Brice Lebrun [ctb] (<https://orcid.org/0000-0001-7503-8685>), Université Bordeaux Montaigne, Université Bordeaux Montaigne [fnd], CNRS [fnd]

Repository CRAN

Date/Publication 2023-11-29 18:40:02 UTC

### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregate</td>
<td>3</td>
</tr>
<tr>
<td>arctic</td>
<td>4</td>
</tr>
<tr>
<td>arithmetic</td>
<td>5</td>
</tr>
<tr>
<td>as_amounts</td>
<td>6</td>
</tr>
<tr>
<td>as_composition</td>
<td>7</td>
</tr>
<tr>
<td>as_features</td>
<td>9</td>
</tr>
<tr>
<td>as_graph</td>
<td>10</td>
</tr>
<tr>
<td>barplot</td>
<td>11</td>
</tr>
<tr>
<td>chemistry</td>
<td>13</td>
</tr>
<tr>
<td>closure</td>
<td>13</td>
</tr>
<tr>
<td>covariance</td>
<td>14</td>
</tr>
<tr>
<td>dist</td>
<td>16</td>
</tr>
<tr>
<td>groups</td>
<td>17</td>
</tr>
<tr>
<td>hongite</td>
<td>19</td>
</tr>
<tr>
<td>identifiers</td>
<td>20</td>
</tr>
<tr>
<td>mahalanobis</td>
<td>21</td>
</tr>
<tr>
<td>margin</td>
<td>22</td>
</tr>
<tr>
<td>mean</td>
<td>23</td>
</tr>
<tr>
<td>metric_var</td>
<td>24</td>
</tr>
<tr>
<td>missing</td>
<td>26</td>
</tr>
<tr>
<td>mix</td>
<td>27</td>
</tr>
<tr>
<td>outliers</td>
<td>28</td>
</tr>
<tr>
<td>pca_coda</td>
<td>30</td>
</tr>
<tr>
<td>perturbation</td>
<td>32</td>
</tr>
<tr>
<td>petrography</td>
<td>33</td>
</tr>
<tr>
<td>plot</td>
<td>35</td>
</tr>
<tr>
<td>plot_logratio</td>
<td>36</td>
</tr>
<tr>
<td>plot_outliers</td>
<td>37</td>
</tr>
<tr>
<td>policy</td>
<td>39</td>
</tr>
<tr>
<td>powering</td>
<td>40</td>
</tr>
<tr>
<td>samples</td>
<td>41</td>
</tr>
<tr>
<td>scalar</td>
<td>43</td>
</tr>
<tr>
<td>scale</td>
<td>44</td>
</tr>
<tr>
<td>slides</td>
<td>45</td>
</tr>
<tr>
<td>subset</td>
<td>46</td>
</tr>
<tr>
<td>totals</td>
<td>48</td>
</tr>
<tr>
<td>transform_alr</td>
<td>49</td>
</tr>
<tr>
<td>transform_clr</td>
<td>50</td>
</tr>
</tbody>
</table>
aggregate

Description

Splits the data into subsets, computes summary statistics for each, and returns the result.

Usage

## S4 method for signature 'CompositionMatrix'
aggregate(x, by, FUN, ...)

Arguments

x
A CompositionMatrix object.

by
A character string specifying the grouping element. It must be one of "samples" or "groups". Any unambiguous substring can be given.

FUN
A function to compute the summary statistics.

... Further arguments to be passed to FUN.

Value

A matrix.

Author(s)

N. Frerebeau

See Also

Other statistics: covariance(), dist.mahalanobis(), margin(), mean(), metric.var(), scale(), variation()
Examples

```r
## Create a data.frame
X <- data.frame(
    groups = c("X", "X", "X", "Y", "Y", "Y", "Y", "Y", "Y"),
    Ca = c(7.72, 7.32, 3.11, 7.19, 7.41, 5, 4.18, 1, 4.51),
    Fe = c(6.12, 5.88, 5.12, 6.18, 6.02, 7.14, 5.25, 5.28, 5.72),
    Na = c(0.97, 1.59, 1.25, 0.86, 0.76, 0.76, 0.51, 0.75, 0.52, 0.56)
)

## Coerce to a compositional matrix
Y <- as_composition(X)

## Compositional mean by sample
aggregate(Y, by = "samples", FUN = mean)

## Metric variance by group
aggregate(Y, by = "groups", FUN = metric_var)
```

---

**arctic**

*Arctic Lake*

**Description**

Sand, silt, clay compositions of 39 sediment samples at different water depths in an Arctic lake.

**Usage**

`arctic`

**Format**

A *data.frame* with 4 variables:

- **sand**  Sand content (percent).
- **silt**  Silt content (percent).
- **clay**  Clay content (percent).
- **depth** Water depth (m).

**References**


**See Also**

Other datasets: `chemistry`, `hongite`, `petrography`, `slides`
Description

Operators performing operations in the simplex.

Usage

x %perturbe% y

x %power% y

## S4 method for signature 'CompositionMatrix,CompositionMatrix'
x %perturbe% y

## S4 method for signature 'CompositionMatrix,numeric'
x %power% y

## S4 method for signature 'numeric,CompositionMatrix'
x %power% y

Arguments

x A CompositionMatrix object.
y A CompositionMatrix object or a numeric vector.

Details

%perturbe% Perturbation operation.
%power% Powering operation.

Value

A CompositionMatrix object or a numeric vector (same as x).

Author(s)

N. Frerebeau

See Also

Other operations in the simplex: closure(), perturbation(), powering(), scalar()
Examples

```r
x <- as_composition(c(1, 2, 3))
y <- as_composition(c(1, 2, 1))

## Perturbation
perturbation(x, y)
x + y

## Powering
powering(y, 2)
y * 2

## Scalar product
scalar(x, y)
```

---

**as_amounts** *Coerce to Amounts*

**Description**

Coerce to Amounts

**Usage**

```r
as_amounts(from, ...)
```

## S4 method for signature 'CompositionMatrix'
```r
as_amounts(from)
```

**Arguments**

<table>
<thead>
<tr>
<th>from</th>
<th>A <code>CompositionMatrix</code> object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>Currently not used.</td>
</tr>
</tbody>
</table>

**Value**

A numeric matrix.

**Author(s)**

N. Frerebeau

**See Also**

Other compositional data tools: `as_composition()`, `as_features()`
Examples

```r
## Create a count matrix
A1 <- matrix(data = as.numeric(sample(1:100, 100, TRUE)), nrow = 20)

## Coerce to compositions
B <- as_composition(A1)

## Row sums are internally stored before coercing to relative frequencies
get_totals(B)

## This allows to restore the source data
A2 <- as_amounts(B)

## Coerce to a data.frame
X <- data.frame(B)
head(X)
```

---

**as_composition**

Coerce to a Closed Compositional Matrix

**Description**

Coerces an object to a `CompositionMatrix` object.

**Usage**

```r
as_composition(from, ...)
```

```r
## S4 method for signature 'numeric'
as_composition(from)
```

```r
## S4 method for signature 'matrix'
as_composition(from)
```

```r
## S4 method for signature 'data.frame'
as_composition(
  from,
  codes = NULL,
  samples = NULL,
  groups = NULL,
  auto = getOption("nexus.autodetect"),
  verbose = getOption("nexus.verbose")
)
```

**Arguments**

- `from` A `matrix` or `data.frame` to be coerced.
- `...` Currently not used.
### as_composition

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>codes</td>
<td>An integer giving the index of the column to be used as laboratory codes (unique identifiers).</td>
</tr>
<tr>
<td>samples</td>
<td>An integer giving the index of the column to be used for sample identification: allows duplicates to identify replicated measurements. If NULL (the default), row names will be used as sample IDs.</td>
</tr>
<tr>
<td>groups</td>
<td>An integer giving the index of the column to be used to group the samples. If NULL (the default), no grouping is stored.</td>
</tr>
<tr>
<td>auto</td>
<td>A logical scalar: try to automatically detect codes, samples and groups columns?</td>
</tr>
<tr>
<td>verbose</td>
<td>A logical scalar: should R report extra information on progress?</td>
</tr>
</tbody>
</table>

### Details

The `CompositionMatrix` class has special slots:

- codes for laboratory codes,
- samples for repeated measurements/observation,
- groups to group data by site/area.

When coercing a data.frame to a `CompositionMatrix` object, an attempt is made to automatically assign values to these slots by mapping column names (case insensitive, plural insensitive). This behavior can be disabled by setting `options(nexus.autodetect = FALSE)` or overridden by explicitly specifying the columns to be used.

See vignette("nexus").

### Value

A `CompositionMatrix` object.

### Note

All non-numeric variable will be removed.

### Author(s)

N. Frerebeau

### See Also

Other compositional data tools: `as_amounts()`, `as_features()`

### Examples

```r
## Create a count matrix
A1 <- matrix(data = as.numeric(sample(1:100, 100, TRUE)), nrow = 20)

## Coerce to compositions
B <- as_composition(A1)
```
## Row sums are internally stored before coercing to relative frequencies
get_totals(B)

## This allows to restore the source data
A2 <- as_amounts(B)

## Coerce to a data.frame
X <- data.frame(B)
head(X)

---

**as_features**  
*Coerce to Features*

### Description

Converts an object to a collection of features.

### Usage

```r
as_features(from, ...)
```

### Arguments

- **from**  
  A `CompositionMatrix` object.

- **...**  
  Currently not used.

### Value

A `data.frame` with all informations as extra columns.

### Author(s)

N. Frerebeau

### See Also

Other compositional data tools: `as_amounts()`, `as_composition()`

### Examples

```r
## Create a count matrix
A1 <- matrix(data = as.numeric(sample(1:100, 100, TRUE)), nrow = 20)

## Coerce to compositions
B <- as_composition(A1)
```
as_graph

Graph of Log-ratios

Description

Produces a graph of log-ratios.

Usage

as_graph(object, ...)

## S4 method for signature 'LR'
as_graph(object)

## S4 method for signature 'ALR'
as_graph(object)

## S4 method for signature 'ILR'
as_graph(object)

Arguments

object A LogRatio object.

... Currently not used.

Value

An igraph graph object.

Author(s)

N. Frerebeau

See Also

Other plot methods: barplot(), plot_logratio, plot()
Examples

library(igraph)

## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Pairwise log-ratio
lr <- transform_lr(coda)
lr_graph <- as_graph(lr)
plot(lr_graph)

## Additive log-ratio
alr <- transform_alr(coda)
alr_graph <- as_graph(alr)
plot(alr_graph)

## Isometric log-ratio
ilr <- transform_ilr(coda)
ilr_graph <- as_graph(ilr)
plot(ilr_graph)

plr <- transform_plr(coda)
plr_graph <- as_graph(plr)
plot(plr_graph)

barplot

Barplot of Compositional Data

Description

Displays a compositional bar chart.

Usage

## S4 method for signature 'CompositionMatrix'
barplot(
  height,
  ..., 
  order = NULL,
  decreasing = FALSE,
  groups = get_groups(height),
  horiz = TRUE,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  sub = NULL,
ann = graphics::par("ann"),
axes = TRUE
)

Arguments

height
A CompositionMatrix object.

... Further parameters to be passed to graphics::barplot().

order
An integer vector giving the index of the column to be used for the ordering of the data.

decreasing
A logical scalar: should the sort order be increasing or decreasing?

groups
A factor in the sense that as.factor(groups) defines the grouping. If set, a matrix of panels defined by groups will be drawn.

horiz
A logical scalar. If FALSE, the bars are drawn vertically with the first bar to the left. If TRUE (the default), the bars are drawn horizontally with the first at the bottom.

xlab, ylab
A character vector giving the x and y axis labels.

main
A character string giving a main title for the plot.

sub
A character string giving a subtitle for the plot.

ann
A logical scalar: should the default annotation (title and x and y axis labels) appear on the plot?

axes
A logical scalar: should axes be drawn on the plot?

Value

barplot() is called for its side-effects: is results in a graphic being displayed (invisibly return height).

Author(s)

N. Frerebeau

See Also

Other plot methods: as_graph(), plot_logratio, plot()

Examples

## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Bar plot
barplot(coda, order = 2)
Data from Day et al. 2011
data("kommos", package = "folio") # Coerce to compositional data
kommos <- remove_NA(kommos, margin = 1) # Remove cases with missing values
coda <- as_composition(kommos, groups = 1) # Use ceramic types for grouping

barplot(coda, order = 1, main = "Kommos Ceramics")
barplot(coda, order = 1, horiz = FALSE, main = "Kommos Ceramics")

## Data from Day et al. 2011

data("kommos", package = "folio") # Coerce to compositional data
kommos <- remove_NA(kommos, margin = 1) # Remove cases with missing values
coda <- as_composition(kommos, groups = 1) # Use ceramic types for grouping

barplot(coda, order = 1, main = "Kommos Ceramics")
barplot(coda, order = 1, horiz = FALSE, main = "Kommos Ceramics")

### chemistry

**Can Sora Chemical Data**

**Description**

Can Sora Chemical Data

**Usage**

chemistry

**Format**

A data.frame with 30 variables.

**References**


**See Also**

Other datasets: arctic, hongite, petrography, slides

### closure

**Closure Operation**

**Description**

Closes compositions to sum up to 1.

**Usage**

closure(x, ...)

## S4 method for signature 'numeric'
closure(x, total = 1, na.rm = FALSE)

## S4 method for signature 'matrix'
closure(x, total = 1, na.rm = FALSE)
covariance

Arguments

- `x`: A numeric vector or matrix.
- `...`: Currently not used.
- `total`: A numeric vector specifying the total amount to which the compositions should be closed (defaults to 1).
- `na.rm`: A logical scalar: should missing values be removed?

Value

A numeric vector or matrix (same as `x`).

Author(s)

N. Frerebeau

See Also

Other operations in the simplex: arithmetic, perturbation(), powering(), scalar()

Examples

```r
x <- as_composition(c(1, 2, 3))
y <- as_composition(c(1, 2, 1))

## Perturbation
perturbation(x, y)
x + y

## Powering
powering(y, 2)
y * 2

## Scalar product
scalar(x, y)
```

---

<table>
<thead>
<tr>
<th>covariance</th>
<th>Covariance Matrix</th>
</tr>
</thead>
</table>

Description

Computes the (centered) log-ratio covariance matrix (see below).
Usage

covariance(x, ...)

## S4 method for signature 'CompositionMatrix'
covariance(x, center = TRUE, method = "pearson")

## S4 method for signature 'ALR'
covariance(x, method = "pearson")

## S4 method for signature 'CLR'
covariance(x, method = "pearson")

Arguments

x A CompositionMatrix object.
...
center A logical scalar: should the centered log-ratio covariance matrix be computed?
method A character string indicating which covariance is to be computed (see stats::cov()).

Value

A matrix.

Methods (by class)

- covariance(ALR): Computes the log-ratio covariance matrix (Aitchison 1986, definition 4.5).
- covariance(CLR): Computes the centered log-ratio covariance matrix (Aitchison 1986, definition 4.6).

Author(s)

N. Frerebeau

References


See Also

Other statistics: aggregate(), dist, mahalanobis(), margin(), mean(), metric_var(), scale(), variation()
Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Log-ratio covariance matrix
## (Aitchison 1986, definition 4.5)
covariance(coda, center = FALSE)

## Centered log-ratio covariance matrix
## (Aitchison 1986, definition 4.6)
covariance(coda, center = TRUE)
```

Description

Computes the log-ratio variance matrix.

Usage

```
## S4 method for signature 'CompositionMatrix'
dist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
```

Arguments

- `x`: A `CompositionMatrix` object.
- `method`: A `character` string specifying the distance measure to be used. See `stats::dist()` for the available distances.
- `diag`: A `logical` scalar indicating whether the diagonal of the distance matrix should be printed.
- `upper`: A `logical` scalar indicating whether the upper triangle of the distance matrix should be printed.
- `p`: An `integer` giving the power of the Minkowski distance.

Details

Distances are computed on CLR-transformed data.

Value

A `stats::dist` object.
### groups

**Author(s)**

N. Frerebeau

**References**


**See Also**

`stats::dist()`

**Examples**

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Aitchison distance
## (euclidean distance between CLR-transformed compositions)
d <- dist(coda)

## Cluster dendrogram
h <- hclust(d, method = "ward.D2")
plot(h)
```

---

### Working With Groups

**Description**

Retrieves or defines the groups to which the observations belong.

**Usage**

- `any_assigned(x)`
- `is_assigned(x)`
- `get_groups(x)`
- `set_groups(x) <- value`
## S4 method for signature 'CompositionMatrix'
is_assigned(x)

## S4 method for signature 'LogRatio'
is_assigned(x)

## S4 method for signature 'OutlierIndex'
is_assigned(x)

## S4 method for signature 'CompositionMatrix'
any_assigned(x)

## S4 method for signature 'LogRatio'
any_assigned(x)

## S4 method for signature 'OutlierIndex'
any_assigned(x)

## S4 method for signature 'CompositionMatrix'
get_groups(x)

## S4 method for signature 'LogRatio'
get_groups(x)

## S4 method for signature 'OutlierIndex'
get_groups(x)

## S4 replacement method for signature 'CompositionMatrix'
set_groups(x) <- value

### Arguments

x    An object from which to get or set groups.

value   A possible value for the groups of x.

### Details

See vignette("nexus").

### Value

- set_groups() returns an object of the same sort as x with the new group names assigned.
- get_groups() returns a character vector giving the group names of x.
- any_assigned() returns a logical scalar specifying whether or not x has groups.
- is_assigned() returns a logical vector specifying whether or not an observation belongs to a group.
hongite

Author(s)
N. Frerebeau

See Also
Other mutators: identifiers, samples, subset(), totals

---

hongite Hongite Mineralogy

Description
Mineral compositions of 25 rock specimens of hongite type.

Usage
hongite

Format
A data.frame with 5 variables (minerals):

A Albite (percent).
B Blandite (percent).
C Cornite (percent).
D Daubite (percent).
E Endite (percent).

References

See Also
Other datasets: arctic, chemistry, petrography, slides
identifiers

Unique Identifiers

Description
Retrieves or defines the unique identifier (eg. laboratory codes) of each observation.

Usage
```r
get_identifiers(x)
set_identifiers(x) <- value
```
```r
## S4 method for signature 'CompositionMatrix'
get_identifiers(x)
```
```r
## S4 method for signature 'LogRatio'
get_identifiers(x)
```
```r
## S4 method for signature 'OutlierIndex'
get_identifiers(x)
```
```r
## S4 replacement method for signature 'CompositionMatrix'
set_identifiers(x) <- value
```

Arguments
- **x**: An object from which to get or set codes.
- **value**: A possible value for the codes of `x`.

Details
See vignette("nexus").

Value
- `set_identifiers()` returns an object of the same sort as `x` with the new identifiers assigned.
- `get_identifiers()` returns a `character` vector giving the unique identifiers of `x`.

Author(s)
N. Frerebeau

See Also
Other mutators: `groups`, `samples`, `subset()`, `totals`
### mahalanobis

**Mahalanobis Distance**

**Description**

Computes the squared Mahalanobis distance of all rows in \( x \).

**Usage**

```r
## S4 method for signature 'CompositionMatrix'
mahalanobis(x, center, cov, ..., robust = TRUE, method = c("mve", "mcd"))

## S4 method for signature 'ILR'
mahalanobis(x, center, cov, ..., robust = TRUE, method = c("mve", "mcd"))
```

**Arguments**

- **x**  
  A `CompositionMatrix` or an `ILR` object.
- **center**  
  A numeric vector giving the mean vector of the distribution. If missing, will be estimated from \( x \).
- **cov**  
  A numeric matrix giving the covariance of the distribution. If missing, will be estimated from \( x \).
- **...**  
  Extra parameters to be passed to `MASS::cov.rob()`. Only used if `robust` is `TRUE`.
- **robust**  
  A logical scalar: should robust location and scatter estimation be used?
- **method**  
  A character string specifying the method to be used. It must be one of "mve" (minimum volume ellipsoid) or "mcd" (minimum covariance determinant). Only used if `robust` is `TRUE`.

**Value**

A numeric vector.

**Author(s)**

N. Frerebeau

**See Also**

`stats::mahalanobis()`

Other statistics: `aggregate()`, `covariance()`, `dist.margin()`, `mean()`, `metric_var()`, `scale()`, `variation()`
Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Mahalanobis distance
mahalanobis(coda)
```

---

**margin**  
*Marginal Compositions*

### Description
Marginal Compositions

### Usage

```r
margin(x, ...)
```

```r
## S4 method for signature 'CompositionMatrix'
margin(x, parts = c(1, 2), name = "*")
```

### Arguments

- **x**  
  A *CompositionMatrix* object.

- **...**  
  Currently not used.

- **parts**  
  An integer or a character vector specifying the columns to be selected.

- **name**  
  A character string giving the name of the amalgamation column.

### Value

A *CompositionMatrix* object.

### Author(s)

N. Frerebeau

### See Also

Other statistics:  
- aggregate(), covariance(), dist, mahalanobis(), mean(), metric_var(), scale(), variation()
## Data from Aitchison 1986
```
data("hongite")
coda <- as_composition(hongite)
mar <- margin(coda, parts = c("B", "D"))
head(mar)
```

---

### mean

**Compositional Mean**

#### Description

Compositional Mean

#### Usage

```
mean(x, ..., na.rm = FALSE)
```

#### Arguments

- **x**: A `CompositionMatrix` object.
- **...**: Currently not used.
- **na.rm**: A `logical` scalar: should missing values be removed?

#### Details

Closed vector of the columns geometric means.

#### Value

A `numeric` vector.

#### Author(s)

N. Frerebeau

#### References

See Also

Other statistics: aggregate(), covariance(), dist.mahalanobis(), margin(), metric_var(), scale(), variation()

Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Mean
mean(coda)

## Metric variance
metric_var(coda)

## Metric standard deviation
metric_sd(coda)
```

---

**metric_var**

**Metric Variance and Standard Deviation**

**Description**

- `metric_var()` computes the metric variance (or total variance), i.e. a global measure of spread.
- `metric_sd()` computes the metric standard deviation.

**Usage**

```r
metric_var(x, ...)
magic_var(x)
magic_sd(x)
```

```r
## S4 method for signature 'CompositionMatrix'
magic_var(x)
magic_sd(x)
```

**Arguments**

- `x` A `CompositionMatrix` object.
- `...` Currently not used.
metric_var

Details

The metric variance is the average of the CLR variances.

Value

A numeric vector.

Author(s)

N. Frerebeau

References


See Also

Other statistics: aggregate(), covariance(), dist, mahalanobis(), margin(), mean(), scale(), variation()

Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Mean
mean(coda)

## Metric variance
metric_var(coda)

## Metric standard deviation
metric_sd(coda)
```
Missing Values Replacement

Description

Multiplicative replacement of missing values.

Usage

```r
## S4 method for signature 'CompositionMatrix'
replace_NA(x, value)
```

Arguments

- `x` A `CompositionMatrix` object.
- `value` A numeric vector giving the replacement values.

Value

An `CompositionMatrix` object, where all missing values have been replaced.

Author(s)

N. Frerebeau

References


See Also

Other imputation methods: `policy`, `zero`

Examples

```r
## Data from Martín-Fernández et al. 2003
X <- data.frame(
  X1 = c(0.0000, 0.1304, 0.1963),
  X2 = c(0.1250, 0.3151, NA),
  X3 = c(0.1237, NA, NA),
  X4 = c(0.7253, 0.2002, 0.0819),
  X5 = c(0.0260, 0.3543, 0.0114)
)

## Coerce to a compositional matrix
Y <- as_composition(X)
```
```
## Replace zeros
Z <- replace_NA(Y, value = 0.2)
Z
```

---

**mix**  
_Mixed-Mode Analysis_

### Description
Mixes chemical and petrographic matrices.

### Usage
```
mix(x, y, ...)
```

#### S4 method for signature 'matrix,matrix'
```
mix(x, y, lambda = 1, ...)
```

#### S4 method for signature 'dist,dist'
```
mix(x, y, mu = 0.5)
```

### Arguments
- `x`: A `matrix` of chemical compositional data or a dissimilarity matrix for these chemical compositional data.
- `y`: A `matrix` of coded mineralogical binary data or a dissimilarity matrix for these mineralogical data.
- `...`: Extra parameters to be passed to `cluster::daisy()`.
- `lambda`: A length-one `numeric` vector giving a weighting factor.
- `mu`: A length-one `numeric` vector that lies between 0 and 1 giving the mixing parameter.

### Value
A `stats::dist` object.

### Methods (by class)
- `mix(x = matrix, y = matrix)`: First approach of mixed-mode analysis.
- `mix(x = dist, y = dist)`: Second approach of mixed-mode analysis.

### Note
Experimental.
Author(s)
N. Frerebeau

References


Examples

```r
## Prepare chemical data
data("chemistry")
chem <- chemistry[-1, major]

## Prepare petrographic data
data("petrography")
petro <- petrography[-c(7, 8), -1]
petro <- cdt(petro) # Get the complete disjunctive table

## First approach
mix1 <- mix(as.matrix(chem), as.matrix(petro), lambda = 2)
mds1 <- stats::cmdscale(mix1) # Multi-Dimensional Scaling
plot(mds1)
```

outliers

### Outlier Detection

#### Description
Outlier Detection

#### Usage
`outliers(object, ...)`

```r
## S4 method for signature 'CompositionMatrix'
outliers(
  object,
```
```
...
  groups = get_groups(object),
  robust = TRUE,
  method = c("mve", "mcd"),
  quantile = 0.975
)
```

**Arguments**

- **object**: A `CompositionMatrix`.
- ... Extra parameters to be passed to `MASS::cov.rob()`. Only used if `robust` is `TRUE`.
- **groups**: A factor in the sense that `as.factor(groups)` defines the grouping. If set, XXX.
- **robust**: A logical scalar: should robust location and scatter estimation be used?
- **method**: A character string specifying the method to be used. It must be one of "mve" (minimum volume ellipsoid) or "mcd" (minimum covariance determinant). Only used if `robust` is `TRUE`.
- **quantile**: A length-one numeric vector giving the significance level. `quantile` is used as a cut-off value for outlier detection: observations with larger (squared) Mahalanobis distance are considered as potential outliers.

**Details**

An outlier can be defined as having a very large Mahalanobis distance from all observations. In this way, a certain proportion of the observations can be identified, e.g. the top 2% of values (i.e. values above the 0.98th percentile of the Chi-2 distribution).

On the one hand, the Mahalanobis distance is likely to be strongly affected by the presence of outliers. Rousseeuw and van Zomeren (1990) thus recommend using robust methods (which are not excessively affected by the presence of outliers).

On the other hand, the choice of the threshold for classifying an observation as an outlier should be discussed. There is no apparent reason why a particular threshold should be applicable to all data sets (Filzmoser, Garrett, and Reimann 2005).

**Value**

An `OutlierIndex` object.

**Author(s)**

N. Frerebeau

**References**


See Also

Other outlier detection methods: *plot_outliers*

Examples

```r
## Data from Day et al. 2011
data("kommos", package = "folio") # Coerce to compositional data
kommos <- remove_NA(kommos, margin = 1) # Remove cases with missing values
coda <- as_composition(kommos, groups = 1) # Use ceramic types for grouping

## Detect outliers
out <- outliers(coda, groups = NULL, robust = FALSE)
plot(out) # Plot
plot(out, qq = TRUE) # Quantile-Quantile plot

## Detect outliers by group
out <- outliers(coda[, 1:15, drop = FALSE])

plot(out, ncol = 2) # Plot
plot(out, qq = TRUE, ncol = 4) # Quantile-Quantile plot
```

### pca_coda

**Principal Components Analysis**

**Description**

Computes a principal components analysis based on the singular value decomposition.

**Usage**

```r
## S4 method for signature 'CompositionMatrix'
pca(
  object,
  center = TRUE,
  scale = FALSE,
```
rank = NULL,
sup_row = NULL,
sup_col = NULL,
weight_row = NULL,
weight_col = NULL
)

## S4 method for signature 'LogRatio'
pca(
  object,
  center = TRUE,
  scale = FALSE,
  rank = NULL,
  sup_row = NULL,
  sup_col = NULL,
  weight_row = NULL,
  weight_col = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>A LogRatio object.</td>
</tr>
<tr>
<td>center</td>
<td>A logical scalar: should the variables be shifted to be zero centered?</td>
</tr>
<tr>
<td>scale</td>
<td>A logical scalar: should the variables be scaled to unit variance?</td>
</tr>
<tr>
<td>rank</td>
<td>An integer value specifying the maximal number of components to be kept in the results. If NULL (the default), ( p - 1 ) components will be returned.</td>
</tr>
<tr>
<td>sup_row</td>
<td>A vector specifying the indices of the supplementary rows.</td>
</tr>
<tr>
<td>sup_col</td>
<td>A vector specifying the indices of the supplementary columns.</td>
</tr>
<tr>
<td>weight_row</td>
<td>A numeric vector specifying the active row (individual) weights. If NULL (the default), uniform weights are used. Row weights are internally normalized to sum 1</td>
</tr>
<tr>
<td>weight_col</td>
<td>A numeric vector specifying the active column (variable) weights. If NULL (the default), uniform weights (1) are used.</td>
</tr>
</tbody>
</table>

Value

A dimensio::PCA object. See package dimensio for details.

Author(s)

N. Frerebeau

References


perturbation

See Also

dimensio::pca(), dimensio::biplot(), dimensio::screeplot(), dimensio::viz_individuals(), dimensio::viz_variables()

Examples

```r
## Data from Day et al. 2011
data("kommos", package = "folio") # Coerce to compositional data
kommos <- remove_NA(kommos, margin = 1) # Remove cases with missing values
coda <- as_composition(kommos, groups = 1) # Use ceramic types for grouping

## Centered log-ratio
clr <- transform_clr(coda)

## PCA
X <- pca(clr, scale = FALSE)

## Explore results
library(dimensio)

## Plot
viz_individuals(X, highlight = get_groups(coda), pch = 16)
viz_variables(X)
```

<table>
<thead>
<tr>
<th>perturbation</th>
<th>Perturbation Operation</th>
</tr>
</thead>
</table>

Description

Perturbation of two compositions.

Usage

```r
perturbation(x, y, ...)
```

Arguments

- `x, y` A numeric vector of compositional data or a CompositionMatrix object.
- `...` Currently not used.
Details

In compositional geometry, perturbation plays the role of sum (translation). It is the closed component-wise product of two compositions.

Value

A numeric vector.

Author(s)

N. Frerebeau

See Also

Other operations in the simplex: arithmetic, closure(), powering(), scalar()

Examples

```r
x <- as_composition(c(1, 2, 3))
y <- as_composition(c(1, 2, 1))

## Perturbation
perturbation(x, y)
x + y

## Powering
powering(y, 2)
y * 2

## Scalar product
scalar(x, y)
```

---

petrography  Can Sora Petrographic Data

Description

Can Sora Petrographic Data

Usage

petrography
Format

A data.frame with 21 variables:

VAR1  Optical activity.
VAR2  Inclusion orientation.
VAR3  Void orientation.
VAR4  Texture.
VAR5  Special components.
VAR6  Plutonic rocks.
VAR7  Volcanic rocks.
VAR8  Metamorphic rocks.
VAR9  Sedimentary rocks.
VAR10  Quartz.
VAR11  Feldspar.
VAR12  Plagioclase.
VAR13  Pyroxenes.
VAR14  Amphiboles.
VAR15  Micas.
VAR16  Phyllosilicates.
VAR17  Carbonates.
VAR18  Other constituents.
VAR19  Packing.

References


See Also

Other datasets: arctic, chemistry, hongite, slides
**plot**

*Plot Compositional Data*

**Description**

Displays a matrix of ternary plots.

**Usage**

```r
## S4 method for signature 'CompositionMatrix,missing'
plot(x, ..., margin = NULL)
```

**Arguments**

- `x`: A `CompositionMatrix` object.
- `...`: Further graphical parameters.
- `margin`: A character string or an integer giving the index of the column to be used as the third part of the ternary plots. If `NULL` (the default), marginal compositions will be used (i.e. the geometric mean of the non-selected parts).

**Value**

`plot()` is called for its side-effects: is results in a graphic being displayed (invisibly return `x`).

**Author(s)**

N. Frerebeau

**See Also**

`isopleuros::ternary_pairs()`, `isopleuros::ternary_plot()`

Other plot methods: `as_graph()`, `barplot()`, `plot_logratio`

**Examples**

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Ternary plots
plot(coda)
```
plot_logratio  

Plot Log-Ratios

Description
Displays a density plot.

Usage

```r
## S4 method for signature 'LogRatio,missing'
plot(
  x,
  ...,
  order = NULL,
  decreasing = FALSE,
  groups = get_groups(x),
  rug = TRUE,
  ticksize = 0.05,
  ncol = NULL,
  flip = FALSE,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  ann = graphics::par("ann"),
  axes = TRUE,
  frame.plot = axes,
  legend = list(x = "topright")
)
```

Arguments

- `x`: A LogRatio object.
- `...`: Further graphical parameters, particularly, border and col.
- `order`: A logical scalar: should the ratio be ordered?
- `decreasing`: A logical scalar: should the sort order be increasing or decreasing?
- `groups`: A factor in the sense that `as.factor(groups)` defines the grouping. If set, a matrix of panels defined by groups will be drawn.
- `rug`: A logical scalar: should a rug representation (1-d plot) of the data be added to the plot?
- `ticksize`: A length-one numeric vector giving the length of the ticks making up the rug. Positive lengths give inwards ticks. Only used if rug is TRUE.
- `ncol`: An integer specifying the number of columns to use when facet is "multiple". Defaults to 1 for up to 4 series, otherwise to 2.
- `flip`: A logical scalar: should the y-axis (ticks and numbering) be flipped from side 2 (left) to 4 (right) from variable to variable?
Description

Plot Outliers

plot_outliers

xlab, ylab  A character vector giving the x and y axis labels.
main  A character string giving a main title for the plot.
ann  A logical scalar: should the default annotation (title and x and y axis labels) appear on the plot?
axes  A logical scalar: should axes be drawn on the plot?
frame.plot  A logical scalar: should a box be drawn around the plot?
legend  A list of additional arguments to be passed to \texttt{graphics::legend()}; names of the list are used as argument names. If NULL, no legend is displayed.

Value

\texttt{plot()} is called for its side-effects: is results in a graphic being displayed (invisibly return \texttt{x}).

Author(s)

N. Frerebeau

See Also

Other plot methods: \texttt{as_graph()}, \texttt{barplot()}, \texttt{plot()}

Examples

```r
## Data from Day et al. 2011
data("kommos", package = "folio") # Coerce to compositional data
kommos <- remove_NA(kommos, margin = 1) # Remove cases with missing values
coda <- as_composition(kommos, groups = 1) # Use ceramic types for grouping

## Log ratio
clr <- transform_clr(coda)
plot(clr, group = NULL, flip = TRUE, border = "black", col = NA)
plot(clr, flip = TRUE)
```

Usage

```r
## S4 method for signature 'OutlierIndex,missing'
plot()
  x,
  ...
  qq = FALSE,
  probs = c(0.25, 0.75),
  ncol = NULL,
  flip = FALSE,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  sub = NULL,
  ann = graphics::par("ann"),
  axes = TRUE,
  frame.plot = axes,
  panel.first = NULL,
  panel.last = NULL
)```

Arguments

- **x**: An `OutlierIndex` object.
- **...**: Further graphical parameters.
- **qq**: A `logical` scalar: should a quantile-quantile plot be produced?
- **probs**: A length-two `numeric` vector representing probabilities. Corresponding quantile pairs define the line drawn (see `stats::qqline()`). Only used if `qq` is `TRUE`.
- **ncol**: An `integer` specifying the number of columns to use when facet is "multiple". Defaults to 1 for up to 4 series, otherwise to 2.
- **flip**: A `logical` scalar: should the y-axis (ticks and numbering) be flipped from side 2 (left) to 4 (right) from group to group?
- **xlab**, **ylab**: A `character` vector giving the x and y axis labels.
- **main**: A `character` string giving a main title for the plot.
- **sub**: A `character` string giving a subtitle for the plot.
- **ann**: A `logical` scalar: should the default annotation (title and x and y axis labels) appear on the plot?
- **axes**: A `logical` scalar: should axes be drawn on the plot?
- **frame.plot**: A `logical` scalar: should a box be drawn around the plot?
- **panel.first**: An an expression to be evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids.
- **panel.last**: An expression to be evaluated after plotting has taken place but before the axes, title and box are added.

Value

`plot()` is called for its side-effects: is results in a graphic being displayed (invisibly return `x`).
Author(s)
N. Frerebeau

References

See Also
Other outlier detection methods: `outliers()`

Examples
```r
## Data from Day et al. 2011
data("kommos", package = "folio") # Coerce to compositional data
kommos <- remove_NA(kommos, margin = 1) # Remove cases with missing values
coda <- as_composition(kommos, groups = 1) # Use ceramic types for grouping

## Detect outliers
out <- outliers(coda, groups = NULL, robust = FALSE)
plot(out) # Plot
plot(out, qq = TRUE) # Quantile-Quantile plot

## Detect outliers by group
out <- outliers(coda[, 1:15, drop = FALSE])
plot(out, ncol = 2) # Plot
plot(out, qq = TRUE, ncol = 4) # Quantile-Quantile plot
```

Description
Missing Values Policy

Details
Compositional data are quantitative (positive) descriptions of the parts of some whole, carrying relative, rather than absolute, information (i.e. only relative changes are relevant; Aitchison 1986). Basically, two situations can be outlined:
• The presence of zeros: these are considered as observed quantities, but which happen to be below the detection limit (thus interpreted as small unknown values).
• The presence of missing values (NA): these indicate that the quantities in question have not been observed.

When creating a `CompositionMatrix` object, the presence of zero and `NA` values is allowed: this makes it possible to explore and visualize the data while preserving the missing structure. However, the user must deal with these missing values before proceeding further (e.g. by removing incomplete cases or replacing the values concerned): log-ratio transformations cannot be computed in the presence of missing values.

**Note**

If you need more advanced features (e.g. imputation of missing values), you should consider the `compositions` or `robCompositions` package.

**References**


**See Also**

Other imputation methods: `missing`, `zero`
Details

In compositional geometry, powering replaces the product of a vector by a scalar (scaling) and is defined as the closed powering of the components by a given scalar.

Value

A numeric vector.

Author(s)

N. Frerebeau

See Also

Other operations in the simplex: arithmetic, closure(), perturbation(), scalar()

Examples

```r
x <- as_composition(c(1, 2, 3))
y <- as_composition(c(1, 2, 1))

## Perturbation
perturbation(x, y)
x + y

## Powering
powering(y, 2)
y * 2

## Scalar product
scalar(x, y)
```

Description

Retrieves or defines the sample names.

Usage

any_replicated(x)

is_replicated(x)

get_samples(x)

set_samples(x) <- value
## S4 method for signature 'CompositionMatrix'
is_replicated(x)

## S4 method for signature 'LogRatio'
is_replicated(x)

## S4 method for signature 'OutlierIndex'
is_replicated(x)

## S4 method for signature 'CompositionMatrix'
any_replicated(x)

## S4 method for signature 'LogRatio'
any_replicated(x)

## S4 method for signature 'OutlierIndex'
any_replicated(x)

## S4 method for signature 'CompositionMatrix'
get_samples(x)

## S4 method for signature 'LogRatio'
get_samples(x)

## S4 method for signature 'OutlierIndex'
get_samples(x)

## S4 replacement method for signature 'CompositionMatrix'
set_samples(x) <- value

### Arguments

- **x**: An object from which to get or set samples.
- **value**: A possible value for the samples of x.

### Details

In some situations, measurements may have been repeated (e.g., multiple chemical analyses on the same sample). The presence of repeated measurements can be specified by giving several observations the same sample name.

See vignette("nexus").

### Value

- `set_samples()` returns an object of the same sort as x with the new sample names assigned.
- `get_samples()` returns a `character` vector giving the sample names of x.
• `any_replicated()` returns a `logical` scalar specifying whether or not `x` has replicated observations.
• `is_replicated()` returns a `logical` vector specifying whether or not an observation is a replicate.

**Author(s)**
N. Frerebeau

**See Also**
Other mutators: `groups`, `identifiers`, `subset()`, `totals`

---

**scalar**

*Scalar Product*

**Description**
Computes the Aitchison scalar product of two compositions.

**Usage**
```
scalar(x, y, ...)
```

```r
## S4 method for signature 'numeric,numeric'
scalar(x, y)
```

```r
## S4 method for signature 'CompositionMatrix,CompositionMatrix'
scalar(x, y)
```

**Arguments**
- `x, y` A `CompositionMatrix` object.
- `...` Currently not used.

**Value**
A `numeric` vector.

**Author(s)**
N. Frerebeau

**See Also**
Other operations in the simplex: `arithmetic`, `closure()`, `perturbation()`, `powering()`
Examples

```r
x <- as_composition(c(1, 2, 3))
y <- as_composition(c(1, 2, 1))

## Perturbation
perturbation(x, y)
x + y

## Powering
powering(y, 2)
y * 2

## Scalar product
scalar(x, y)
```

---

scale

**Scaling and Centering of Compositional Data**

**Description**

Scaling and Centering of Compositional Data

**Usage**

```r
## S4 method for signature 'CompositionMatrix'
scale(x, center = TRUE, scale = TRUE)
```

**Arguments**

- `x`: A `CompositionMatrix` object.
- `center`: A `logical` scalar or a `numeric` vector giving the center to be substracted.
- `scale`: A `logical` scalar or a length-one `numeric` vector giving a scaling factor for multiplication.

**Value**

A `CompositionMatrix` object.

**Author(s)**

N. Frerebeau

**References**


See Also

Other statistics: `aggregate()`, `covariance()`, `dist.mahalanobis()`, `margin()`, `mean()`, `metric_var()`, `variation()`

Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Center and scale
scaled <- scale(coda, center = TRUE, scale = TRUE)
mean(scaled)
head(scaled)
```

<table>
<thead>
<tr>
<th>slides</th>
<th>Thin Sections</th>
</tr>
</thead>
</table>

Description

Mineral compositions of five slides as reported by five analysts.

Usage

slides

Format

A `data.frame` with 9 variables:

- `analyst` Analyst number.
- `slide` Slide number.
- `quartz` Quartz (percent).
- `microcline` Microcline (percent).
- `plagioclase` Plagioclase (percent).
- `biotite` Biotite (percent).
- `plagioclase` Plagioclase (percent).
- `muscovite` Muscovite (percent).
- `opales` Opaque minerals (percent).
- `nonopales` Non-opaque minerals (percent).

References

See Also
Other datasets: arctic, chemistry, hongite, petrography

subset
Extract or Replace Parts of an Object

Description
Operators acting on objects to extract or replace parts.

Usage

## S4 method for signature 'CompositionMatrix,missing,missing,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CompositionMatrix,missing,missing,logical'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CompositionMatrix,index,missing,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CompositionMatrix,index,missing,logical'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CompositionMatrix,missing,index,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CompositionMatrix,missing,index,logical'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CompositionMatrix,index,index,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'CompositionMatrix,index,index,logical'
x[i, j, ..., drop = TRUE]

## S4 replacement method for signature 'CompositionMatrix'
x[i, j, ...] <- value

## S4 replacement method for signature 'CompositionMatrix'
x[[i, j, ...]] <- value

Arguments

x
An object from which to extract element(s) or in which to replace element(s).
**subset**

Indices specifying elements to extract or replace. Indices are numeric, integer or character vectors or empty (missing) or NULL. Numeric values are coerced to integer as by \texttt{as.integer()}. Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected.

... Currently not used.

**drop** A logical scalar: should the result be coerced to the lowest possible dimension? This only works for extracting elements, not for the replacement.

**value** A possible value for the element(s) of \texttt{x}.

**Value**

A subsetted object of the same sort as \texttt{x}.

**Subcomposition**

If \texttt{drop} is \texttt{FALSE}, subsetting some of the possible components of a \texttt{CompositionMatrix} object will produce a closed subcomposition (see examples).

**Author(s)**

N. Frerebeau

**See Also**

Other mutators: \texttt{groups}, \texttt{identifiers}, \texttt{samples}, \texttt{totals}

**Examples**

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)
head(coda)

## Subset
coda[[1, 1]] # Get the first value
coda[1] # Get the first value
coda[, ] # Get all values
coda[1, , drop = FALSE] # Get the first row

## Subcomposition
subcoda <- coda[, 1:3, drop = FALSE] # Get the first three column
head(subcoda)
```
totals | Row Sums
---|---

**Description**
Retrieves or defines the row sums (before closure).

**Usage**

```r
get_totals(x)
```

```r
set_totals(x) <- value
```

```r
## S4 method for signature 'CompositionMatrix'
get_totals(x)
```

```r
## S4 replacement method for signature 'CompositionMatrix'
set_totals(x) <- value
```

**Arguments**

- `x`: An object from which to get or set totals.
- `value`: A possible value for the totals of `x`.

**Value**

- `set_totals()` returns an object of the same sort as `x` with the new row sums assigned.
- `get_totals()` returns the row sums of `x`.

**Author(s)**

N. Frerebeau

**See Also**

Other mutators: `groups`, `identifiers`, `samples`, `subset()`

**Examples**

```r
## Create a count matrix
A1 <- matrix(data = as.numeric(sample(1:100, 100, TRUE)), nrow = 20)

## Coerce to compositions
B <- as_composition(A1)

## Row sums are internally stored before coercing to relative frequencies
get_totals(B)
```
### Description

Computes ALR transformation.

### Usage

```r
transform_alr(object, ...)  
```

```r
## S4 method for signature 'CompositionMatrix'  
transform_alr(object, j = ncol(object))  
```

### Arguments

- **object**: A `CompositionMatrix` object.
- **...**: Currently not used.
- **j**: An `integer` giving the index of the rationing part (denominator).

### Details

The ALR transformation is the logratio of a pair of parts with respect to a fixed part.

### Value

An `ALR` object.

### Author(s)

N. Frerebeau

### References


See Also

Other log-ratio transformations: `transform_clr()`, `transform_ilr()`, `transform_inverse()`, `transform_lr()`, `transform_plr()`

Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Pairwise log-ratio
lr <- transform_lr(coda)

## Centered log-ratio
clr <- transform_clr(coda)

## Additive log-ratio
alr <- transform_alr(coda)

## Isometric log-ratio
ilr <- transform_ilr(coda)
plr <- transform_plr(coda)

## Inverse transformation
inv_clr <- transform_inverse(clr)
all.equal(coda, inv_clr)

inv_alr <- transform_inverse(alr)
all.equal(coda, inv_alr)

inv_ilr <- transform_inverse(ilr)
all.equal(coda, inv_ilr)

inv_plr <- transform_inverse(plr)
all.equal(coda, inv_plr)
```

---

**transform_clr**

**Centered Log-Ratios (CLR)**

**Description**

Computes CLR transformation.

**Usage**

`transform_clr(object, ...)`
## S4 method for signature 'CompositionMatrix'
transform_clr(object, weights = FALSE)

**Arguments**
- `object`: A `CompositionMatrix` object.
- `...`: Currently not used.
- `weights`: A logical scalar: should a varying weight be used. If FALSE (the default), equally-weighted parts are used. Alternatively, a positive numeric vector of weights can be specified.

**Details**
The CLR transformation computes the log of each part relative to the geometric mean of all parts.

**Value**
A `CLR` object.

**Author(s)**
N. Frerebeau

**References**

**See Also**
Other log-ratio transformations: `transform_alr()`, `transform_ilr()`, `transform_inverse()`, `transform_lr()`, `transform_plr()`

**Examples**
```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Pairwise log-ratio
lr <- transform_lr(coda)

## Centered log-ratio
clr <- transform_clr(coda)

## Additive log-ratio
```
alr <- transform_alr(coda)

## Isometric log-ratio
ilr <- transform_ilr(coda)
plr <- transform_plr(coda)

## Inverse transformation
inv_clr <- transform_inverse(clr)
all.equal(coda, inv_clr)

inv_alr <- transform_inverse(alr)
all.equal(coda, inv_alr)

inv_ilr <- transform_inverse(ilr)
all.equal(coda, inv_ilr)

inv_plr <- transform_inverse(plr)
all.equal(coda, inv_plr)

---

**transform_ilr**  
*Isometric Log-Ratios (ILR)*

**Description**

Computes ILR transformations.

**Usage**

```r
transform_ilr(object, ...)
```

## S4 method for signature 'CompositionMatrix'
transform_ilr(object)

**Arguments**

- `object`: A `CompositionMatrix` object.
- `...`: Currently not used.

**Details**

The ILR transformation provides the coordinates of any composition with respect to a given orthonormal basis. `transform_ilr()` uses the orthonormal basis (Helmert matrix) originally defined by Egozcue *et al.* (2003).

**Value**

An ILR object.
Author(s)
N. Frerebeau

References

See Also
Other log-ratio transformations: `transform_alr()`, `transform_clr()`, `transform_inverse()`, `transform_lr()`, `transform_plr()`

Examples
```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Pairwise log-ratio
lr <- transform_lr(coda)

## Centered log-ratio
clr <- transform_clr(coda)

## Additive log-ratio
alr <- transform_alr(coda)

## Isometric log-ratio
ilr <- transform_ilr(coda)
plr <- transform_plr(coda)

## Inverse transformation
inv_clr <- transform_inverse(clr)
all.equal(coda, inv_clr)

inv_alr <- transform_inverse(alr)
all.equal(coda, inv_alr)

inv_ilr <- transform_inverse(ilr)
all.equal(coda, inv_ilr)

inv_plr <- transform_inverse(plr)
all.equal(coda, inv_plr)
```
transform_inverse  Inverse Log-Ratio Transformation

Description

Computes inverse log-ratio transformations.

Usage

transform_inverse(object, origin, ...)

## S4 method for signature 'CLR,missing'
transform_inverse(object)

## S4 method for signature 'ALR,missing'
transform_inverse(object)

## S4 method for signature 'ILR,missing'
transform_inverse(object)

## S4 method for signature 'matrix,ILR'
transform_inverse(object, origin)

Arguments

object  A LogRatio object.
origin  A LogRatio object to be used for the inverse transformation.
...     Currently not used.

Value

A CompositionMatrix object.

Author(s)

N. Frerebeau

References


See Also

Other log-ratio transformations: `transform_alr()`, `transform_clr()`, `transform_ilr()`, `transform_lr()`, `transform_plr()`

Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Pairwise log-ratio
lr <- transform_lr(coda)

## Centered log-ratio
clr <- transform_clr(coda)

## Additive log-ratio
alr <- transform_alr(coda)

## Isometric log-ratio
ilr <- transform_ilr(coda)
plr <- transform_plr(coda)

## Inverse transformation
inv_clr <- transform_inverse(clr)
all.equal(coda, inv_clr)

inv_alr <- transform_inverse(alr)
all.equal(coda, inv_alr)

inv_ilr <- transform_inverse(ilr)
all.equal(coda, inv_ilr)

inv_plr <- transform_inverse(plr)
all.equal(coda, inv_plr)
```

transform_lr

Pairwise Log-Ratios (LR)

Description

Computes all pairwise log-ratio transformation.

Usage

`transform_lr(object, ...)`
## S4 method for signature 'CompositionMatrix'
transform_lr(object)

### Arguments

- **object**
  - A `CompositionMatrix` object.
- **...**
  - Currently not used.

### Value

A `LR` object.

### Author(s)

N. Frerebeau

### References


### See Also

Other log-ratio transformations: `transform_alr()`, `transform_clr()`, `transform_ilr()`, `transform_inverse()`, `transform_plr()`

### Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Pairwise log-ratio
lr <- transform_lr(coda)

## Centered log-ratio
clr <- transform_clr(coda)

## Additive log-ratio
alr <- transform_alr(coda)

## Isometric log-ratio
ilr <- transform_ilr(coda)
plr <- transform_plr(coda)

## Inverse transformation
inv_clr <- transform_inverse(clr)
```
all.equal(coda, inv_clr)

inv_alr <- transform_inverse(alr)
all.equal(coda, inv_alr)

inv_ilr <- transform_inverse(ilr)
all.equal(coda, inv_ilr)

inv_plr <- transform_inverse(plr)
all.equal(coda, inv_plr)

---

**transform_plr**  
*Pivot Log-Ratios (PLR)*

**Description**

Computes PLR transformations.

**Usage**

```r
transform_plr(object, ...)
```

```r
## S4 method for signature 'CompositionMatrix'
transform_plr(object, pivot = 1)
```

**Arguments**

- `object`  
  A `CompositionMatrix` object.
- `...`  
  Currently not used.
- `pivot`  
  An integer giving the index of the pivotal variable.

**Value**

A PLR object.

**Author(s)**

N. Frerebeau

**References**


Variation Matrix

Description

Computes the variation matrix (Aitchison 1986, definition 4.4).

Usage

variation(x, ...)

## S4 method for signature 'CompositionMatrix'
variation(x)
Arguments

x A CompositionMatrix object.

... Currently not used.

Value

A matrix.

Author(s)

N. Frerebeau

References


See Also

Other statistics: aggregate(), covariance(), dist, mahalanobis(), margin(), mean(), metric_var(), scale()

Examples

```r
## Data from Aitchison 1986
data("hongite")

## Coerce to compositional data
coda <- as_composition(hongite)

## Variation matrix
## (Aitchison 1986, definition 4.4)
(varia <- variation(coda))

## Cluster dendrogram
d <- as.dist(varia)
h <- hclust(d, method = "ward.D2")
plot(h)
```

Description

Multiplicative replacement of zeros.
Usage

```r
## S4 method for signature 'CompositionMatrix'
replace_zero(x, value, delta = 2/3)
```

Arguments

- `x`: A `CompositionMatrix` object.
- `value`: A `numeric` vector giving the detection limits of each part (in \((0, 1)\)).
- `delta`: A `numeric` vector specifying the fraction of the detection limit to be used in replacement.

Value

An `CompositionMatrix` object, where all zero values have been replaced.

Author(s)

N. Frerebeau

References


See Also

Other imputation methods: `missing.policy`

Examples

```r
## Data from Martín-Fernández et al. 2003
X <- data.frame(
  X1 = c(0.0000, 0.1304, 0.1963),
  X2 = c(0.1250, 0.3151, NA),
  X3 = c(0.1237, NA, NA),
  X4 = c(0.7253, 0.2002, 0.0819),
  X5 = c(0.0260, 0.3543, 0.0114)
)

## Coerce to a compositional matrix
Y <- as_composition(X)

## Replace zeros
Z <- replace_zero(Y, value = 0.02, delta = 2/3)
Z
```
Index

* compositional data tools
  as_amounts, 6
  as_composition, 7
  as_features, 9
* datasets
  arctic, 4
  chemistry, 13
  hongite, 19
  petrography, 33
  slides, 45
* imputation methods
  missing, 26
  policy, 39
  zero, 59
* log-ratio transformations
  transform_alr, 49
  transform_clr, 50
  transform_ilr, 52
  transform_inverse, 54
  transform_lr, 55
  transform_plr, 57
* multivariate analysis
  pca_coda, 30
* mutators
  groups, 17
  identifiers, 20
  samples, 41
  subset, 46
  totals, 48
* operations in the simplex
  arithmetic, 5
  closure, 13
  perturbation, 32
  powering, 40
  scalar, 43
* outlier detection methods
  outliers, 28
  plot_outliers, 37
* plot methods
  as_graph, 10
  barplot, 11
  plot, 35
  plot_logratio, 36
* sourcing methods
  mix, 27
* statistics
  aggregate, 3
  covariance, 14
  dist, 16
  mahalanobis, 21
  margin, 22
  mean, 23
  metric_var, 24
  scale, 44
  variation, 58
  [,CompositionMatrix,index,index,logical-method (subset), 46
  [,CompositionMatrix,index,index,missing-method (subset), 46
  [,CompositionMatrix,index,missing,logical-method (subset), 46
  [,CompositionMatrix,index,missing,missing-method (subset), 46
  [,CompositionMatrix,missing,index,logical-method (subset), 46
  [,CompositionMatrix,missing,index,missing-method (subset), 46
  [,CompositionMatrix,missing,missing,logical-method (subset), 46
  [,CompositionMatrix,missing,missing,missing-method (subset), 46
  [<-,CompositionMatrix-method (subset), 46
  [[<-,CompositionMatrix-method (subset), 46
  %perturbe%(arithmetic), 5
  %perturbe%,CompositionMatrix,CompositionMatrix-method (arithmetic), 5
function, 3
get_groups(groups), 17
get_groups,CompositionMatrix-method (groups), 17
get_groups,LogRatio-method (groups), 17
get_groups,OutlierIndex-method (groups), 17
get_groups-method (groups), 17
get_identifiers(identifiers), 20
get_identifiers,CompositionMatrix-method (identifiers), 20
get_identifiers,LogRatio-method (identifiers), 20
get_identifiers,OutlierIndex-method (identifiers), 20
get_identifiers-method (identifiers), 20
get_samples(samples), 41
get_samples,CompositionMatrix-method (samples), 41
get_samples,LogRatio-method (samples), 41
get_samples,OutlierIndex-method (samples), 41
get_samples-method (samples), 41
get_totals(totals), 48
get_totals,CompositionMatrix-method (totals), 48
get_totals-method (totals), 48
get_totals-method (totals), 48
graphical parameters, 35, 36, 38
graphics::barplot(), 12
graphics::legend(), 37
group data by site/area, 8
groups, 17, 20, 43, 47, 48
hongite, 4, 13, 19, 34, 46
identifiers, 19, 20, 43, 47, 48
ILR, 21, 52
integer, 8, 12, 16, 22, 31, 35, 36, 38, 47, 49, 57
is_assigned(groups), 17
is Assigned,CompositionMatrix-method (groups), 17
is Assigned,LogRatio-method (groups), 17
is Assigned,OutlierIndex-method (groups), 17
is Assigned-method (groups), 17
is_replicated(samples), 41
is_replicated,CompositionMatrix-method (samples), 41
is_replicated,LogRatio-method (samples), 41
is_replicated,OutlierIndex-method (samples), 41
is_replicated-method (samples), 41
isopleuros::ternary_pairs(), 35
isopleuros::ternary_plot(), 35
laboratory codes, 8
list, 37
logical, 8, 12, 14–16, 18, 21, 23, 29, 31, 36–38, 43, 44, 47, 51
LogRatio, 10, 31, 36, 54
LR, 56
mahalanobis, 3, 15, 17, 21, 22, 24, 25, 45, 59
mahalanobis,CompositionMatrix-method (mahalanobis), 21
mahalanobis,ILR-method (mahalanobis), 21
margin, 3, 15, 17, 21, 22, 24, 25, 45, 59
margin,CompositionMatrix-method (margin), 22
margin-method (margin), 22
MASS::cov.rob(), 21, 29
matrix, 3, 6, 7, 15, 27, 59
mean, 3, 15, 17, 21, 22, 23, 25, 45, 59
mean,CompositionMatrix-method (mean), 23
metric_sd(metric_var), 24
metric_sd,CompositionMatrix-method (metric_var), 24
metric_sd-method (metric_var), 24
metric_var, 3, 15, 17, 21, 22, 24, 24, 45, 59
metric_var,CompositionMatrix-method (metric_var), 24
metric_var-method (metric_var), 24
missing, 26, 40, 60
mix, 27
mix,dist,dist-method (mix), 27
mix,matrix,matrix-method (mix), 27
mix-method (mix), 27
NA, 40
numeric, 5, 6, 14, 21, 23, 25–27, 29, 31–33, 36, 38, 40, 41, 43, 44, 47, 51, 60
OutlierIndex, 29, 38
outliers, 28, 39
perturbation, CompositionMatrix-method
(outliers), 28
outliers-method (outliers), 28

pca, CompositionMatrix-method
(pca_coda), 30
pca, LogRatio-method (pca_coda), 30
pca_coda, 30
perturbation, 5, 14, 32, 41, 43
Perturbation operation, 5
perturbation, CompositionMatrix, CompositionMatrix-method
(perturbation), 32
perturbation, CompositionMatrix, matrix-method
(perturbation), 32
perturbation, CompositionMatrix, numeric-method
(perturbation), 32
perturbation, numeric, numeric-method
(perturbation), 32
perturbation-method (perturbation), 32
petrography, 4, 13, 19, 33, 46
plot, 10, 12, 35, 37
plot, CompositionMatrix, missing-method
(plot), 35
plot, LogRatio, missing-method
(plot_logratio), 36
plot, OutlierIndex, missing-method
(plot_outliers), 37
plot_outliers, 10, 12, 35, 36
PLR, 57
policy, 26, 39, 60
powering, 5, 14, 33, 40, 43
Powering operation, 5
powering, CompositionMatrix, numeric-method
(powering), 40
powering, numeric, numeric-method
(powering), 40
powering-method (powering), 40

repeated measurements/observation, 8
replace NA, CompositionMatrix-method
(missing), 26
replace zero, CompositionMatrix-method
(zero), 59
robCompositions, 40

samples, 19, 20, 41, 47, 48
scalar, 5, 14, 33, 41, 43
scalar, CompositionMatrix, CompositionMatrix-method
(scalar), 43
scalar, numeric, numeric-method (scalar), 43
scalar-method (scalar), 43
scale, 3, 15, 17, 21, 22, 24, 25, 44, 59
scale, CompositionMatrix-method (scale), 44
set_groups, CompositionMatrix-method
(groups), 17
set_groups-method (groups), 17
set_groups<- (groups), 17
set_groups<-, CompositionMatrix-method
(groups), 17
set_identifiers, CompositionMatrix-method
(identifiers), 20
set_identifiers-method (identifiers), 20
set_identifiers<- (identifiers), 20
set_identifiers<-, CompositionMatrix-method
(identifiers), 20
set_samples, CompositionMatrix-method
(samples), 41
set_samples-method (samples), 41
set_samples<- (samples), 41
set_samples<-, CompositionMatrix-method
(samples), 41
set_totals, CompositionMatrix-method
(totals), 48
set_totals-method (totals), 48
set_totals<- (totals), 48
set_totals<-, CompositionMatrix-method
(totals), 48
slides, 4, 13, 19, 34, 45
stats::cov(), 15
stats::dist, 16, 27
stats::dist(), 16, 17
stats::mahalanobis(), 21
stats::qqline(), 38
subset, 19, 20, 43, 46, 48
totals, 19, 20, 43, 47, 48
transform_alr, 49, 51, 53, 55, 56, 58
transform_alr, CompositionMatrix-method
(transform_alr), 49
transform_alr-method (transform_alr), 49
transform_clr, 50, 50, 53, 55, 56, 58
transform_clr, CompositionMatrix-method
(transform_clr), 50
transform_clr-method (transform_clr), 50
transform_ilr, 50, 51, 52, 55, 56, 58
transform_ilr,CompositionMatrix-method
(transform_transform_ilr), 52
transform_inverse, 50, 51, 53, 54, 56, 58
transform_inverse,ALR,missing-method
(transform_transform_inverse), 54
transform_inverse,CLR,missing-method
(transform_transform_inverse), 54
transform_inverse,ILR,missing-method
(transform_transform_inverse), 54
transform_inverse,method
(transform_transform_inverse), 54
transform_inverse,method
(transform_transform_inverse), 54
transform_lr, 50, 51, 53, 55, 58
transform_lr,CompositionMatrix-method
(transform_transform_lr), 55
transform_plr, 50, 51, 53, 55, 56, 57
transform_plr,CompositionMatrix-method
(transform_transform_plr), 57
transform_plr-method (transform_transform_plr), 57
variation, 3, 15, 17, 21, 22, 24, 25, 45, 58
variation,CompositionMatrix-method
(variation), 58
variation-method (variation), 58
zero, 26, 40, 59