

# Package ‘tabula’

December 3, 2018

**Title** Analysis, Seriation and Visualization of Archaeological Count Data

**Version** 1.0.0

**Date** 2018-11-22

**Maintainer** Nicolas Frerebeau <data@archaeo.science>

**Description** An easy way to examine archaeological count data (artifacts, faunal remains, etc.). This package includes several measures of diversity, e.g. richness, rarefaction, diversity, turnover, similarity, etc. It also provides matrix seriation methods for chronological modeling and dating. The package make it easy to visualize count data and statistical thresholds: rank/abundance plots, Ford and Bertin diagrams, etc.

**URL** <http://github.com/nfrerebeau/tabula>

**BugReports** <http://github.com/nfrerebeau/tabula/issues>

**Depends** R (>= 3.4)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Collate** 'utilities.R' 'tabula.R' 'AllClasses.R' 'AllGenerics.R'  
'LogicalMatrix.R' 'NumericMatrix.R' 'index-richness.R'  
'index-diversity.R' 'alpha-diversity.R' 'index-turnover.R'  
'index-similarity.R' 'beta-diversity.R' 'data.R'  
'method-seriation.R' 'statistics.R' 'plot.R' 'seriation.R'

**RoxygenNote** 6.1.1

**Suggests** testthat, knitr, rmarkdown

**Imports** dplyr (>= 0.7), FactoMineR, ggplot2 (>= 3.0.0), grDevices,  
magrittr, methods, rlang, stats, tidyr, utils

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Nicolas Frerebeau [aut, cre] (<<https://orcid.org/0000-0001-5759-4944>>),  
Brice Lebrun [ctb] (<<https://orcid.org/0000-0001-7503-8685>>),  
Matthew Peeples [ctb] (<<https://orcid.org/0000-0003-4496-623X>>)

**Repository** CRAN

**Date/Publication** 2018-12-03 09:40:03 UTC

## R topics documented:

|                        |    |
|------------------------|----|
| tabula-package         | 2  |
| accessors              | 3  |
| alpha-diversity        | 4  |
| beta-diversity         | 8  |
| BootCA-class           | 11 |
| boves                  | 12 |
| compiegne              | 13 |
| CountMatrix-class      | 14 |
| FrequencyMatrix-class  | 15 |
| IncidenceMatrix-class  | 17 |
| mississippi            | 18 |
| OccurrenceMatrix-class | 19 |
| PermutationOrder-class | 20 |
| plotBar                | 21 |
| plotMatrix             | 22 |
| plotRank               | 24 |
| plotSpot               | 25 |
| seriation              | 26 |
| zuni                   | 30 |

**Index** **31**

---

|                |   |
|----------------|---|
| tabula-package | <i>Analysis, Seriation and Visualization of Archaeological Count Data</i> |
|----------------|---|

---

## Description

An easy way to examine archaeological count data (artifacts, faunal remains, etc.). This package includes several measures of diversity, e.g. richness, rarefaction, diversity, turnover, similarity, etc. It also provides matrix seriation methods for chronological modeling and dating. The package make it easy to visualize count data and statistical thresholds: rank/abundance plots, Ford and Bertin diagrams, etc.

## Details

**Package:** tabula  
**Type:** Package  
**Version:** 1.0.0  
**Date:** 2018-11-22  
**License:** GPL-3  
**DOI:** [10.5281/zenodo.1489944](https://doi.org/10.5281/zenodo.1489944)

**Author(s)****Full list of authors and contributors** (alphabetic order)

Nicolas Frerebeau *IRAMAT-CRP2A, Université Bordeaux Montaigne, France*  
Brice Lebrun *CEREGE, Aix-Marseille Université, France*  
Matthew Peeples *SNArchy, Arizona State University, USA*

**Bug reporting**

- <https://github.com/nfrerebeau/tabula/issues>

**Package maintainer**

Nicolas Frerebeau, IRAMAT-CRP2A, Université Bordeaux Montaigne, Pessac, France,  
<nicolas.frerebeau@u-bordeaux-montaigne.fr>

---

accessors

*Accessors*

---

**Description**

Accessors

**Usage**

columns(x)

method(x)

rows(x)

totals(x)

**Arguments**

x                    An object.

**Author(s)**

N. Frerebeau

---

|                 |                    |
|-----------------|--------------------|
| alpha-diversity | <i>α-diversity</i> |
|-----------------|--------------------|

---

### Description

Measures within-sample diversity. `diversity` returns a diversity or dominance index. `evenness` returns an evenness measure. `richness` returns sample richness. `rarefaction` returns Hurlbert's unbiased estimate of Sander's rarefaction.

### Usage

```
diversity(object, ...)

evenness(object, ...)

rarefaction(object, ...)

richness(object, ...)

## S4 method for signature 'CountMatrix'
richness(object, method = c("margalef",
  "menhinick"), simplify = FALSE)

## S4 method for signature 'CountMatrix'
rarefaction(object, sample)

## S4 method for signature 'CountMatrix'
diversity(object, method = c("berger",
  "brillouin", "mcintosh", "shannon", "simpson"), simplify = FALSE, ...)

## S4 method for signature 'CountMatrix'
evenness(object, method = c("brillouin",
  "mcintosh", "shannon", "simpson"), simplify = FALSE, ...)
```

### Arguments

|                       |   |
|-----------------------|---|
| <code>object</code>   | A $m \times p$ matrix of count data.  |
| <code>...</code>      | Further arguments passed to other methods.  |
| <code>method</code>   | A <a href="#">character</a> string specifying the index to be computed. This must be one or more of "berger", "brillouin", "margalef", "mcintosh", "menhinick", "shannon", "simpson" (see details). Any unambiguous substring can be given. |
| <code>simplify</code> | A <a href="#">logical</a> scalar: should the result be simplified to a matrix? The default value, FALSE, returns a list.  |
| <code>sample</code>   | A length-one <a href="#">numeric</a> vector giving the sub-sample size.   |

**Value**

rarefaction returns a numeric vector.

If `simplify` is FALSE, then diversity, evenness and richness return a list (default), else return a matrix.

**Richness and Rarefaction**

The number of different taxa, provides an instantly comprehensible expression of diversity. While the number of taxa within a sample is easy to ascertain, as a term, it makes little sense: some taxa may not have been seen, or there may not be a fixed number of taxa (e.g. in an open system; Peet 1974). As an alternative, *richness* ( $S$ ) can be used for the concept of taxa number (McIntosh 1967).

It is not always possible to ensure that all sample sizes are equal and the number of different taxa increases with sample size and sampling effort (Magurran 1988). Then, *rarefaction* ( $E(S)$ ) is the number of taxa expected if all samples were of a standard size (i.e. taxa per fixed number of individuals). Rarefaction assumes that imbalances between taxa are due to sampling and not to differences in actual abundances.

The following richness measures are available:

**margalef** Margalef richness index:  $D_{Mg} = \frac{S-1}{\ln N}$

**menhinick** Menhinick richness index:  $D_{Mn} = \frac{S}{\sqrt{N}}$

**Diversity and Evenness**

*Diversity* measurement assumes that all individuals in a specific taxa are equivalent and that all types are equally different from each other (Peet 1974). A measure of diversity can be achieved by using indices built on the relative abundance of taxa. These indices (sometimes referred to as non-parametric indices) benefit from not making assumptions about the underlying distribution of taxa abundance: they only take evenness and *richness* into account. Peet (1974) refers to them as indices of *heterogeneity*.

Diversity indices focus on one aspect of the taxa abundance and emphasize either *richness* (weighting towards uncommon taxa) or dominance (weighting towards abundant taxa; Magurran 1988).

*Evenness* is a measure of how evenly individuals are distributed across the sample.

The following heterogeneity index and corresponding evenness measures are available (see Magurran 1988 for details):

**berger** Berger-Parker dominance index. The Berger-Parker index expresses the proportional importance of the most abundant type.

**brillouin** Brillouin diversity index. The Brillouin index describes a known collection: it does not assume random sampling in an infinite population. Pielou (1975) and Laxton (1978) argues for the use of the Brillouin index in all circumstances, especially in preference to the Shannon index.

**mcintosh** McIntosh dominance index. The McIntosh index expresses the heterogeneity of a sample in geometric terms. It describes the sample as a point of a S-dimensional hypervolume and uses the Euclidean distance of this point from the origin.

**shannon** Shannon-Wiener diversity index. The Shannon index assumes that individuals are randomly sampled from an infinite population and that all taxa are represented in the sample (it does not reflect the sample size). The main source of error arises from the failure to include all taxa in the sample: this error increases as the proportion of species discovered in the sample declines (Peet 1974, Magurran 1988). The maximum likelihood estimator (MLE) is used for the relative abundance, this is known to be negatively biased.

**simpson** Simpson dominance index for finite sample. The Simpson index expresses the probability that two individuals randomly picked from a finite sample belong to two different types. It can be interpreted as the weighted mean of the proportional abundances.

The berger, mcintosh and simpson methods return a *dominance* index, not the reciprocal or inverse form usually adopted, so that an increase in the value of the index accompanies a decrease in diversity.

#### Note

Ramanujan approximation is used for  $x!$  computation if  $x > 170$ .

#### Author(s)

N. Frerebeau

#### References

- Berger, W. H. & Parker, F. L. (1970). Diversity of Planktonic Foraminifera in Deep-Sea Sediments. *Science*, 168(3937), 1345-1347. DOI: [10.1126/science.168.3937.1345](https://doi.org/10.1126/science.168.3937.1345).
- Brillouin, L. (1956). *Science and information theory*. New York: Academic Press.
- Hurlbert, S. H. (1971). The Nonconcept of Species Diversity: A Critique and Alternative Parameters. *Ecology*, 52(4), 577-586. DOI: [10.2307/1934145](https://doi.org/10.2307/1934145).
- Laxton, R. R. (1978). The measure of diversity. *Journal of Theoretical Biology*, 70(1), 51-67. DOI: [10.1016/0022-5193\(78\)90302-8](https://doi.org/10.1016/0022-5193(78)90302-8).
- Magurran, A. E. (1988). *Ecological Diversity and its Measurement*. Princeton, NJ: Princeton University Press. DOI:[10.1007/978-94-015-7358-0](https://doi.org/10.1007/978-94-015-7358-0).
- Margalef, R. (1958). Information Theory in Ecology. *General Systems*, 3, 36-71.
- McIntosh, R. P. (1967). An Index of Diversity and the Relation of Certain Concepts to Diversity. *Ecology*, 48(3), 392-404. DOI: [10.2307/1932674](https://doi.org/10.2307/1932674).
- Menhinick, E. F. (1964). A Comparison of Some Species-Individuals Diversity Indices Applied to Samples of Field Insects. *Ecology*, 45(4), 859-861. DOI: [10.2307/1934933](https://doi.org/10.2307/1934933).
- Peet, R. K. (1974). The Measurement of Species Diversity. *Annual Review of Ecology and Systematics*, 5(1), 285-307. DOI: [10.1146/annurev.es.05.110174.001441](https://doi.org/10.1146/annurev.es.05.110174.001441).
- Pielou, E. C. (1975). *Ecological Diversity*. New York: Wiley. DOI: [10.4319/lo.1977.22.1.0174b](https://doi.org/10.4319/lo.1977.22.1.0174b)
- Sander, H. L. (1968). Marine Benthic Diversity: A Comparative Study. *The American Naturalist*, 102(925), 243-282.
- Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27, 379-423. DOI: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).
- Simpson, E. H. (1949). Measurement of Diversity. *Nature*, 163(4148), 688-688. DOI: [10.1038/163688a0](https://doi.org/10.1038/163688a0).

**See Also**[turnover similarity](#)**Examples**

```

# Richness
## Margalef and Menhinick index
## Data from Magurran 1988, p. 128-129
trap <- CountMatrix(data = c(9, 3, 0, 4, 2, 1, 1, 0, 1, 0, 1, 1,
                             1, 0, 1, 0, 0, 0, 1, 2, 0, 5, 3, 0),
                    nrow = 2, byrow = TRUE, dimnames = list(c("A", "B"), NULL))

richness(trap, method = c("margalef", "menhinick"), simplify = TRUE)
## A 2.55 1.88
## B 1.95 1.66

# Rarefaction
rarefaction(trap, sample = 13) # 6.56

# Diversity
## Shannon diversity index
## Data from Magurran 1988, p. 145-149
birds <- CountMatrix(
  data = c(35, 26, 25, 21, 16, 11, 6, 5, 3, 3,
           3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 0, 0,
           30, 30, 3, 65, 20, 11, 0, 4, 2, 14,
           0, 3, 9, 0, 0, 5, 0, 0, 0, 0, 1, 1),
  nrow = 2, byrow = TRUE, dimnames = list(c("oakwood", "spruce"), NULL))

diversity(birds, "shannon") # 2.40 2.06
evenness(birds, "shannon") # 0.80 0.78

## Brillouin diversity index
## Data from Magurran 1988, p. 150-151
moths <- CountMatrix(data = c(17, 15, 11, 4, 4, 3, 3, 3, 2, 2, 1, 1, 1),
                    nrow = 1, byrow = TRUE)

diversity(moths, "brillouin") # 1.88
evenness(moths, "brillouin") # 0.83

## Simpson dominance index
## Data from Magurran 1988, p. 152-153
trees <- CountMatrix(
  data = c(752, 276, 194, 126, 121, 97, 95, 83, 72, 44, 39,
           16, 15, 13, 9, 9, 9, 8, 7, 4, 2, 2, 1, 1, 1),
  nrow = 1, byrow = TRUE
)

diversity(trees, "simpson") # 1.19
evenness(trees, "simpson") # 0.21

## McIntosh dominance index

```

```

## Data from Magurran 1988, p. 154-155
invertebrates <- CountMatrix(
  data = c(254, 153, 90, 69, 68, 58, 51, 45, 40, 39, 25, 23, 19, 18, 16, 14, 14,
           11, 11, 11, 11, 10, 6, 6, 6, 6, 5, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1),
  nrow = 1, byrow = TRUE
)

diversity(invertebrates, "mcintosh") # 0.71
evenness(invertebrates, "mcintosh") # 0.82

## Berger-Parker dominance index
## Data from Magurran 1988, p. 156-157
fishes <- CountMatrix(
  data = c(394, 3487, 275, 683, 22, 1, 0, 1, 6, 8, 1, 1, 2,
           1642, 5681, 196, 1348, 12, 0, 1, 48, 21, 1, 5, 7, 3,
           90, 320, 180, 46, 2, 0, 0, 1, 0, 0, 2, 1, 5,
           126, 17, 115, 436, 27, 0, 0, 3, 1, 0, 0, 1, 0,
           32, 0, 0, 5, 0, 0, 0, 0, 13, 9, 0, 0, 4),
  nrow = 5, byrow = TRUE,
  dimnames = list(c("station 1", "station 2", "station 3",
                    "station 4", "station 5"), NULL)
)

diversity(fishes, "berger") # 0.71 0.63 0.50 0.60 0.51

```

---

beta-diversity

*β-diversity*

---

## Description

Measures differentiation diversity. `similarity` returns a similarity matrix. `turnover` returns the degree of turnover in taxa composition along a gradient or transect.

## Usage

```
turnover(object, ...)
```

```
similarity(object, ...)
```

```
## S4 method for signature 'CountMatrix'
turnover(object, method = c("whittaker", "cody",
  "routledge1", "routledge2", "routledge3", "wilson"), simplify = FALSE,
  ...)
```

```
## S4 method for signature 'IncidenceMatrix'
turnover(object, method = c("whittaker",
  "cody", "routledge1", "routledge2", "routledge3", "wilson"),
  simplify = FALSE, ...)
```



```
## S4 method for signature 'CountMatrix'
similarity(object, method = c("brainerd", "bray",
  "jaccard", "morisita", "sorenson"), ...)

## S4 method for signature 'IncidenceMatrix'
similarity(object, method = c("jaccard",
  "sorenson"), ...)
```

### Arguments

|                       |  |
|-----------------------|--|
| <code>object</code>   | A $m \times p$ matrix of count data.   |
| <code>...</code>      | Further arguments passed to other methods.   |
| <code>method</code>   | A <a href="#">character</a> string specifying the method to be used (see details). Any unambiguous substring can be given. |
| <code>simplify</code> | A <a href="#">logical</a> scalar: should the result be simplified to a matrix? The default value, FALSE, returns a list.   |

### Value

`similarity` returns a  $m \times m$  symmetric matrix.  
 If `simplify` is FALSE, `turnover` returns a list (default), else returns a matrix.

### Turnover

The following methods can be used to ascertain the degree of *turnover* in taxa composition along a gradient ( $\beta$ -diversity) on qualitative (presence/absence) data. This assumes that the order of the matrix rows (from 1 to  $n$ ) follows the progression along the gradient/transect.

**whittaker** Whittaker measure.

**cody** Cody measure.

**routledge1** Routledge first measure.

**routledge2** Routledge second measure.

**routledge3** Routledge third measure. This is the exponential form of the second measure.

**wilson** Wilson measure.

### Similarity

$\beta$ -diversity can also be measured by addressing *similarity* between pairs of sites. This provides a scale of similarity from 0-1 where 1 is perfect similarity and 0 is no similarity (with the exception of the Brainerd-Robinson index which is scaled between 0 and 200):

**brainerd** Brainerd-Robinson quantitative index. This is a city-block metric of similarity.

**bray** Sorenson quantitative index (Bray and Curtis modified version of the Sorenson index).

**jaccard** Jaccard qualitative index.

**morisita** Morisita-Horn quantitative index.

**sorenson** Sorenson qualitative index.

**Author(s)**

N. Frerebeau

**References**

Brainerd, G. W. (1951). The Place of Chronological Ordering in Archaeological Analysis. *American Antiquity*, 16(04), 301-313. DOI: [10.2307/276979](https://doi.org/10.2307/276979).

Bray, J. R. & Curtis, J. T. (1957). An Ordination of the Upland Forest Communities of Southern Wisconsin. *Ecological Monographs*, 27(4), 325-349. DOI: [10.2307/1942268](https://doi.org/10.2307/1942268).

Cody, M. L. (1975). Towards a theory of continental species diversity: Bird distributions over Mediterranean habitat gradients. In M. L. Cody & J. M. Diamond (Eds.), *Ecology and Evolution of Communities*. Cambridge, MA: Harvard University Press, p. 214–257.

Magurran, A. E. (1988). *Ecological Diversity and its Measurement*. Princeton, NJ: Princeton University Press. DOI:[10.1007/978-94-015-7358-0](https://doi.org/10.1007/978-94-015-7358-0).

Robinson, W. S. (1951). A Method for Chronologically Ordering Archaeological Deposits. *American Antiquity*, 16(04), 293-301. DOI: [10.2307/276978](https://doi.org/10.2307/276978).

Routledge, R. D. (1977). On Whittaker's Components of Diversity. *Ecology*, 58(5), 1120-1127. DOI: [10.2307/1936932](https://doi.org/10.2307/1936932).

Whittaker, R. H. (1960). Vegetation of the Siskiyou Mountains, Oregon and California. *Ecological Monographs*, 30(3), 279-338. DOI: [10.2307/1943563](https://doi.org/10.2307/1943563).

Wilson, M. V., & Shmida, A. (1984). Measuring Beta Diversity with Presence-Absence Data. *The Journal of Ecology*, 72(3), 1055-1064. DOI: [10.2307/2259551](https://doi.org/10.2307/2259551).

**See Also**

[richness rarefaction diversity evenness](#)

**Examples**

```
# Degree of turnover
# Data from Magurran 1988, p. 162
trees <- IncidenceMatrix(
  data = c(1, 1, 1, 0, 0, 0,
           1, 1, 1, 1, 1, 1,
           0, 0, 1, 0, 1, 0,
           0, 0, 0, 1, 1, 1,
           0, 0, 0, 0, 1, 1,
           0, 0, 0, 1, 0, 1),
  nrow = 6, byrow = FALSE,
  dimnames = list(c("1", "2", "3", "4", "5", "6"),
                  c("Birch", "Oak", "Rowan", "Beech", "Hazel", "Holly")))
)

## Whittaker's measure
turnover(trees, "whittaker") # 1

## Cody's measure
turnover(trees, "cody") # 3
```

```

## Routledge's measures
turnover(trees, "routledge1") # 0.29
turnover(trees, "routledge2") # 0.56
turnover(trees, "routledge3") # 1.75

## Wilson and Shmida's measure
turnover(trees, "wilson") # 1

# Similarity measures
# Data from Magurran 1988, p. 166
birds <- CountMatrix(
  data = c(1.4, 4.3, 2.9, 8.6, 4.2, 15.7, 2.0, 50, 1, 11.4, 11.4, 4.3, 13.0,
           14.3, 8.6, 7.1, 10.0, 1.4, 2.9, 5.7, 1.4, 11.4, 2.9, 4.3, 1.4, 2.9,
           0, 0, 0, 2.9, 0, 0, 0, 10, 0, 0, 5.7, 2.5, 5.7, 8.6, 5.7, 2.9, 0, 0,
           2.9, 0, 0, 5.7, 0, 2.9, 0, 2.9) * 10,
  nrow = 2, byrow = TRUE, dimnames = list(c("unmanaged", "managed"), NULL)
)

## Jaccard measure
## (presence/absence data)
similarity(birds, "jaccard") # 0.46

## Sorenson measure
## (presence/absence data)
similarity(birds, "sorenson") # 0.63

## Jaccard measure (Bray's formula)
## (count data)
similarity(birds, "bray") # 0.44

## Morisita-Horn measure
## (count data)
similarity(birds, "morisita") # 0.81

```

---

BootCA-class

*Partial bootstrap CA*

---

## Description

An S4 class to store partial bootstrap correspondance analysis results.

## Usage

```

## S4 method for signature 'BootCA'
x[[i]]

## S4 method for signature 'BootCA'
rows(x)

```

```
## S4 method for signature 'BootCA'  
columns(x)
```

### Arguments

x A BootCA object from which to extract element(s).  
i A **character** string specifying elements to extract. Character vectors will be matched to the name of the slots.

### Methods (by generic)

- `[]`: Returns information about the individual slots.
- `rows`: Returns the convex hull vertice coordinates for each individual.
- `columns`: Returns the convex hull vertice coordinates for each variable.

### Slots

`rows` A three columns **data.frame** giving the vertices coordinates of the samples convex hull and a identifier to link each row to a sample.

`columns` A three columns **data.frame** giving the vertices coordinates of the variables convex hull and a identifier to link each row to a variable.

`lengths` A **numeric** vector giving the convex hull maximum dimension length of each sample.

`cutoff` A length-one **numeric** vector giving the cutoff value for sample selection.

`keep` A named **numeric** vector giving the subscript of the samples to be kept.

### Author(s)

N. Frerebeau

---

boves

*Boves ceramics*

---

### Description

A dataset containing the ceramic counts from the castle site of Boves (Somme, France). The data are grouped into eight periods ranging from the 10th to the 18th century and thirteen ceramic types.

### Usage

boves

**Format**

A numeric matrix with 8 rows and 13 variables:

**I**

**IIa**

**IIb**

**IIIa**

**IIIb**

**IIIc**

**IVa**

**IVb**

**Va**

**Vb**

**VI**

**VII**

**Source**

Racinet P. (2002). Le site castral et prioral de Boves du Xe au XVIIe siècle. Bilan des recherches 1996-2000. *Revue archéologique de Picardie*. Numéro spécial 20, 123 p.

---

compiegne

*Compiegne ceramics*

---

**Description**

A dataset containing the ceramic counts from the Place des Hallettes in Compiègne (Oise, France). The data are grouped into five periods of about a century, ranging from the 9th to the 14th century, and sixteen ceramic types.

**Usage**

compiegne

**Format**

A numeric matrix with 5 rows (chronological periods, numbered from the oldest to the most recent from 1 to 5) and 16 variables (ceramic types):

**A** Red to white ceramics with fine sized inclusions.

**B** Red to white ceramics with medium sized inclusions.

**C** Dark ceramics with fine sized inclusions.

**D** Dark ceramics with medium sized inclusions.

- E** Ceramics close to those of groups B or D, with similarities to group F.
- F** Black, red or beige ceramics with coarse inclusions.
- G** Red polished ceramics with fine to medium sized inclusions.
- H** Black polished ceramics with fine sized inclusions.
- I** Black polished ceramics with medium sized inclusions.
- J** Polished and painted ceramics with fine to medium sized inclusions.
- K** Painted ceramics, similar to those of group A.
- L** Painted ceramics, similar to those of group B.
- M** Painted ceramics with coarse inclusions.
- N** Glazed ceramics.
- O** Stamped ceramics.
- P** Coated ceramics.

### Source

Lacroix, M. C. (1997). La céramique médiévale du site des Hallettes à Compiègne (Oise). *Revue archéologique de Picardie*. Numéro spécial, 13(1), 135-168. DOI: [10.3406/pica.1997.1945](https://doi.org/10.3406/pica.1997.1945)

---

|                   |                     |
|-------------------|---------------------|
| CountMatrix-class | <i>Count matrix</i> |
|-------------------|---------------------|

---

### Description

An S4 class to represent a count matrix.

### Usage

```
CountMatrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,
            dimnames = NULL)
```

### Arguments

- |          |  |
|----------|--|
| data     | an optional data vector (including a list or <a href="#">expression</a> vector). Non-atomic classed R objects are coerced by <a href="#">as.vector</a> and all attributes discarded.   |
| nrow     | the desired number of rows.  |
| ncol     | the desired number of columns.   |
| byrow    | logical. If FALSE (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.   |
| dimnames | A <a href="#">dimnames</a> attribute for the matrix: NULL or a list of length 2 giving the row and column names respectively. An empty list is treated as NULL, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions. |

**Details**

Numeric values are coerced to [integer](#) as by [as.integer](#) (and hence truncated towards zero).

**Note**

This class extends the base [matrix](#).

**Author(s)**

N. Frerebeau

**See Also**

[matrix](#)

Other abundance matrix: [FrequencyMatrix-class](#)

**Examples**

```
# Create a count data matrix
A1 <- CountMatrix(data = sample(0:10, 100, TRUE),
                  nrow = 10, ncol = 10, byrow = TRUE)

# Coerce counts to frequencies
B <- as(A1, "FrequencyMatrix")

# Row sums are internally stored before coercing to a frequency matrix
totals(B) # Get row sums

# This allows to restore the source data
A2 <- as(B, "CountMatrix")
all(A1 == A2)
```

---

FrequencyMatrix-class *Frequency matrix*

---

**Description**

An S4 class to represent a frequency matrix.

**Usage**

```
## S4 method for signature 'FrequencyMatrix'
totals(x)
```

**Arguments**

x                    A FrequencyMatrix object from which to extract element.

## Details

To ensure data integrity, a FrequencyMatrix can only be created by coercion from a [CountMatrix](#) (see examples).

## Methods (by generic)

- `totals`: Returns the row sums (counts).

## Slots

`total` A [numeric](#) vector.

## Note

This class extends the base [matrix](#).

## Author(s)

N. Frerebeau

## See Also

[matrix](#)

Other abundance matrix: [CountMatrix-class](#)

## Examples

```
# Create a count data matrix
A1 <- CountMatrix(data = sample(0:10, 100, TRUE),
                  nrow = 10, ncol = 10, byrow = TRUE)

# Coerce counts to frequencies
B <- as(A1, "FrequencyMatrix")

# Row sums are internally stored before coercing to a frequency matrix
totals(B) # Get row sums

# This allows to restore the source data
A2 <- as(B, "CountMatrix")
all(A1 == A2)
```



---

IncidenceMatrix-class *Incidence matrix*

---

### Description

An S4 class to represent an incidence (presence/absence) matrix.

### Usage

```
IncidenceMatrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,  
  dimnames = NULL)
```

### Arguments

|          |  |
|----------|--|
| data     | an optional data vector (including a list or <a href="#">expression</a> vector). Non-atomic classed R objects are coerced by <a href="#">as.vector</a> and all attributes discarded.   |
| nrow     | the desired number of rows.  |
| ncol     | the desired number of columns.   |
| byrow    | logical. If FALSE (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.   |
| dimnames | A <a href="#">dimnames</a> attribute for the matrix: NULL or a list of length 2 giving the row and column names respectively. An empty list is treated as NULL, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions. |

### Details

Numeric values are coerced to [logical](#) as by [as.logical](#).

### Note

This class extends the base [matrix](#).

### Author(s)

N. Frerebeau

### See Also

[matrix](#)

Other logical matrix: [OccurrenceMatrix-class](#)

**Examples**

```
# Create an incidence (presence/absence) matrix
# Data will be coerced with as.logical()
C1 <- IncidenceMatrix(data = sample(0:1, 100, TRUE),
                      nrow = 10, ncol = 10)

# Coerce to a co-occurrence matrix
D <- as(C1, "OccurrenceMatrix")

# Create a count data matrix
A <- CountMatrix(data = sample(0:10, 100, TRUE),
                 nrow = 10, ncol = 10, byrow = TRUE)

# Coerce to presence/absence
C2 <- as(A, "IncidenceMatrix")
```

---

mississippi

*Mississippi ceramics*

---

**Description**

A dataset containing ceramic counts from the Mississippi region.

**Usage**

mississippi

**Format**

A numeric matrix with 20 rows and 10 variables:

**ParkinPunctate**

**BartonKentMPI**

**Painted**

**FortuneNoded**

**RanchIncised**

**WallsEngraved**

**WallaceIncised**

**RhodesIncised**

**VernonPaulApplique**

**HullEngraved**

**Source**

Lipo, C. P., Madsen, M. E. & Dunnell, R. C. (2015). A Theoretically-Sufficient and Computationally-Practical Technique for Deterministic Frequency Seriation. *PLOS ONE*, 10(4), e0124942. DOI: [10.1371/journal.pone.0124942](https://doi.org/10.1371/journal.pone.0124942).

---

OccurrenceMatrix-class

*Co-occurrence matrix*

---

### Description

An S4 class to represent a co-occurrence matrix.

### Details

A co-occurrence matrix is a symmetric matrix with zeros on its main diagonal, which works out which pairs of taxa occur together in at least one sample

### Note

This class extends the base [matrix](#).

### Author(s)

N. Frerebeau

### See Also

[matrix](#)

Other logical matrix: [IncidenceMatrix-class](#)

### Examples

```
# Create an incidence (presence/absence) matrix
# Data will be coerced with as.logical()
C1 <- IncidenceMatrix(data = sample(0:1, 100, TRUE),
                      nrow = 10, ncol = 10)

# Coerce to a co-occurrence matrix
D <- as(C1, "OccurrenceMatrix")

# Create a count data matrix
A <- CountMatrix(data = sample(0:10, 100, TRUE),
                 nrow = 10, ncol = 10, byrow = TRUE)

# Coerce to presence/absence
C2 <- as(A, "IncidenceMatrix")
```

---

PermutationOrder-class

*Permutation order*

---

### Description

An S4 class to represent a permutation order.

### Usage

```
## S4 method for signature 'PermutationOrder'  
x[[i]]
```

```
## S4 method for signature 'PermutationOrder'  
rows(x)
```

```
## S4 method for signature 'PermutationOrder'  
columns(x)
```

```
## S4 method for signature 'PermutationOrder'  
method(x)
```

### Arguments

|   |  |
|---|--|
| x | A PermutationOrder object from which to extract element(s).  |
| i | A <a href="#">character</a> string specifying elements to extract. Character vectors will be matched to the name of the slots. |

### Methods (by generic)

- `[[`: Returns information about the individual slots.
- `rows`: Returns the rows permutation.
- `columns`: Returns the columns permutation.
- `method`: Returns the method used for seriation.

### Slots

`rows` A [integer](#) vector giving the rows permutation.

`columns` A [integer](#) vector giving the columns permutation.

`seriation` A [character](#) vector indicating the seriation method used.

### Author(s)

N. Frerebeau

---

|         |                 |
|---------|-----------------|
| plotBar | <i>Bar plot</i> |
|---------|-----------------|

---

### Description

Plots a Bertin or a Ford (battleship curve) diagram.

### Usage

```
plotBar(object, ...)  
  
## S4 method for signature 'CountMatrix'  
plotBar(object, level = 0.05, EPPM = FALSE,  
        center = TRUE, horizontal = FALSE)  
  
## S4 method for signature 'FrequencyMatrix'  
plotBar(object, level = 0.05, EPPM = FALSE,  
        center = TRUE, horizontal = FALSE)
```

### Arguments

|            |  |
|------------|--|
| object     | An object to be plotted.   |
| ...        | Further arguments passed to other methods.   |
| level      | A length-one <a href="#">numeric</a> vector giving the confidence level to be drawn.   |
| EPPM       | A <a href="#">logical</a> scalar: should the EPPM be drawn (see details)?  |
| center     | A <a href="#">logical</a> scalar: should the bar plot be centered? The default, TRUE, produces a Ford diagram, otherwise it produces a Bertin diagram.                                     |
| horizontal | A <a href="#">logical</a> scalar: should the bar plot be horizontal? The default, FALSE, means variables in rows and cases in columns (i.e. Bertin diagram). Only used if center is FALSE. |

### Details

If EPPM is TRUE and if a relative abundance is greater than the mean percentage of the type, the exceeding part is highlighted. This positive difference from the column mean percentage (in french "écart positif au pourcentage moyen", EPPM) represents a deviation from the situation of statistical independence. As independence can be interpreted as the absence of relationships between types and the chronological order of the assemblages, EPPM is a usefull graphical tool to explore significance of relationship between rows and columns related to [seriation](#).

### Author(s)

N. Frerebeau

## References

- Bertin, J. (1977). *La graphique et le traitement graphique de l'information*. Paris: Flammarion. Nouvelle Bibliothèque Scientifique.
- Desachy, B. (2004). Le sériographe EPPM: un outil informatisé de sériation graphique pour tableaux de comptages. *Revue archéologique de Picardie*, 3(1), 39-56. DOI: [10.3406/pica.2004.2396](https://doi.org/10.3406/pica.2004.2396).
- Ford, J. A. (1962). *A quantitative method for deriving cultural chronology*. Washington, DC: Pan American Union. Technical manual 1.

## See Also

Other plot: [plotMatrix](#), [plotRank](#), [plotSpot](#)

## Examples

```
## Not run:

# Coerce dataset to abundance matrix
# Data from Desachy 2004
count <- as(compiegne, "CountMatrix")
freq <- as(boves, "FrequencyMatrix")

# Plot Ford diagram...
## ...without threshold
plotBar(count)
plotBar(freq)

## ...with EPPM (i.e. Bruno Desachy's sériographe)
plotBar(count, EPPM = TRUE)
plotBar(freq, EPPM = TRUE)

# Plot Bertin diagram...
## ...without EPPM...
plotBar(count, center = FALSE, horizontal = TRUE)
## ...and confidence intervals
plotBar(freq, level = FALSE, center = FALSE, horizontal = FALSE)

## ...with EPPM (i.e. Bruno Desachy's sériographe)
plotBar(count, EPPM = TRUE, center = FALSE, horizontal = TRUE)

## End(Not run)
```

---

plotMatrix

*Matrix plot*

---

## Description

Plots a heatmap.

**Usage**

```
plotMatrix(object, ...)

## S4 method for signature 'CountMatrix'
plotMatrix(object, PVI = FALSE)

## S4 method for signature 'FrequencyMatrix'
plotMatrix(object, PVI = FALSE)

## S4 method for signature 'IncidenceMatrix'
plotMatrix(object)
```

**Arguments**

|        |   |
|--------|---|
| object | An object to be plotted.  |
| ...    | Further arguments passed to other methods.  |
| PVI    | A <a href="#">logical</a> scalar: should the PVI be drawn instead of frequencies (see details)? |

**Details**

If PVI is FALSE, it plots a heatmap of relative abundances (frequency), otherwise percentages of the independence value are drawn (in french, "pourcentages de valeur d'indépendance", PVI).

PVI is calculated for Each cell as the percentage to the column theoretical independence value: PVI greater than 1 represent positive deviations from the independence, whereas PVI smaller than 1 represent negative deviations (Desachy 2004).

The PVI matrix allows to explore deviations from independence (an intuitive graphical approach to  $\chi^2$ ), in such a way that a high-contrast matrix has quite significant deviations, with a low risk of being due to randomness (Desachy 2004).

**Author(s)**

N. Frerebeau

**References**

Desachy, B. (2004). Le sériographe EPPM: un outil informatisé de sériation graphique pour tableaux de comptages. *Revue archéologique de Picardie*, 3(1), 39-56. DOI: [10.3406/pica.2004.2396](https://doi.org/10.3406/pica.2004.2396).

**See Also**

Other plot: [plotBar](#), [plotRank](#), [plotSpot](#)

**Examples**

```
# Abundance data
## Coerce dataset to abundance matrix
## Data from Desachy 2004
count <- as(compiegne, "CountMatrix")
```

```

# Plot matrix diagram...
## ...without threshold (i.e. heatmap)
plotMatrix(count)

## ...with PVI as threshold (i.e. Bruno Desachy's matrigraphe)
plotMatrix(count, PVI = TRUE) +
  ggplot2::scale_fill_gradient2(midpoint = 1)

# Presence/absence data
bin <- IncidenceMatrix(data = sample(0:1, size = 100, replace = TRUE),
  nrow = 10, ncol = 10)

# Plot matrix diagram
plotMatrix(bin) +
  ggplot2::scale_fill_manual(values = c("TRUE" = "black", "FALSE" = "white"))

```

---

plotRank

*Rank vs abundance plot*


---

## Description

Plots a rank vs relative abundance diagram.

## Usage

```

plotRank(object, ...)

## S4 method for signature 'CountMatrix'
plotRank(object, log = NULL, facet = TRUE)

## S4 method for signature 'FrequencyMatrix'
plotRank(object, log = NULL, facet = TRUE)

```

## Arguments

|        |   |
|--------|---|
| object | An object to be plotted.  |
| ...    | Further arguments passed to other methods.  |
| log    | A <a href="#">character</a> string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic (base 10). |
| facet  | A <a href="#">logical</a> scalar: should a matrix of panels defined by case/sample be drawn?  |

## Details

Note that rows are scaled to 0-1 (frequencies).



**Author(s)**

N. Frerebeau

**See Also**Other plot: [plotBar](#), [plotMatrix](#), [plotSpot](#)**Examples**

```
# Coerce dataset to abundance matrix
# Data from Desachy 2004
count <- as(compiegne, "CountMatrix")
freq <- as(boves, "FrequencyMatrix")

# Plot rank vs abundance
plotRank(count)
plotRank(count, log = "xy")

# Plot rank vs abundance
plotRank(freq, facet = FALSE)
plotRank(freq, facet = FALSE, log = "xy")
```

---

plotSpot

*Spot plot*

---

**Description**

Plots a spot matrix.

**Usage**

```
plotSpot(object, ...)

## S4 method for signature 'CountMatrix'
plotSpot(object, threshold = NULL)

## S4 method for signature 'FrequencyMatrix'
plotSpot(object, threshold = NULL)
```

**Arguments**

|           |   |
|-----------|---|
| object    | An object to be plotted.  |
| ...       | Further arguments passed to other methods.  |
| threshold | A <a href="#">function</a> that takes a numeric vector as argument and returns a single numeric value (see details). If NULL, no threshold is computed. |

**Details**

Note that rows are scaled to 0-1 (frequencies).

**Note**

Adapted from Dan Gopstein's original [spot matrix](#). Credit should be given to him.

**Author(s)**

N. Frerebeau

**See Also**

Other plot: [plotBar](#), [plotMatrix](#), [plotRank](#)

**Examples**

```
# Coerce dataset to abundance matrix
# Data from Desachy 2004
count <- as(compiegne, "CountMatrix")

# Plot spot diagram...
## ...without threshold
plotSpot(count)

## ...with the column means as threshold
plotSpot(count, threshold = mean)

## ...with the column medians as threshold
plotSpot(count, threshold = median)
```

---

seriation

*Matrix seriation*

---

**Description**

`refine` performs a partial bootstrap correspondance analysis seriation refinement.

`seriate` computes a permutation order for rows and/or columns.

`permute` rearranges a data matrix according to a permutation order.

**Usage**

```
refine(object, ...)
```

```
seriate(object, ...)
```

```
permute(object, order, ...)
```

```
## S4 method for signature 'CountMatrix'
refine(object, cutoff, n = 1000, axes = c(1,
  2), ...)

## S4 method for signature 'CountMatrix'
seriate(object, method = c("correspondance",
  "reciprocal"), EPPM = FALSE, margin = c(1, 2), stop = 100, ...)

## S4 method for signature 'IncidenceMatrix'
seriate(object, method = c("correspondance",
  "reciprocal"), margin = c(1, 2), stop = 100, ...)

## S4 method for signature 'CountMatrix,PermutationOrder'
permute(object, order)

## S4 method for signature 'IncidenceMatrix,PermutationOrder'
permute(object, order)
```

## Arguments

|        |  |
|--------|--|
| object | An $m \times p$ data matrix.   |
| ...    | Further arguments passed to other methods.   |
| order  | An object giving the permutation order for rows and columns.   |
| cutoff | A function that takes a numeric vector as argument and returns a single numeric value (see details).   |
| n      | A non-negative <a href="#">integer</a> giving the number of partial bootstrap replications (see details).  |
| axes   | A <a href="#">numeric</a> vector giving the subscripts of the CA axes to use (see details).  |
| method | A <a href="#">character</a> string specifying the method to be used. This must be one of "reciprocal", "correspondance" (see details). Any unambiguous substring can be given.   |
| EPPM   | A <a href="#">logical</a> scalar: should the seriation be computed on EPPM instead of raw data?  |
| margin | A <a href="#">numeric</a> vector giving the subscripts which the rearrangement will be applied over. E.g., for a matrix 1 indicates rows, 2 indicates columns, c(1, 2) indicates rows then columns, c(2, 1) indicates columns then rows. |
| stop   | A length-one <a href="#">numeric</a> vector giving the stopping rule (i.e. maximum number of iterations) to avoid infinite loop.   |

## Details

The matrix seriation problem in archaeology is based on three conditions and two assumptions, which Dunell (1970) summarizes as follows.

The homogeneity conditions state that all the groups included in a seriation must:

1. Be of comparable duration.

2. Belong to the same cultural tradition.
3. Come from the same local area.

The mathematical assumptions state that the distribution of any historical or temporal class:

1. Is continuous through time.
2. Exhibits the form of a unimodal curve.

These assumptions create a distributional model and ordering is accomplished by arranging the matrix so that the class distributions approximate the required pattern. The resulting order is inferred to be chronological.

The following seriation methods are available:

**correspondance** Correspondance analysis-based seriation. Correspondance analysis (CA) is an effective method for the seriation of archaeological assemblages. The order of the rows and columns is given by the coordinates along one dimension of the CA space, assumed to account for temporal variation. The direction of temporal change within the correspondance analysis space is arbitrary: additional information is needed to determine the actual order in time.

**reciprocal** Reciprocal ranking (incidence data) or averaging (frequency data) seriation. These procedures iteratively rearrange rows and/or columns according to their weighted rank in the data matrix until convergence. Note that this procedure could enter into an infinite loop. If no convergence is reached before the maximum number of iterations, it stops with a warning.

`refine` allows to identify samples that are subject to sampling error or samples that have underlying structural relationships and might be influencing the ordering along the CA space. This relies on a partial bootstrap approach to CA-based seriation where each sample is replicated  $n$  times. The maximum dimension length of the convex hull around the sample point cloud allows to remove samples for a given cutoff value.

According to Peebles and Schachner (2012), "[this] point removal procedure [results in] a reduced dataset where the position of individuals within the CA are highly stable and which produces an ordering consistent with the assumptions of frequency seriation."

`refine` returns the subscript of samples to be kept (i.e. samples with maximum dimension length of the convex hull smaller than the cutoff value).

## Value

`refine` returns a [BootCA](#) object.

`seriate` returns a [PermutationOrder](#) object.

`permute` returns either a [CountMatrix](#), [FrequencyMatrix](#) or [IncidenceMatrix](#) (the same as object).

## Author(s)

N. Frerebeau

## References

- Desachy, B. (2004). Le sériographe EPPM: un outil informatisé de sériation graphique pour tableaux de comptages. *Revue archéologique de Picardie*, 3(1), 39-56. DOI: [10.3406/pica.2004.2396](https://doi.org/10.3406/pica.2004.2396).
- Dunnell, R. C. (1970). Seriation Method and Its Evaluation. *American Antiquity*, 35(03), 305-319. DOI: [10.2307/278341](https://doi.org/10.2307/278341).
- Ihm, P. (2005). A Contribution to the History of Seriation in Archaeology. In C. Weihs & W. Gaul (Eds.), *Classification: The Ubiquitous Challenge* (p. 307-316). Berlin Heidelberg: Springer. DOI: [10.1007/3-540-28084-7\\_34](https://doi.org/10.1007/3-540-28084-7_34).
- Peeples, M. A., & Schachner, G. (2012). Refining correspondence analysis-based ceramic seriation of regional data sets. *Journal of Archaeological Science*, 39(8), 2818-2827. DOI: [10.1016/j.jas.2012.04.040](https://doi.org/10.1016/j.jas.2012.04.040).

## See Also

[CA](#)

## Examples

```
## Not run:

# Refine matrix seriation (this is a long running example)
# Reproduces Peeples and Schachner 2012 results
count <- as(zuni, "CountMatrix")

## Samples with convex hull maximum dimension length greater than the cutoff
## value will be marked for removal.
## Define cutoff as one standard deviation above the mean
fun <- function(x) { mean(x) + sd(x) }

## Get indices of samples to be kept
## Warning: this may take a few seconds!
refined <- refine(count, cutoff = fun)
refined[["keep"]]

# Matrix seriation
# Reproduces Desachy 2004 results
## Coerce dataset to abundance matrix
count <- as(compiegne, "CountMatrix")

## Plot new matrix
plotBar(count, EPPM = TRUE)

## Get seriation order for columns on EPPM using the reciprocal averaging method
## Expected column order: N, A, C, K, P, L, B, E, I, M, D, G, O, J, F, H
indices <- seriate(count, method = "reciprocal", EPPM = TRUE, margin = 2)

## Permute columns
new <- permute(count, indices)

## Plot new matrix
plotBar(new, EPPM = TRUE)
```

## End(Not run)

---

zuni

*Zuni ceramics*

---

### Description

A dataset containing ceramic counts from the Zuni region of the American Southwest.

### Usage

zuni

### Format

A numeric matrix with 420 rows and 18 variables:

**LINO**

**KIAT**

**RED**

**GALL**

**ESC**

**PUBW**

**RES**

**TULA**

**PINE**

**PUBR**

**WING**

**WIPO**

**SJ**

**LSJ**

**SPR**

**PINER**

**HESH**

**KWAK**

### Source

Peebles, M. A., & Schachner, G. (2012). Refining correspondence analysis-based ceramic seriation of regional data sets. *Journal of Archaeological Science*, 39(8), 2818-2827. DOI: [10.1016/j.jas.2012.04.040](https://doi.org/10.1016/j.jas.2012.04.040).

# Index

## \*Topic **datasets**

- boves, [12](#)
  - compiegne, [13](#)
  - mississippi, [18](#)
  - zuni, [30](#)
- `[[`, `BootCA`-method (`BootCA`-class), [11](#)  
`[[`, `PermutationOrder`-method (`PermutationOrder`-class), [20](#)
- accessors, [3](#)  
alpha-diversity, [4](#)  
as.integer, [15](#)  
as.logical, [17](#)  
as.vector, [14](#), [17](#)
- beta-diversity, [8](#)  
`BootCA`, [28](#)  
`BootCA`-class, [11](#)  
boves, [12](#)
- `CA`, [29](#)  
character, [4](#), [9](#), [12](#), [20](#), [24](#), [27](#)  
columns (accessors), [3](#)  
columns, `BootCA`-method (`BootCA`-class), [11](#)  
columns, `PermutationOrder`-method (`PermutationOrder`-class), [20](#)  
compiegne, [13](#)  
`CountMatrix`, [16](#), [28](#)  
`CountMatrix` (`CountMatrix`-class), [14](#)  
`CountMatrix`-class, [14](#)
- data.frame, [12](#)  
dimnames, [14](#), [17](#)  
diversity, [10](#)  
diversity (alpha-diversity), [4](#)  
diversity, `CountMatrix`-method (alpha-diversity), [4](#)  
diversity-method (alpha-diversity), [4](#)
- evenness, [10](#)  
evenness (alpha-diversity), [4](#)
- evenness, `CountMatrix`-method (alpha-diversity), [4](#)  
evenness-method (alpha-diversity), [4](#)  
expression, [14](#), [17](#)
- `FrequencyMatrix`, [28](#)  
`FrequencyMatrix`-class, [15](#)  
function, [25](#)
- `IncidenceMatrix`, [28](#)  
`IncidenceMatrix` (`IncidenceMatrix`-class), [17](#)  
`IncidenceMatrix`-class, [17](#)  
integer, [15](#), [20](#), [27](#)
- logical, [4](#), [9](#), [17](#), [21](#), [23](#), [24](#), [27](#)
- matrigraphe (`plotMatrix`), [22](#)  
matrix, [15–17](#), [19](#)  
method (accessors), [3](#)  
method, `PermutationOrder`-method (`PermutationOrder`-class), [20](#)  
mississippi, [18](#)
- numeric, [4](#), [12](#), [16](#), [21](#), [27](#)
- `OccurrenceMatrix`-class, [19](#)
- `PermutationOrder`, [28](#)  
`PermutationOrder`-class, [20](#)  
permute (seriation), [26](#)  
permute, `CountMatrix`, `PermutationOrder`-method (seriation), [26](#)  
permute, `IncidenceMatrix`, `PermutationOrder`-method (seriation), [26](#)  
permute-method (seriation), [26](#)  
`plotBar`, [21](#), [23](#), [25](#), [26](#)  
`plotBar`, `CountMatrix`-method (`plotBar`), [21](#)  
`plotBar`, `FrequencyMatrix`-method (`plotBar`), [21](#)  
`plotBar`-method (`plotBar`), [21](#)

- plotMatrix, [22](#), [22](#), [25](#), [26](#)
- plotMatrix, CountMatrix-method  
(plotMatrix), [22](#)
- plotMatrix, FrequencyMatrix-method  
(plotMatrix), [22](#)
- plotMatrix, IncidenceMatrix-method  
(plotMatrix), [22](#)
- plotMatrix-method (plotMatrix), [22](#)
- plotRank, [22](#), [23](#), [24](#), [26](#)
- plotRank, CountMatrix-method (plotRank),  
[24](#)
- plotRank, FrequencyMatrix-method  
(plotRank), [24](#)
- plotRank-method (plotRank), [24](#)
- plotSpot, [22](#), [23](#), [25](#), [25](#)
- plotSpot, CountMatrix-method (plotSpot),  
[25](#)
- plotSpot, FrequencyMatrix-method  
(plotSpot), [25](#)
- plotSpot-method (plotSpot), [25](#)
  
- rarefaction, [10](#)
- rarefaction (alpha-diversity), [4](#)
- rarefaction, CountMatrix-method  
(alpha-diversity), [4](#)
- rarefaction-method (alpha-diversity), [4](#)
- refine (seriation), [26](#)
- refine, CountMatrix-method (seriation),  
[26](#)
- refine-method (seriation), [26](#)
- richness, [10](#)
- richness (alpha-diversity), [4](#)
- richness, CountMatrix-method  
(alpha-diversity), [4](#)
- richness-method (alpha-diversity), [4](#)
- rows (accessors), [3](#)
- rows, BootCA-method (BootCA-class), [11](#)
- rows, PermutationOrder-method  
(PermutationOrder-class), [20](#)
  
- seriate (seriation), [26](#)
- seriate, CountMatrix-method (seriation),  
[26](#)
- seriate, IncidenceMatrix-method  
(seriation), [26](#)
- seriate-method (seriation), [26](#)
- seriation, [21](#), [26](#)
- seriographie (plotBar), [21](#)
- similarity, [7](#)
- similarity (beta-diversity), [8](#)
- similarity, CountMatrix-method  
(beta-diversity), [8](#)
- similarity, IncidenceMatrix-method  
(beta-diversity), [8](#)
- similarity-method (beta-diversity), [8](#)
  
- tabula (tabula-package), [2](#)
- tabula-package, [2](#)
- totals (accessors), [3](#)
- totals, FrequencyMatrix-method  
(FrequencyMatrix-class), [15](#)
- turnover, [7](#)
- turnover (beta-diversity), [8](#)
- turnover, CountMatrix-method  
(beta-diversity), [8](#)
- turnover, IncidenceMatrix-method  
(beta-diversity), [8](#)
- turnover-method (beta-diversity), [8](#)
  
- zuni, [30](#)