LazyData true

NeedsCompilation yes

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 2.1.0), ggplot2, jsonlite

VignetteBuilder knitr

Author Marc Comas-Cufí [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-9759-0622>>)

Maintainer Marc Comas-Cufí <mcomas@imae.udg.edu>

Repository CRAN

Date/Publication 2026-05-08 14:10:02 UTC

Contents

alimentation	3
alr_basis	3
arctic_lake	4
blood_mn	5

bmi_activity	5
cc_basis	6
cdp_basis	6
cdp_partition	7
center	7
closure	8
clr_basis	9
coda.base	10
coda_replacement	10
composition	11
conditional_obasis	12
coordinates	14
dist	15
dist_coda	16
eurostat_employment	17
foraminiferals	18
gen_coda_with_zeros_and_missings	18
gmean	19
household_budget	20
house_expend	21
ilr_basis	21
kilauea_iki	22
mammals_milk	23
milk_cows	23
montana	24
pairwise_basis	24
parliament2017	25
partial_pb_constrained	25
partial_pb_exact	26
partial_pb_tabu_search	27
pb_basis	29
pb_tabu_search	30
pc_basis	31
perturbation	32
petrafm	33
plot_balance	33
pollen	35
pottery	35
powering	36
random_composition_with_fixed_pb	37
read_cdp	38
sbp_basis	38
serprot	39
statistician_time	40
variation_array	40
waste	41
weibo_hotels	42

`alimentation`*Food consumption in European countries*

Description

The ‘alimentation’ data set contains the percentage composition of food consumption in 25 European countries during the 1980s. The food categories are:

- ‘RM’: red meat (pork, veal, beef),
- ‘WM’: white meat (chicken),
- ‘E’: eggs,
- ‘M’: milk,
- ‘F’: fish,
- ‘C’: cereals,
- ‘S’: starch (potatoes),
- ‘N’: nuts,
- ‘FV’: fruits and vegetables.

The data set also contains categorical variables indicating whether the country belongs to the North or South/Mediterranean group, and whether it is an Eastern or Western European country.

Usage`alimentation`**Format**

An object of class `data.frame` with 25 rows and 13 columns.

`alr_basis`*Additive log-ratio basis*

Description

Construct the transformation matrix associated with additive log-ratio (alr) coordinates.

Usage

```
alr_basis(dim, denominator = NULL, numerator = NULL)
```

Arguments

dim	Number of parts. It can be a single integer, a matrix or data frame, or a character vector of part names.
denominator	Part used as denominator. By default, the last part is used.
numerator	Parts used as numerators. By default, all parts except the denominator are used, preserving their original order.

Value

A matrix defining the alr coordinate system.

References

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. Chapman & Hall, London.

Examples

```
alr_basis(5)
alr_basis(5, 3)
alr_basis(5, 3, c(1, 5, 2, 4))
```

arctic_lake

Arctic lake sediments at different depths

Description

The ‘arctic_lake’ data set records the three-part composition [*sand, silt, clay*] of 39 sediment samples collected at different water depths in an Arctic lake.

Usage

```
arctic_lake
```

Format

An object of class `data.frame` with 39 rows and 5 columns.

blood_mn *The MN blood system*

Description

In humans, the main blood group systems are the ABO system, the Rh system, and the MN system. The MN blood system is related to proteins of the red blood cell plasma membrane. Its inheritance pattern is autosomal with codominance, meaning that the heterozygous phenotype is distinct from both homozygous phenotypes.

The three phenotypes are M, N, and MN. Their frequencies vary across populations. Under the Hardy-Weinberg principle, allele and genotype frequencies remain constant across generations in the absence of evolutionary forces, implying that

$$\frac{x_{MM}x_{NN}}{x_{MN}^2} = \frac{1}{4}$$

where x_{MM} and x_{NN} are the genotype frequencies of the homozygotes and x_{MN} is the genotype frequency of heterozygotes.

Usage

blood_mn

Format

An object of class `data.frame` with 49 rows and 5 columns.

bmi_activity *Physical activity and body mass index*

Description

The ‘bmi_activity’ data set records the proportion of daily time spent in sleep (‘sleep’), sedentary behaviour (‘sedent’), light physical activity (‘Lpa’), moderate physical activity (‘Mpa’), and vigorous physical activity (‘Vpa’) for 393 children. The standardized body mass index (‘zBMI’) of each child is also included.

This data set was used in the example of Dumuid et al. (2019) to examine the expected differences in zBMI associated with reallocations of daily time between sleep, sedentary behaviour, and physical activity. Because the original data are confidential, ‘bmi_activity’ contains simulated data that mimic the main features of the original study.

Usage

bmi_activity

Format

An object of class `data.frame` with 393 rows and 8 columns.

References

Dumuid, D., Pedisic, Z., Stanford, T. E., Martín-Fernández, J. A., Hron, K., Maher, C., Lewis, L. K., & Olds, T. S. (2019). *The Compositional Isotemporal Substitution Model: a Method for Estimating Changes in a Health Outcome for Reallocation of Time between Sleep, Sedentary Behaviour, and Physical Activity*. *Statistical Methods in Medical Research*, **28**(3), 846–857.

cc_basis	<i>Canonical-correlation log-ratio basis</i>
----------	--

Description

Construct an ilr basis rotated according to canonical correlations between a compositional response data set and an explanatory data set.

Usage

```
cc_basis(Y, X)
```

Arguments

Y	A compositional data set.
X	An explanatory data set.

Value

A matrix whose columns define a canonical-correlation-oriented ilr basis.

cdp_basis	<i>CoDaPack default ilr basis</i>
-----------	-----------------------------------

Description

Construct the default isometric log-ratio basis used in CoDaPack.

Usage

```
cdp_basis(dim)
```

Arguments

dim	Number of parts. It can be a single integer, a matrix or data frame, or a character vector of part names.
-----	---

Value

A matrix with D rows and $D - 1$ columns containing the CoDaPack default ilr basis.

Examples

```
cdp_basis(5)
cdp_basis(c("a", "b", "c", "d"))
```

cdp_partition	<i>CoDaPack's default binary partition</i>
---------------	--

Description

Compute the default binary partition used in CoDaPack's software

Usage

```
cdp_partition(ncomp)
```

Arguments

ncomp	number of parts
-------	-----------------

Value

matrix

Examples

```
cdp_partition(4)
```

center	<i>Dataset center</i>
--------	-----------------------

Description

Generic function to calculate the center of a compositional dataset

Usage

```
center(X, zero.rm = FALSE, na.rm = FALSE)
```

Arguments

X	compositional dataset
zero.rm	a logical value indicating whether zero values should be stripped before the computation proceeds.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Examples

```
X = matrix(exp(rnorm(5*100)), nrow=100, ncol=5)
g = rep(c('a','b','c','d'), 25)
center(X)
(by_g <- by(X, g, center))
center(t(simplify2array(by_g)))
```

 closure

Closure operation for compositional data

Description

Applies the closure operation to a numeric vector, matrix or data frame so that each composition sums to a prescribed constant k .

Usage

```
closure(X, k = 1)
```

Arguments

X	A numeric vector, matrix, data frame, or an object coercible to one of these. For matrices and data frames, rows are interpreted as compositions.
k	A numeric vector of length 1 or length $nrow(X)$. Must contain strictly positive values.

Details

If X is:

- a vector, the returned vector sums to k ;
- a matrix or data frame, closure is applied row-wise, and each row sums to the corresponding value of k .

The argument k may be:

- a single positive number, recycled to all rows;
- a numeric vector of length $nrow(X)$, specifying a different closure constant for each row.

For a composition $x = (x_1, \dots, x_D)$ with positive sum, the closure to constant k is

$$C(x) = k \frac{x}{\sum_{j=1}^D x_j}.$$

This function requires all entries of X to be finite and non-negative, and every row sum (or the vector sum) must be strictly positive.

Value

If X is a vector, a numeric vector of the same length.

If X is a matrix, a numeric matrix with the same dimensions, dimnames, and row-wise sums equal to k .

If X is a data frame, a data frame with the same row and column names, and row-wise sums equal to k .

Examples

```
closure(c(2, 3, 5))
closure(c(2, 3, 5), k = 100)

X <- matrix(c(1, 1, 2,
              2, 3, 5), nrow = 2, byrow = TRUE)
closure(X)
closure(X, k = c(1, 100))

df <- data.frame(a = c(1, 2), b = c(1, 3), c = c(2, 5))
closure(df, k = 10)
```

 clr_basis

Centered log-ratio basis

Description

Construct the transformation matrix associated with centered log-ratio (clr) coordinates.

Usage

```
clr_basis(dim)
```

Arguments

dim Number of parts. It can be a single integer, a matrix or data frame, or a character vector of part names.

Details

CLR coordinates are linearly dependent and lie in the $D - 1$ dimensional clr-plane.

Value

A square matrix defining the clr coordinate system.

References

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. Chapman & Hall, London.

Examples

```
B <- clr_basis(5)
clr_coordinates <- coordinates(c(1, 2, 3, 4, 5), B)
sum(clr_coordinates) < 1e-15
```

coda.base

coda.base

Description

A minimum set of functions to perform compositional data analysis using the log-ratio approach introduced by John Aitchison (1982) <<https://www.jstor.org/stable/2345821>>. Main functions have been implemented in c++ for better performance.

Author(s)

Marc Comas-Cufí

See Also

Useful links:

- <https://mcomas.net/coda.base/>
- <https://github.com/mcomas/coda.base>

coda_replacement

Replacement of missing values and below-detection zeros in compositional data

Description

Performs imputation of missing values and/or values below the detection limit in compositional data using an EM algorithm assuming normality on the simplex.

Usage

```
coda_replacement(  
  X,  
  DL = NULL,  
  dl_prop = 0.65,  
  eps = 1e-04,  
  parameters = FALSE,  
  debug = FALSE,  
  maxit = 500  
)
```

Arguments

<code>X</code>	A compositional data set: numeric matrix or data frame where rows represent observations and columns represent parts.
<code>DL</code>	An optional matrix or vector of detection limits. If 'NULL', the minimum non-zero value in each column of 'X' is used.
<code>dl_prop</code>	A numeric value between 0 and 1 used for initialization in the EM algorithm.
<code>eps</code>	Convergence tolerance.
<code>parameters</code>	Logical; if 'TRUE', return additional estimated parameters.
<code>debug</code>	Logical; if 'TRUE', print the log-likelihood at each iteration.
<code>maxit</code>	Maximum number of iterations

Value

If 'parameters = FALSE', the imputed object with the same format as 'X' ('matrix' or 'data.frame', preserving data-frame subclasses when possible) and preserving original names. If 'parameters = TRUE', a list with the estimated clr mean, clr covariance, and imputed clr coordinates.

Examples

```
X <- matrix(c(
  0.00, 0.30, 0.70,
  0.20, NA, 0.80,
  0.40, 0.60, 0.00,
  0.25, 0.25, 0.50,
  0.10, 0.30, 0.60
), ncol = 3, byrow = TRUE)
colnames(X) <- c("sand", "silt", "clay")

DL <- c(0.05, 0.05, 0.05)

X_imp <- coda_replacement(X, DL = DL, maxit = 20)
X_imp

set.seed(10)
X <- composition(matrix(rnorm(3*10), ncol = 3))
X[sample(c(TRUE, FALSE), 4*10, replace = TRUE, c(1, 3))] <- NA
params <- coda_replacement(X, parameters = TRUE, debug = TRUE)
names(params)
params$clr_mu
composition(params$clr_h)
```

Description

Reconstruct a composition from coordinates with respect to a given basis.

Usage

```
composition(H, basis = "ilr")
```

```
comp(H, basis = "ilr")
```

Arguments

H	Coordinates of a composition. It can be a numeric matrix, a data frame, or a numeric vector.
basis	Basis used to interpret the coordinates. Either a character string naming a pre-defined basis or a matrix.

Value

A composition corresponding to the given coordinates.

See Also

[coordinates](#), [ilr_basis](#), [alr_basis](#), [clr_basis](#), [sbp_basis](#)

conditional_obasis	<i>Conditional orthonormal basis</i>
--------------------	--------------------------------------

Description

Compute orthonormal ilr bases adapted to row-wise conditioning patterns.

Usage

```
conditional_obasis(X, scheme = c("zero", "zero_na"))
```

Arguments

X	A numeric matrix or data frame with one observation or conditioning pattern per row and one part per column.
scheme	Character string indicating the conditioning scheme. Possible values are "zero" and "zero_na". Default is "zero".

Details

Each row of 'X' defines one conditioning pattern on the parts of a composition. According to 'scheme', the parts are split into ordered blocks:

- "zero": parts equal to '0' and parts with strictly positive values,
- "zero_na": missing values ('NA'), zeros, and strictly positive values.

For each row, the function constructs an orthonormal basis of the clr-plane preserving the block structure induced by the selected scheme.

Under 'scheme = "zero"', if a row contains 'nz' zeros, then:

- the first 'nz - 1' coordinates describe the internal log-ratio structure of the zero block,
- the coordinate 'nz' describes the balance between the zero block and the positive block,
- the remaining coordinates describe the internal log-ratio structure of the positive block.

Under 'scheme = "zero_na"', the blocks are ordered as:

- missing values ('NA'),
- zeros,
- strictly positive values.

In this case:

- the first coordinates describe the internal structure of the 'NA' block,
- the next coordinate contrasts the 'NA' block with the positive block,
- the following coordinates describe the internal structure of the zero block,
- the next coordinate contrasts the zero block with the positive block,
- the remaining coordinates describe the internal structure of the positive block.

Value

A three-dimensional array of dimension '(D - 1, D, nrow(X))', where 'D' is the number of parts. Each slice contains one orthonormal ilr basis.

Examples

```
C <- rbind(
  c(0, 0, 1, 1, 0),
  c(0, 1, 0, 1, 0)
)
```

```
conditional_obasis(C)
```

```
X <- rbind(
  c(1, NA, 0, 2),
  c(NA, 3, 0, 4),
  c(1, 2, 3, 4)
)
```

```
conditional_obasis(X, scheme = "zero_na")
```

coordinates

Coordinates of compositions with respect to a basis

Description

Compute coordinates of a composition or a compositional data set with respect to a given log-ratio basis.

The ‘basis’ argument can be either:

- a character string identifying a predefined coordinate system, or
- a matrix whose columns define a system of log-contrasts.

The predefined options are:

- “ilr”: isometric log-ratio coordinates,
- “olr”: orthonormal log-ratio coordinates,
- “clr”: centered log-ratio coordinates,
- “alr”: additive log-ratio coordinates,
- “pw”: pairwise log-ratios,
- “pc”: principal component log-ratio coordinates,
- “pb”: principal balance coordinates,
- “cdp”: CoDaPack default balances.

Usage

```
coordinates(X, basis = "ilr")
```

```
coord(..., basis = "ilr")
```

```
alr_c(X)
```

```
clr_c(X)
```

```
ilr_c(X)
```

```
olr_c(X)
```

Arguments

X	A compositional data set. It can be a numeric matrix, a data frame, or a numeric vector.
basis	Basis used to compute the coordinates. Either a character string naming a predefined basis or a matrix with log-ratio basis vectors in columns.
...	components of the composition

Value

Coordinates of ‘X’ with respect to the given ‘basis’. The returned object has the same general type as the input when possible.

See Also

[ilr_basis](#), [alr_basis](#), [clr_basis](#), [sbp_basis](#), [composition](#)

Examples

```
coordinates(1:5)

B <- ilr_basis(5)
coordinates(1:5, B)

X <- rbind(1:5, 2:6)
coordinates(X, "clr")
```

 dist

Distance Matrix Computation (including Aitchison distance)

Description

Compute a distance matrix for compositional data, including the Aitchison distance as an extension of [dist](#).

Usage

```
dist(x, method = "euclidean", ...)
```

Arguments

x	A data matrix whose rows are compositions.
method	The distance measure to be used. This must be one of "aitchison", "euclidean", "maximum", "manhattan", "canberra", "binary", or "minkowski". Any unambiguous abbreviation can be given.
...	Additional arguments passed to dist .

Value

An object of class "dist".

See Also

[dist_coda](#), [dist](#)

Examples

```
X <- exp(matrix(rnorm(10 * 50), ncol = 50, nrow = 10))

(d <- dist_coda(X, method = "aitchison"))
plot(hclust(d))

# In contrast to Euclidean distance
dist(rbind(c(1, 1, 1), c(100, 100, 100)), method = "euc")

# Using Aitchison distance, only relative information is of importance
dist_coda(rbind(c(1, 1, 1), c(100, 100, 100)), method = "ait")
```

dist_coda

Distance Matrix Computation for CoDa distances

Description

Compute a distance matrix for compositional data using selected CoDa distances.

Usage

```
dist_coda(x, method = "aitchison", ...)
```

Arguments

x	A data matrix whose rows are compositions.
method	The distance measure to be used. This must be one of "aitchison", "L1", "L1-pw", or "L1-clr". Any unambiguous abbreviation can be given.
...	Additional arguments. diag and upper are passed to as.dist for L1 distances and all arguments are passed to dist for the Aitchison distance.

Value

An object of class "dist".

References

Saperas-Riera, J.; Mateu-Figueras, G.; Martín-Fernández, J.A. (2024). *Lp-Norm for Compositional Data: Exploring the CoDa L1-Norm in Penalised Regression*. *Mathematics*, 12(9), 1388. [doi:10.3390/math12091388](https://doi.org/10.3390/math12091388).

See Also

[dist](#), [dist](#)

Examples

```
set.seed(1)
X <- exp(matrix(rnorm(10 * 5), ncol = 5, nrow = 10))

dist_coda(X, method = "aitchison")
dist_coda(X, method = "L1")
dist_coda(X, method = "L1-pw")
dist_coda(X, method = "L1-clr")
```

eurostat_employment *Employment distribution in EUROSTAT countries*

Description

According to the three-sector theory, employment shifts from the primary sector (raw material extraction), to the secondary sector (industry, energy, and construction), and then to the tertiary sector (services) as economies develop. The ‘eurostat_employment’ data set contains EUROSTAT data on employment, aggregated for both sexes and all ages, distributed by economic activity in 2008 for 29 EUROSTAT member countries.

A related variable is the logarithm of gross domestic product per person in EUR at current prices (‘logGDP’). For exploratory purposes, it is also categorised as a binary variable indicating values above or below the median (‘Binary GDP’).

The employment composition has 11 parts:

- Primary sector
- Manufacturing
- Energy
- Construction
- Trade repair transport
- Hotels restaurants
- Financial intermediation
- Real estate
- Educ admin defense soc sec
- Health social work
- Other services

Usage

```
eurostat_employment
```

Format

An object of class `data.frame` with 29 rows and 17 columns.

foraminiferals	<i>Paleocological compositions</i>
----------------	------------------------------------

Description

The ‘foraminiferals’ data set (Aitchison, 1986) is a classical example of paleocological compositional data. It contains the composition of four fossil types (Neogloboquadrina atlantica, Neogloboquadrina pachyderma, Globorotalia obesa, and Globigerinoides triloba) at 30 different depths.

Because the data contain rounded zeros, zero-replacement techniques are typically required before analysis. A natural goal is then to study the association between fossil composition and depth.

Usage

```
foraminiferals
```

Format

An object of class `data.frame` with 30 rows and 5 columns.

```
gen_coda_with_zeros_and_missings
```

Generate compositional data with zeros and missing values

Description

Simulate compositional data and optionally introduce structural zeros (interpreted as values below a detection limit) and missing values.

The function first generates a compositional data set ‘X0’, then creates a modified version ‘X’ by:

- replacing values below ‘dl_par’ by zero, if ‘zeros = TRUE’,
- introducing missing values at random, if ‘missings = TRUE’.

A matrix of detection limits ‘DL’ is also returned. It contains ‘dl_par’ in the positions that were censored to zero, and ‘0’ elsewhere.

Usage

```
gen_coda_with_zeros_and_missings(
  n,
  d,
  missings = TRUE,
  zeros = TRUE,
  dl_par = 0.05,
  na_p = 0.15
)
```

Arguments

n	Number of observations.
d	Dimension of the latent coordinate space used to generate the compositions.
missings	Logical; if 'TRUE', introduce missing values at random.
zeros	Logical; if 'TRUE', replace values below 'dl_par' by zero.
dl_par	Detection-limit threshold used to generate zeros.
na_p	Probability that any entry is replaced by 'NA' when 'missings = TRUE'.

Details

Compositions are generated from multivariate normal coordinates and mapped to the simplex through 'composition()'. The eigenvector rotation is included to induce a non-trivial covariance structure in the generated coordinates.

Missing values are introduced completely at random, independently for each cell, with probability 'na_p'.

Value

A list with three components:

X The generated compositional data set with simulated zeros and/or missing values.

DL A matrix of detection limits, with 'dl_par' in censored positions and '0' elsewhere.

X0 The original simulated compositional data set before introducing zeros or missing values.

Examples

```
set.seed(123)
sim <- gen_coda_with_zeros_and_missings(100, 4)

str(sim)
summary(sim$X0)
summary(sim$X)
table(sim$X == 0, useNA = "ifany")
```

gmean

Geometric Mean

Description

Generic function for the (trimmed) geometric mean.

Usage

```
gmean(x, zero.rm = FALSE, trim = 0, na.rm = FALSE)
```

Arguments

<code>x</code>	A nonnegative vector.
<code>zero.rm</code>	a logical value indicating whether zero values should be stripped before the computation proceeds.
<code>trim</code>	the fraction (0 to 0.5) of observations to be trimmed from each end of <code>x</code> before the mean is computed. Values of <code>trim</code> outside that range are taken as the nearest endpoint.
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.

See Also

[center](#)

household_budget	<i>Household budget patterns</i>
------------------	----------------------------------

Description

In a sample survey of single persons living alone in rented accommodation, twenty men and twenty women were randomly selected and asked to record their expenditure over one month in the following four mutually exclusive and exhaustive commodity groups:

- ‘Hous’: housing, including fuel and light,
- ‘Food’: foodstuffs, including alcohol and tobacco,
- ‘Serv’: services, including transport and vehicles,
- ‘Other’: other goods, including clothing, footwear, and durable goods.

Usage

```
household_budget
```

Format

An object of class `data.frame` with 40 rows and 6 columns.

house_expend	<i>Household expenditures</i>
--------------	-------------------------------

Description

The ‘house_expend’ data set, obtained from Eurostat, records the composition of mean household consumption expenditure across 12 expenditure categories in 27 European Union countries. Some values are rounded zeros.

In addition, the data set contains gross domestic product values for 2005 (‘GDP05’) and 2014 (‘GDP14’). A relevant analysis is the relationship between expenditure compositions and GDP.

Usage

```
house_expend
```

Format

An object of class `data.frame` with 27 rows and 15 columns.

ilr_basis	<i>Isometric and orthonormal log-ratio bases</i>
-----------	--

Description

Construct an isometric log-ratio (ilr) basis for a composition with D parts. The ilr basis is an orthonormal basis of the clr-plane and provides $D - 1$ coordinates. The same basis is sometimes referred to as an orthonormal log-ratio (olr) basis.

Usage

```
ilr_basis(dim, type = "default")
```

```
olr_basis(dim, type = "default")
```

Arguments

dim	Number of parts. It can be: <ul style="list-style-type: none"> • a single integer, • a matrix or data frame, in which case the number of columns is used, • a character vector of part names, in which case its length is used.
type	Type of ilr basis to construct. Available options are: <ul style="list-style-type: none"> • "default": standard Helmert-type ilr basis, • "pivot": pivot balance basis, • "cdp": CoDaPack default basis.

Details

For ‘type = "default"’, the function returns the standard Helmert-type ilr basis. Alternative constructions are available through ‘type = "pivot"’ and ‘type = "cdp"’.

The default basis vectors are:

$$h_i = \sqrt{\frac{i}{i+1}} \log \frac{\sqrt[i]{\prod_{j=1}^i x_j}}{x_{i+1}}, \quad i = 1, \dots, D-1$$

Value

A matrix with D rows and $D-1$ columns representing an orthonormal log-ratio basis.

References

Egozcue, J. J., Pawlowsky-Glahn, V., Mateu-Figueras, G., & Barceló-Vidal, C. (2003). *Isometric logratio transformations for compositional data analysis*. *Mathematical Geology*, **35**(3), 279–300.

Examples

```
ilr_basis(5)
ilr_basis(alimentation[, 1:9])
ilr_basis(c("a", "b", "c", "d"), type = "pivot")
```

kilauea_iki

Chemical composition of volcanic rocks from Kilauea Iki

Description

The ‘kilauea_iki’ data set contains the chemical composition of volcanic rocks sampled from the lava lake at Kilauea Iki (Hawaii). The data represent major oxide concentrations in fractional form.

Usage

```
kilauea_iki
```

Format

A data frame with 17 observations and 11 variables:

SiO2 Silicon dioxide

TiO2 Titanium dioxide

Al2O3 Aluminium oxide

Fe2O3 Ferric oxide

FeO Ferrous oxide

MnO Manganese oxide

MgO Magnesium oxide
CaO Calcium oxide
Na₂O Sodium oxide
K₂O Potassium oxide
P₂O₅ Phosphorus pentoxide

Details

The variability in oxide concentrations is attributed to magnesian olivine fractionation from a single magmatic mass, as suggested by Richter and Moore (1966).

Source

Richter, D. H., & Moore, J. G. (1966). *Petrology of Kilauea Iki lava lake, Hawaii*. Geological Survey Professional Paper 537-B.

mammals_milk	<i>Mammals' milk</i>
--------------	----------------------

Description

The 'mammals_milk' data set contains the percentages of five constituents of the milk of 24 mammals: [W, P, F, L, A], where 'W' is water, 'P' is protein, 'F' is fat, 'L' is lactose, and 'A' is ash.

Usage

mammals_milk

Format

An object of class `data.frame` with 24 rows and 6 columns.

milk_cows	<i>Milk composition study</i>
-----------	-------------------------------

Description

In an attempt to improve the quality of cow milk, milk from thirty cows was assessed before and after a controlled dietary and hormonal regime over eight weeks. A control group of thirty cows kept under the usual regime was also included.

The 'milk_cows' data set provides the complete before/after milk composition data for the sixty cows, with the proportions of protein ('pr'), milk fat ('mf'), carbohydrate ('ch'), calcium ('Ca'), sodium ('Na'), and potassium ('K').

Usage

```
milk_cows
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 116 rows and 10 columns.

montana	<i>Concentration of minor elements in coal ashes</i>
---------	--

Description

The ‘montana’ data set contains 229 samples of the concentration (in ppm) of five minor elements [*Cr, Cu, Hg, U, V*] in coal ashes from the Fort Union formation (Montana, USA), in the Powder River Basin.

The five measured elements form a fully observed subcomposition of a much larger chemical composition. Since the data are given in parts per million and all concentrations were measured, a residual component could in principle be added to close the compositions to 10^6 .

Usage

```
montana
```

Format

An object of class `data.frame` with 229 rows and 6 columns.

pairwise_basis	<i>Pairwise log-ratio generating system</i>
----------------	---

Description

Construct the system of all pairwise log-ratios between parts.

Usage

```
pairwise_basis(dim)
```

Arguments

dim	Number of parts. It can be a single integer, a matrix or data frame, or a character vector of part names.
-----	---

Value

A matrix, or a sparse matrix for large dimensions, whose columns represent all pairwise log-ratio generators.

parliament2017	<i>Catalan Parliament election results in 2017 by region</i>
----------------	--

Description

The 'parliament2017' data set contains the results of the 2017 Catalan Parliament election aggregated by region.

Usage

```
parliament2017
```

Format

A data frame with 42 rows and 9 variables:

com Region
cs Votes for the Ciutadans party
jxcat Votes for the Junts per Catalunya party
erc Votes for the Esquerra Republicana de Catalunya party
psc Votes for the Partit Socialista de Catalunya party
catsp Votes for the Catalunya Sí que es Pot party
cup Votes for the Candidatura d'Unitat Popular party
pp Votes for the Partit Popular party
other Votes for other parties

Source

Idescat, statistics on Catalan Parliament elections.

partial_pb_constrained	<i>Constrained search for a partial principal balance on grouped parts</i>
------------------------	--

Description

Builds a single grouped constrained principal balance from the first principal component of the grouped composition.

Usage

```
partial_pb_constrained(X, lI = NULL, constrained.criterion = "variance")
```

Arguments

- `X` A numeric matrix with strictly positive finite entries. Rows are observations and columns are compositional parts.
- `lI` A list defining a partition of a subset of the columns of `X`. If `NULL`, each column of `X` is used as a singleton group.
- `constrained.criterion` Criterion used to choose the constrained balance. Either "variance" (default) or "angle".

Value

A list with the following elements:

- `dim` Dimension of the grouped problem, equal to `length(lI) - 1`.
- `lI` The input grouping structure.
- `variance` Variance criterion of the selected grouped balance.
- `balance_raw` Integer vector in $\{-1, 0, 1\}$ describing the selected grouped split.
- `balance` The corresponding one-column balance basis.
- `constrained.criterion` Criterion used to construct the balance.

`partial_pb_exact` *Exact search for a partial principal balance on grouped parts*

Description

Finds the grouped balance with maximum variance among all assignments whose number of active groups is between `min_parts` and `max_parts`.

Usage

```
partial_pb_exact(
  X,
  lI = NULL,
  min_parts = 2,
  max_parts = NULL,
  method = "restricted"
)
```

Arguments

- `X` A numeric matrix with strictly positive finite entries. Rows are observations and columns are compositional parts.
- `lI` A list defining a partition of a subset of the columns of `X`. If `NULL`, each column of `X` is used as a singleton group.

min_parts	Integer. Minimum number of active groups.
max_parts	Integer or NULL. Maximum number of active groups. If NULL, all groups may be active.
method	Exhaustive search method. Currently only "restricted" is implemented; it enumerates only supports whose sizes are inside the requested range and assigns signs in binary Gray-code order.

Details

The search enumerates only supports whose size is between `min_parts` and `max_parts`. For each support, signs are generated in binary Gray-code order, fixing the first active group on the left side to avoid evaluating both a balance and its sign reversal.

Value

A list with the following elements:

<code>dim</code>	Dimension of the grouped problem, equal to <code>length(lI) - 1</code> .
<code>lI</code>	The input grouping structure.
<code>variance</code>	Variance criterion of the best grouped balance.
<code>balance_raw</code>	Integer vector in $\{-1, 0, 1\}$ describing the best grouped split.
<code>balance</code>	The corresponding one-column balance basis.
<code>min_parts</code>	Minimum number of active groups.
<code>max_parts</code>	Maximum number of active groups.

partial_pb_tabu_search

Tabu search for a partial principal balance on grouped parts

Description

Finds a single grouped balance by tabu search over a partition of selected parts. The search is carried out on groups of parts defined by `lI`, using configurable neighbourhood moves.

Usage

```
partial_pb_tabu_search(
  X,
  lI = NULL,
  min_parts = 2,
  max_parts = NULL,
  iter = 100,
  tabu_size = length(lI),
  ini = NULL,
  remove_active = TRUE,
```

```

    add_left = TRUE,
    add_right = TRUE,
    flip_side = FALSE,
    swap_zero = FALSE,
    swap_sides = FALSE,
    debug = FALSE,
    constrained.criterion = "variance"
)

```

Arguments

<code>X</code>	A numeric matrix with strictly positive finite entries. Rows are observations and columns are compositional parts.
<code>II</code>	A list defining a partition of a subset of the columns of <code>X</code> . If <code>NULL</code> , each column of <code>X</code> is used as a singleton group.
<code>min_parts</code>	Integer. Minimum number of active groups.
<code>max_parts</code>	Integer or <code>NULL</code> . Maximum number of groups from <code>II</code> allowed to be active in the balance. If <code>NULL</code> , all groups may be active.
<code>iter</code>	Integer. Maximum number of tabu search iterations.
<code>tabu_size</code>	Integer. Maximum size of the tabu list.
<code>ini</code>	Initial grouped split. If <code>NULL</code> , the constrained principal balance of the grouped subcomposition is used.
<code>remove_active</code>	Logical. Allow moves from -1 or $+1$ to 0 .
<code>add_left</code>	Logical. Allow moves from 0 to -1 .
<code>add_right</code>	Logical. Allow moves from 0 to $+1$.
<code>flip_side</code>	Logical. Allow direct moves from -1 to $+1$ and from $+1$ to -1 .
<code>swap_zero</code>	Logical. Allow swaps between one active group and one inactive group, preserving the active side.
<code>swap_sides</code>	Logical. Allow swaps between one left group and one right group.
<code>debug</code>	Logical. If <code>TRUE</code> , progress information is printed during the search.
<code>constrained.criterion</code>	Criterion used to initialise the constrained balance when <code>ini = NULL</code> . Either "variance" (default) or "angle".

Details

When `ini = NULL`, the constrained grouped balance is adjusted greedily so that the initial solution has exactly `max_parts` active groups.

Value

A list with the selected balance, its variance criterion, the search path, and a neighbourhoods element recording the active neighbourhood types.

pb_basis	<i>Principal balance basis</i>
----------	--------------------------------

Description

Construct a basis of principal balances for a compositional data set.

Usage

```
pb_basis(
  X,
  method,
  constrained.criterion = "variance",
  cluster.method = "ward.D2",
  ordering = TRUE,
  ...
)
```

Arguments

<code>X</code>	Compositional data set.
<code>method</code>	Method used to construct the principal balances. One of "exact", "exact2", "constrained", or "cluster".
<code>constrained.criterion</code>	Criterion used by the constrained method. Either "variance" (default) or "angle".
<code>cluster.method</code>	Linkage criterion passed to <code>hclust</code> when 'method = "cluster"'.
<code>ordering</code>	Logical; if 'TRUE', reorder balances by decreasing explained variance.
<code>...</code>	Additional arguments passed to <code>hclust</code> .

Details

Several methods are available:

- "exact": exact computation of principal balances,
- "exact2": exact computation using incremental Gray-code updates,
- "constrained": constrained approximation based on a target criterion,
- "cluster": approximation based on hierarchical clustering.

Value

A matrix whose columns are principal balances.

References

Martín-Fernández, J. A., Pawlowsky-Glahn, V., Egozcue, J. J., & Tolosana-Delgado, R. (2018). *Advances in Principal Balances for Compositional Data*. *Mathematical Geosciences*, 50, 273–298.

Examples

```

set.seed(1)
X <- matrix(exp(rnorm(5 * 100)), nrow = 100, ncol = 5)

v1 <- apply(coordinates(X, "pc"), 2, var)
v2 <- apply(coordinates(X, pb_basis(X, method = "exact")), 2, var)
v3 <- apply(coordinates(X, pb_basis(X, method = "constrained")), 2, var)
v4 <- apply(coordinates(X, pb_basis(X, method = "cluster")), 2, var)

barplot(
  rbind(v1, v2, v3, v4),
  beside = TRUE,
  ylim = c(0, 2),
  legend = c(
    "Principal Components",
    "PB (Exact method)",
    "PB (Constrained)",
    "PB (Ward approximation)"
  ),
  names = paste0("Comp.", 1:4),
  args.legend = list(cex = 0.8),
  ylab = "Variance"
)

```

pb_tabu_search

Tabu search approximation to a sequential binary partition

Description

Builds a sequential binary partition (SBP) by repeatedly applying grouped tabu search to select balances over the current sets of parts. At each step, the best candidate split is retained and the remaining candidate subproblems are explored until an SBP with $D - 1$ balances is obtained.

Usage

```
pb_tabu_search(X, iter = 100, debug = FALSE)
```

Arguments

X	A numeric matrix with strictly positive finite entries. Rows are observations and columns are compositional parts.
iter	Integer. Maximum number of tabu search iterations used in each partial search.
debug	Logical. If TRUE, progress information is printed during the partial searches.

Details

This function provides a heuristic approximation to a principal balance basis. The first balance is searched on the full set of parts, and the subsequent balances are obtained by recursively refining the best currently available split.

All partial searches are initialized with the constrained principal balance of the corresponding grouped composition.

The procedure starts from the trivial grouping where each part forms its own singleton group. After each partial tabu search, up to three candidate subproblems may be generated from the selected solution:

- the split between active and inactive groups,
- the left active branch,
- the right active branch.

All generated candidates are stored, and at each stage the candidate with the largest variance criterion is selected for inclusion in the SBP and for further refinement.

This is a heuristic search strategy and does not guarantee a globally optimal SBP.

Value

An integer matrix representing a sequential binary partition. Rows correspond to the original parts of X and columns correspond to balances. Entries are in $\{-1, 0, 1\}$. The returned matrix has attribute "max_steps", giving the largest iteration index at which a best partial solution was found among all partial searches performed.

See Also

[partial_pb_tabu_search](#), [sbp_basis](#)

Examples

```
set.seed(1)
X <- matrix(rexp(500), ncol = 5)

SBP <- pb_tabu_search(X, iter = 30)
SBP
attr(SBP, "max_steps")
```

pc_basis

Principal component log-ratio basis

Description

Construct an ilr basis rotated according to the principal components of the log-ratio coordinates of a compositional data set.

Usage

```
pc_basis(X)
```

Arguments

X Compositional data set.

Value

A matrix whose columns define a principal-component-oriented ilr basis.

perturbation

Perturbation of compositional data

Description

The perturbation operation combines two compositions by component-wise multiplication and then applies closure to ensure the result remains a valid composition.

Usage

```
perturbation(X, Y)
```

Arguments

X A numeric vector, matrix or data.frame containing compositions.

Y A numeric vector, matrix or data.frame with the same number of parts as X. If one input is a matrix or data.frame and the other is a vector, the vector is applied to every row. If both inputs are matrices or data.frames, they must have the same dimensions, or Y may contain a single composition to be applied to all rows of X.

Details

Perturbation is the analogue of addition in the simplex. Each part of X is multiplied by the corresponding part of Y, and the result is closed with [closure](#) so that each composition has constant sum.

Value

An object with the same format as X containing the perturbed compositions, except that vector X with matrix or data.frame Y returns the same rectangular format as Y.

Examples

```
x <- c(a = 1, b = 2, c = 3)
y <- c(a = 1, b = 1, c = 2)
perturbation(x, y)

X <- rbind(
  c(1, 2, 3),
  c(4, 5, 6)
)
perturbation(X, c(1, 1, 2))
perturbation(c(1, 1, 2), X)
```

petrafm

Calc-alkaline and tholeiitic volcanic rocks

Description

The ‘petrafm’ data set contains 100 classified volcanic rock samples from Ontario (Canada). The three-part composition is

$$[A : Na_2O + K_2O; F : FeO + 0.8998 Fe_2O_3; M : MgO]$$

Rocks from the calc-alkaline magma series (25 samples) can be distinguished from those of the tholeiitic magma series (75 samples) using an AFM diagram.

Usage

```
petrafm
```

Format

An object of class `data.frame` with 100 rows and 4 columns.

plot_balance

Plot a balance with node labels under horizontal branches

Description

Plot a balance with node labels under horizontal branches

Usage

```
plot_balance(
  B,
  data = NULL,
  main = "Balance dendrogram",
  summary_fun = NULL,
  cex_node = 0.9,
  offset_node = 0.05,
  ...
)
```

Arguments

B	Balance basis matrix
data	Optional compositional data used to compute balance summaries
main	Plot title
summary_fun	Optional function applied to each balance coordinate vector. It must take a numeric vector and return a character string.
cex_node	Character expansion for node labels
offset_node	Vertical offset below the horizontal branch, relative to max height
...	Further arguments passed to plot

Value

Invisibly returns a data.frame with node coordinates and labels

Examples

```
X = waste[,5:9]
B = pb_basis(X, method = 'exact')

plot_balance(B)

plot_balance(B, data = X,
  summary_fun = function(x){
    q = quantile(x, probs = c(0.25, 0.5, 0.75))
    sprintf("%0.2f [%0.2f-%0.2f]", q[2], q[1], q[3])
  })
```

pollen	<i>Pollen composition in fossils</i>
--------	--------------------------------------

Description

The ‘pollen’ data set contains 30 fossil pollen samples from three different locations (recorded in variable ‘group’). The measured composition is the three-part composition [*pinus, abies, quercus*].

Usage

pollen

Format

An object of class `data.frame` with 30 rows and 4 columns.

pottery	<i>Chemical compositions of Romano-British pottery</i>
---------	--

Description

The ‘pottery’ data set contains the chemical composition of 45 specimens of Romano-British pottery. The measurements were obtained by atomic absorption spectrophotometry and include nine oxides: Al₂O₃, Fe₂O₃, MgO, CaO, Na₂O, K₂O, TiO₂, MnO, and BaO.

The specimens come from five different kiln sites.

Usage

pottery

Format

An object of class `data.frame` with 45 rows and 11 columns.

`powering`*Powering of compositional data*

Description

The powering operation raises each part of a composition to a scalar exponent and then applies closure to re-normalize the result as a composition.

Usage

```
powering(X, alpha)
```

Arguments

<code>X</code>	A numeric vector, matrix or data.frame containing compositions.
<code>alpha</code>	A numeric scalar or vector. If <code>X</code> is a matrix or data.frame, <code>alpha</code> may have length 1, length equal to the number of rows of <code>X</code> , or length equal to the number of parts of <code>X</code> . If <code>X</code> is a vector and <code>alpha</code> has length greater than 1, one powered composition is returned for each value of <code>alpha</code> .

Details

Powering is the analogue of scalar multiplication in the simplex. Each part is raised to `alpha`, and the result is closed with `closure`. When `alpha` has one value per row, each composition is powered by its corresponding value. When it has one value per part, each part receives its corresponding exponent. For vector `X` and vector `alpha`, each row of the result is `X` powered by the corresponding element of `alpha`.

Value

An object with the same format as `X` containing the powered compositions, except that vector `X` with vector `alpha` returns a matrix.

Examples

```
x <- c(a = 1, b = 2, c = 3)
powering(x, 2)
powering(x, c(1, 2))

X <- rbind(
  c(1, 2, 3),
  c(4, 5, 6)
)
powering(X, c(1, 2))
```

`random_composition_with_fixed_pb`*Generate a random composition with a prescribed first principal balance*

Description

Simulates a random composition whose coordinate system is constructed from a sequential binary partition induced by a given first balance. The supplied balance is completed to a full orthonormal basis using `sbp_basis` with `fill = TRUE`.

Usage

```
random_composition_with_fixed_pb(principal_balance, n = 100, sd1 = 5)
```

Arguments

<code>principal_balance</code>	An integer or numeric vector in $\{-1, 0, 1\}$ defining the first balance of the SBP.
<code>n</code>	Integer. Number of observations to generate.
<code>sd1</code>	Numeric value used to scale the first latent coordinate before rotating the simulated coordinates.

Details

Standard normal latent coordinates are first generated in dimension $D - 1$, where D is the number of parts. Their sample covariance matrix is then diagonalized, and the associated eigenvectors are used to rotate the latent coordinates before mapping them back to the simplex using the basis induced by `principal_balance`.

This function is mainly intended for examples, simulation studies, and experiments where a specific first balance structure is desired.

Value

A composition matrix with `n` rows and `length(principal_balance)` columns.

See Also

[sbp_basis](#), [composition](#)

read_cdp	<i>Import data from a codapack workspace</i>
----------	--

Description

Import data from a codapack workspace

Usage

```
read_cdp(fname)
```

Arguments

fname	cdp file name
-------	---------------

sbp_basis	<i>Basis from a sequential binary partition</i>
-----------	---

Description

Construct a balance basis from a sequential binary partition (SBP) or from a more general collection of balances.

Usage

```
sbp_basis(sbp, data = NULL, fill = FALSE, silent = FALSE)
```

Arguments

sbp	A list of formulas or a matrix describing balances.
data	Optional compositional data set used to extract part names when 'sbp' is given as a list of formulas.
fill	Logical; if 'TRUE', complete the supplied balances to obtain a full basis.
silent	Logical; if 'FALSE', report whether the resulting balances form a basis, and whether they are orthogonal or orthonormal.

Details

The argument 'sbp' can be specified in two ways:

- as a list of formulas, where each formula defines the numerator and the denominator groups of a balance,
- as a matrix with one column per balance and one row per part. Positive entries indicate parts in the numerator, negative entries indicate parts in the denominator, and zeros indicate unused parts.

Value

A matrix whose columns are balances.

Examples

```
X <- data.frame(
  a = 1:2, b = 2:3, c = 4:5,
  d = 5:6, e = 10:11, f = 100:101, g = 1:2
)

# Sequential SBP construction
sbp_basis(list(
  b1 = a ~ b + c + d + e + f + g,
  b2 = b ~ c + d + e + f + g,
  b3 = c ~ d + e + f + g,
  b4 = d ~ e + f + g,
  b5 = e ~ f + g,
  b6 = f ~ g
), data = X)

# Chain construction
sbp_basis(list(
  b1 = a ~ b,
  b2 = b1 ~ c,
  b3 = b2 ~ d,
  b4 = b3 ~ e,
  b5 = b4 ~ f,
  b6 = b5 ~ g
), data = X)

# Non-orthogonal system of balances
sbp_basis(list(
  b1 = a + b + c ~ e + f + g,
  b2 = d ~ a + b + c,
  b3 = d ~ e + g,
  b4 = a ~ e + b,
  b5 = b ~ f,
  b6 = c ~ g
), data = X)

# Direct construction from a contrast matrix
sbp_basis(cbind(
  c( 1,  1, -1, -1),
  c( 1, -1,  1, -1),
  c( 1, -1, -1,  1)
))
```

Description

The ‘serprot’ data set records the percentages of four serum proteins from blood samples of 30 patients. Fourteen patients have one disease and sixteen have another.

The four-part compositions are formed by [*albumin, pre-albumin, globulin A, globulin B*].

Usage

```
serprot
```

Format

An object of class `data.frame` with 36 rows and 7 columns.

```
statistician_time      A statistician's time budget
```

Description

The ‘statistician_time’ data set records the daily time budget of an academic statistician across 20 working days. The six activities are teaching (‘T’), consultation (‘C’), administration (‘A’), research (‘R’), other wakeful activities (‘O’), and sleep (‘S’).

These activities may also be grouped into work (‘T’, ‘C’, ‘A’, ‘R’) and leisure (‘O’, ‘S’). The data allow investigation of the relationship between detailed time-allocation patterns and the broader division between work and leisure.

Usage

```
statistician_time
```

Format

An object of class `data.frame` with 20 rows and 7 columns.

```
variation_array      Variation array is returned.
```

Description

Variation array is returned.

Usage

```
variation_array(X, include_means = FALSE, ml_covariance = FALSE)
```

Arguments

<code>X</code>	Compositional dataset
<code>include_means</code>	if TRUE logratio means are included in the lower-left triangle
<code>ml_covariance</code>	if TRUE Maximum-likelihood estimation of the covariance for the multivariate normal distribution is used (dividing the scatter matrix by n instead of $n-1$)

Value

variation array matrix

Examples

```
set.seed(1)
X = matrix(exp(rnorm(5*100)), nrow=100, ncol=5)
variation_array(X)
variation_array(X, include_means = TRUE)
```

waste

Urban waste composition in Catalonia

Description

The ‘waste‘ data set studies the relationship between waste composition and floating population in Catalonia. The actual population of a municipality combines census population and floating population (tourists, seasonal visitors, temporary workers, and similar short-term residents), expressed as equivalent full-time residents.

The composition of urban solid waste is classified into five parts:

- ‘x1’: non-recyclable waste,
- ‘x2’: glass,
- ‘x3’: light containers,
- ‘x4’: paper and cardboard,
- ‘x5’: biodegradable waste.

Waste generation and composition are influenced by floating population, which makes waste composition a useful predictor of this difficult-to-measure demographic quantity.

Usage

waste

Format

An object of class `data.frame` with 215 rows and 10 columns.

References

Coenders, G., Martín-Fernández, J. A., & Ferrer-Rosell, B. (2017). *When relative and absolute information matter: compositional predictor with a total in generalized linear models*. *Statistical Modelling*, **17**(6), 494–512.

weibo_hotels

Hotel posts in social media

Description

The ‘weibo_hotels’ data set compares the use of Weibo (the Chinese equivalent of Facebook) in hospitality e-marketing between small and medium establishments and larger hotel businesses in China.

The 50 latest posts from the Weibo page of each hotel ($n = 10$) were content-analysed and coded into a four-part composition: [*facilities, food, events, promotions*]. Hotels were also classified by size as large (‘L’) or small (‘S’).

Usage

weibo_hotels

Format

An object of class `data.frame` with 10 rows and 5 columns.

Index

* datasets

- alimentation, 3
 - arctic_lake, 4
 - blood_mn, 5
 - bmi_activity, 5
 - eurostat_employment, 17
 - foraminiferals, 18
 - house_expend, 21
 - household_budget, 20
 - kilauea_iki, 22
 - mammals_milk, 23
 - milk_cows, 23
 - montana, 24
 - parliament2017, 25
 - petrafm, 33
 - pollen, 35
 - pottery, 35
 - serprot, 39
 - statistician_time, 40
 - waste, 41
 - weibo_hotels, 42
- alimentation, 3
- alr_basis, 3, 12, 15
- alr_c (coordinates), 14
- arctic_lake, 4
- as.dist, 16
- blood_mn, 5
- bmi_activity, 5
- cc_basis, 6
- cdp_basis, 6
- cdp_partition, 7
- center, 7, 20
- closure, 8, 32, 36
- clr_basis, 9, 12, 15
- clr_c (coordinates), 14
- coda.base, 10
- coda.base-package (coda.base), 10
- coda_replacement, 10
- comp (composition), 11
- composition, 11, 15, 37
- conditional_obasis, 12
- coord (coordinates), 14
- coordinates, 12, 14
- dist, 15, 15, 16
- dist_coda, 15, 16
- eurostat_employment, 17
- foraminiferals, 18
- gen_coda_with_zeros_and_missings, 18
- gmean, 19
- hclust, 29
- house_expend, 21
- household_budget, 20
- ilr_basis, 12, 15, 21
- ilr_c (coordinates), 14
- kilauea_iki, 22
- mammals_milk, 23
- milk_cows, 23
- montana, 24
- olr_basis (ilr_basis), 21
- olr_c (coordinates), 14
- pairwise_basis, 24
- parliament2017, 25
- partial_pb_constrained, 25
- partial_pb_exact, 26
- partial_pb_tabu_search, 27, 31
- pb_basis, 29
- pb_tabu_search, 30
- pc_basis, 31

perturbation, [32](#)
petrafm, [33](#)
plot_balance, [33](#)
pollen, [35](#)
pottery, [35](#)
powering, [36](#)

random_composition_with_fixed_pb, [37](#)
read_cdp, [38](#)

sbp_basis, [12](#), [15](#), [31](#), [37](#), [38](#)
serprot, [39](#)
statistician_time, [40](#)

variation_array, [40](#)

waste, [41](#)
weibo_hotels, [42](#)