Package ‘coda.base’

July 18, 2022

Type Package

Title A Basic Set of Functions for Compositional Data Analysis

Version 0.5.2

Date 2022-07-06

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


Depends R (>= 3.5)

Imports Rcpp (>= 0.12.12), stats, Matrix, jsonlite

LinkingTo Rcpp, RcppArmadillo (>= 0.11.2)

License GPL

Encoding UTF-8

LazyData true

NeedsCompilation yes

RoxygenNote 7.2.0

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

VignetteBuilder knitr

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

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

Repository CRAN

Date/Publication 2022-07-18 09:30:02 UTC

R topics documented:

  alr_basis .......................................................... 2
  arctic_lake .......................................................... 3
  basis .............................................................. 3
Additive log-ratio basis

Compute the transformation matrix to express a composition using the oblique additive log-ratio coordinates.

Usage

```r
alr_basis(dim, denominator = dim, numerator = which(denominator != 1:dim))
```

Arguments

- `dim`: number of parts
- `denominator`: part used as denominator (default behaviour is to use last part)
- `numerator`: parts to be used as numerator. By default all except the denominator parts are chosen following original order.

Value

matrix
References


Examples

```r
alr_basis(5)
# Third part is used as denominator
alr_basis(5, 3)
# Third part is used as denominator, and
# other parts are rearranged
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 [sand, silt, clay] compositions of 39 sediment

**Usage**

```r
arctic_lake
```

**Format**

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

---

**basis**

*Coordinates basis*

**Description**

Obtain coordinates basis

**Usage**

```r
basis(H)
```

**Arguments**

- `H` coordinates for which basis should be shown

**Value**

basis used to create coordinates H
cbalance_approx | Balance generated from the first canonical correlation component

**Description**

Balance generated from the first canonical correlation component

**Usage**

cbalance_approx(Y, X)

**Arguments**

Y  compositional dataset
X  explanatory dataset

**Value**

matrix

cc_basis | Isometric log-ratio basis based on canonical correlations

**Description**

Isometric log-ratio basis based on canonical correlations

**Usage**

cc_basis(Y, X)

**Arguments**

Y  compositional dataset
X  explanatory dataset

**Value**

matrix
**cdp_basis**

Isometric log-ratio basis based on Balances.

**Description**

The function return default balances used in CoDaPack software.

**Usage**

```R
cdp_basis(dim)
```

**Arguments**

- `dim` dimension to build the ILR basis based on balanced balances

**Value**

matrix

---

**cdp_partition**

CoDaPack’s default binary partition

**Description**

Compute the default binary partition used in CoDaPack’s software

**Usage**

```R
cdp_partition(ncomp)
```

**Arguments**

- `ncomp` number of parts

**Value**

matrix

**Examples**

```R
cdp_partition(4)
```
**center**  
*Dataset center*

**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**

```r
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)))
```

---

**clr_basis**  
*Centered log-ratio basis*

**Description**
Compute the transformation matrix to express a composition using the linearly dependant centered log-ratio coordinates.

**Usage**
`clr_basis(dim)`

**Arguments**
- `dim`: number of parts

**Value**
matrix
References

Examples

```r
(B <- clr_basis(5))
# CLR coordinates are linearly dependant coordinates.
(clr_coordinates <- coordinates(c(1,2,3,4,5), B))
# The sum of all coordinates equal to zero
sum(clr_coordinates) < 1e-15
```

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

Author(s)
Marc Comas-Cufí

Composition

Get composition from coordinates w.r.t. an specific basis

Description
Calculate a composition from coordinates with respect a given basis

Usage

```r
composition(H, basis = NULL)
comp(H, basis = NULL)
```

Arguments

- `H` coordinates of a composition. Either a matrix, a data.frame or a vector
- `basis` basis used to calculate the coordinates

Value
coordinates with respect the given basis
coordinates

See Also

See functions \texttt{ilr\_basis}, \texttt{alr\_basis}, \texttt{clr\_basis}, \texttt{sbp\_basis} to define different compositional basis. See function \texttt{coordinates} to obtain details on how to calculate coordinates of a given composition.

\begin{tabular}{ll}
\texttt{coordinates} & \textit{Get coordinates from compositions w.r.t. an specific basis} \\
\end{tabular}

Description

Calculate the coordinates of a composition with respect a given basis

Usage

\begin{verbatim}
coordinates(X, basis = "ilr", basis_return = TRUE)
coord(..., basis = "ilr")
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{X} \hspace{1cm} compositional dataset. Either a matrix, a data.frame or a vector
\item \texttt{basis} \hspace{1cm} basis used to calculate the coordinates. \texttt{basis} can be either a string or a matrix. Accepted values for strings are: 'ilr' (default), 'clr', 'alr', 'pw', 'pc', 'pb' and 'cdp'. If \texttt{basis} is a matrix, it is expected to have log-ratio basis given in columns.
\item \texttt{basis\_return} \hspace{1cm} Should the basis be returned as attribute? (default: \texttt{TRUE})
\item \texttt{...} \hspace{1cm} components of the compositional data
\end{itemize}

Details

\texttt{coordinates} function calculates the coordinates of a composition w.r.t. a given basis. \texttt{basis} parameter is used to set the basis, it can be either a matrix defining the log-contrasts in columns or a string defining some well-known log-contrast: 'ilr' 'clr', 'alr', 'pw', 'pc', 'pb' and 'cdp', for the additive log-ratio, centered log-ratio, isometric log-ratio, pairwise log-ratio, clr principal components, clr principal balances or default's CoDaPack balances respectively.

Value

Coordinates of composition \texttt{X} with respect the given basis.

See Also

See functions \texttt{ilr\_basis}, \texttt{alr\_basis}, \texttt{clr\_basis}, \texttt{sbp\_basis} to define different compositional basis. See function \texttt{composition} to obtain details on how to calculate a compositions from given coordinates.
Examples

coordinates(c(1,2,3,4,5))
h = coordinates(c(1,2,3,4,5))
basis(h)
# basis is shown if 'coda.base.basis' option is set to TRUE
options('coda.base.basis' = TRUE)
coordinates(c(1,2,3,4,5))
# Default transformation can improve performance.
N = 100
K = 1000
X = matrix(exp(rnorm(N*K)), nrow=N, ncol=K)
system.time(coordinates(X, alr_basis(K)))
system.time(coordinates(X, 'alr'))

---

dist Distance Matrix Computation (including Aitchison distance)

Description

This function overwrites dist function to contain Aitchison distance between compositions.

Usage

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

Arguments

x compositions method
method the distance measure to be used. This must be one of "aitchison", "euclidean",
"maximum", "manhattan", "canberra", "binary" or "minkowski". Any unambiguos substring can be given.
...
arguments passed to dist function

Value

dist returns an object of class "dist".

See Also

See functions dist.
Examples

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

(d <- dist(X, method = 'aitchison'))
plot(hclust(d))

# In contrast to Euclidean distance
dist(rbind(c(1,1,1), c(100, 100, 100)), method = 'euc') # method = 'euclidean'
# using Aitchison distance, only relative information is of importance
dist(rbind(c(1,1,1), c(100, 100, 100)), method = 'ait') # method = 'aitchison'

---

gmean

Geometric Mean

Description

Generic function for the (trimmed) geometric mean.

Usage

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

Arguments

x
A nonnegative vector.

zero.rm
a logical value indicating whether zero values should be stripped before the computation proceeds.

trim
the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

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

See Also
center
**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 over a period of one month their expenditures on 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.

---

**ilr_basis**

*Isometric log-ratio basis for log-transformed compositions.*

**Description**

By default the basis of the clr-given by Egozcue et al., 2013 Build an isometric log-ratio basis for a composition with k+1 parts

\[
h_i = \frac{1}{i+1} \log \sqrt{\prod_{j=1}^{i} x_j} x_{i+1}
\]

for \( i \in 1 \ldots k \).

**Usage**

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

**Arguments**

- `dim` number of components
- `type` if different than `pivot` (pivot balances) or `cdp` (codapack balances) default balances are returned, which computes a triangular Helmert matrix as defined by Egozcue et al., 2013.

**Details**

Modifying parameter type (pivot or cdp) other ilr basis can be generated
References


Examples

ilr_basis(5)

Description

The function returns all combinations of pairs of log-ratios.

Usage

pairwise_basis(dim)

Arguments

dim dimension to build the pairwise log-ratio generator system

Value

matrix

parliament2017

Results of catalan parliament elections in 2017 by regions.

Description

Results of catalan parliament elections in 2017 by regions.

Usage

parliament2017
Format

A data frame with 42 rows and 9 variables:

- **com**: Region
- **cs**: Votes to Ciutadans party
- **jxcat**: Votes to Junts per Catalunya party
- **erc**: Votes to Esquerra republicana de Catalunya party
- **psc**: Votes to Partit socialista de Catalunya party
- **catsp**: Votes to Catalunya si que es pot party
- **cup**: Votes to Candidatura d’unitat popular party
- **pp**: Votes to Partit popular party
- **other**: Votes to other parties

Source

https://www.idescat.cat/tema/elecc

---

pb_basis

Isometric log-ratio basis based on Principal Balances.

Description

Exact method to calculate the principal balances of a compositional dataset. Different methods to approximate the principal balances of a compositional dataset are also included.

Usage

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

Arguments

- **X**: compositional dataset
- **method**: method to be used with Principal Balances. Methods available are: ’exact’, ’constrained’ or ’cluster’.
When searching up, should the algorithm try to find possible siblings for the current balance (TRUE) or build a parent directly forcing current balance to be part of the next balance (default: FALSE). While the first is more exhaustive and given better results the second is faster and can be used with high dimensional datasets.

Method to be used with the hclust function (default: ‘ward.D2’) or any other method available in hclust function

should the principal balances found be returned ordered? (first column, first principal balance and so on)

parameters passed to hclust function

matrix


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

# Optimal variance obtained with Principal components
(v1 <- apply(coordinates(X, 'pc'), 2, var))
# Optimal variance obtained with Principal balances
(v2 <- apply(coordinates(X,pb_basis(X, method='exact')), 2, var))
# Solution obtained using constrained method
(v3 <- apply(coordinates(X,pb_basis(X, method='constrained')), 2, var))
# Solution obtained using Ward method
(v4 <- apply(coordinates(X,pb_basis(X, method='cluster')), 2, var))

# Plotting the variances
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')

Isometric log-ratio basis based on Principal Components.

Different approximations to approximate the principal balances of a compositional dataset.
Usage

\texttt{pc\_basis(X)}

Arguments

\texttt{X} \hspace{1cm} \text{compositional dataset}

Value

\text{matrix}

---

\texttt{plot\_balance} \hspace{1cm} \textit{Plot a balance}

Description

Plot a balance

Usage

\texttt{plot\_balance(B, data = NULL, main = "Balance dendrogram", \ldots)}

Arguments

\texttt{B} \hspace{1cm} \text{Balance to plot}
\texttt{data} \hspace{1cm} \text{(Optional) Data used to calculate the statistics associated to a balance}
\texttt{main} \hspace{1cm} \text{Plot title}
\ldots \hspace{1cm} \text{further arguments passed to plot}

Value

\text{Balance plot}
Description

The pottery data set consists of data pertaining to the chemical composition of 45 specimens of Romano-British pottery. The method used to generate these data is atomic absorption spectrophotometry, and readings for nine oxides (Al₂O₃, Fe₂O₃, MgO, CaO, Na₂O, K₂O, TiO₂, MnO, BaO) are provided. These samples come from five different kiln sites.

Usage

pottery

Format

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

print.coda

Printing coordinates

Description

The function hides the basis attribute. An option is included to show such basis.

Usage

```r
## S3 method for class 'coda'
print(x, ..., basis = getOption("coda.base.basis"))
```

Arguments

- `x`: coordinates
- `...`: parameters passed to print function
- `basis`: boolean to show or not the basis with the output
**Description**

Import data from a codapack workspace

**Usage**

```
read_cdp(fname)
```

**Arguments**

- `fname`: cdp file name

---

**sbp_basis**

Isometric log-ratio basis based on Balances Build an ilr_basis using a sequential binary partition or a generic coordinate system based on balances.

**Description**

Isometric log-ratio basis based on Balances Build an **ilr_basis** using a sequential binary partition or a generic coordinate system based on balances.

**Usage**

```
sbp_basis(..., data = NULL, silent = F)
```

**Arguments**

- `...`: balances to consider
- `data`: composition from where name parts are extracted
- `silent`: inform about orthogonality

**Value**

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

# A non-orthogonal basis can also be calculated.
sbp_basis(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)

variation_array

Variation array is returned.

Description

Variation array is returned.

Usage

variation_array(X, only_variation = FALSE)

Arguments

X Compositional dataset

only_variation if TRUE only the variation matrix is calculated

Value

variation array matrix

Examples

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

* datasets
  arctic_lake, 3
  household_budget, 11
  parliament2017, 12
  pottery, 16
  alr_basis, 2, 8
  arctic_lake, 3
  basis, 3
  cbalance_approx, 4
  cc_basis, 4
  cdp_basis, 5
  cdp_partition, 5
  center, 6, 10
  clr_basis, 6, 8
  coda.base, 7
  comp (composition), 7
  composition, 7, 8
  coord (coordinates), 8
  coordinates, 8, 8
  dist, 9, 9
  gmean, 10
  household_budget, 11
  ilr_basis, 8, 11, 17
  pairwise_basis, 12
  parliament2017, 12
  pb_basis, 13
  pc_basis, 14
  plot_balance, 15
  pottery, 16
  print.coda, 16
  read_cdp, 17
  sbp_basis, 8, 17
  variation_array, 18