

Package ‘mefa’

November 15, 2009

Type Package

Title Multivariate Data Handling in Ecology and Biogeography

Version 3.1-4

Date 2009-11-10

Author Peter Solymos

Maintainer Peter Solymos <solymos@ualberta.ca>

Description A framework package aimed to provide standardized computational environment for specialist work via object classes to represent the data coded by samples, taxa and segments (i.e. subpopulations, repeated measures). It supports easy processing of the data along with cross tabulation and relational data tables for samples and taxa. An object of class ‘mefa’ is a project specific compendium of the data and can be easily used in further analyses. Methods are provided for extraction, aggregation, conversion, plotting, summary and reporting of ‘mefa’ objects. Reports can be generated in plain text or LaTeX format. Vignette contains worked examples.

Suggests MASS, ade4, vegan

License GPL-2

URL <http://mefa.r-forge.r-project.org/>

LazyLoad yes

LazyData true

Repository CRAN

Repository/R-Forge/Project mefa

Repository/R-Forge/Revision 488

Date/Publication 2009-11-15 13:51:19

R topics documented:

aggregate.mefa	2
as.data.frame.dist	4
as.mefa	6
dolina	8
fill.na	9
label	10
mat2list	11
mefa	13
mefaCheck	17
mefaCrosstab	18
mefadocs	19
mefalogo	20
mefaNestedless	21
mefaTables	22
mefaweb	23
melt	24
plot.mefa	26
qvector	29
rep.data.frame	30
report.mefa	31
stcs	34

Index	37
--------------	-----------

aggregate.mefa *Extract and Aggregate Objects of Class 'mefa'*

Description

These methods are useful when extracting or aggregating 'mefa' objects. The count data matrix, the matrices for segments and the linked tables of samples and taxa are extracted or aggregated at the same time.

Usage

```
## S3 method for class 'mefa':
aggregate(x, by.samp = NULL, by.taxa = NULL, ...)
## S3 method for class 'mefa':
x[i = 1:dim(x)[1], j = 1:dim(x)[2], k = 1:dim(x)[3], drop = FALSE]
```

Arguments

`x` an object of class 'mefa'.
`by.samp` a vector with length as the number of samples and more than one unique values to use aggregating samples.

<code>by.taxa</code>	a vector with length as the number of taxa and more than one unique values to use aggregating taxa.
<code>i, j, k</code>	index value or a vector of indices for samples, taxa and segments, respectively. Can be numeric or character. Negative numeric values indicate items to be excluded.
<code>drop</code>	logical, if TRUE the unused factor levels in the linked tables for samples and species will be dropped after extraction. If FALSE (default) unused factor levels are left intact.
<code>...</code>	other arguments passed to the generic function.

Details

If a 'mefa' object is aggregated, either or both tables for samples and taxa (depending on if the aggregation was made by rows, columns or both) are set to `NULL`. Because aggregation is not straightforward for data frames with mixed type of data (e.g. factors, character vectors), and the applied function (mean, sum, etc.) depends on the type of the numeric variables. Consequently, if new data table is needed, that must be defined again by the function `mefa`. If the original data frame contained only numeric columns, it can be easily aggregated if the function to apply is known. For factors and numeric columns, however, such aggregation is not defined.

The ordering of dimensions follow the ordering of the indices in the extraction call, except for nested subsets, when sequence of nested segments is not allowed to be affected by the extraction (error message is given). The extracting method retains samples and taxa tables as well. If non-nested segments are extracted, the `x$xtab` matrix will contain the sum of segments, and not the original values of `x$xtab`. Thus if a resulting 'mefa' object has only one segment, it will be identical to the `x$xtab` matrix. If nested segments are extracted, `x$xtab` will be identical to the last segment and not the sum of segments. See the examples for these differences. If the original object had nested segments and the first is excluded, segment names change as a result. If only one segment remains in the result, the nested attribute is turned off.

If the aggregation is done over a vector that is not part of the tables inside the 'mefa' object, it must have a class attribute to be recognised, use e.g. the `as.factor` function (see 'Examples').

The resulting 'mefa' object must contain at least 2 samples and 2 taxa.

Value

An object of class 'mefa'.

Note

The `aggregate` method replaces the function `strify`, the `[]` method replaces the function `exclmf` of the old (< 2.0) versions of the **mefa** package.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

S'olymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

S'olymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

See Also

[mefa](#)

Examples

```
data(dol.count, dol.samp, dol.taxa)
x <- mefa(stcs(dol.count), dol.samp, dol.taxa)
## These two are identical
aggregate(x, "microhab")
aggregate(x, x$samp$microhab, inside = FALSE)
## Interaction
aggregate(x, c("method", "microhab"))
## Both samples and taxa tables
aggregate(x, "microhab", "familia")
## To use a factor
aggregate(x, "microhab", as.factor(rbinom(dim(x)[2], 1, 0.5)))
## Extraction
x[1:10, ]
x[, 1:5]
x[, , "broken"]
x[1:10, 1:5]
x[1:10, 1:5, 2]
## Note factor levels
y <- x[which(x$samp$method == "time"), ]
levels(y$samp$method)
z <- x[which(x$samp$method == "time"), drop = TRUE]
levels(z$samp$method)
## Note the differences in total sums
y <- mefa(stcs(dol.count), dol.samp, dol.taxa, nested = TRUE)
x[ , , 2]
y[ , , 2]
```

as.data.frame.dist *Manipulation Tools for Class 'dist'*

Description

Converts a 'dist' object into vector and vice versa.

Usage

```
vec2dist(x, size, labels = NULL, diag = FALSE, upper = FALSE,
call = FALSE, method = NULL)
is.dist(x)
## S3 method for class 'dist':
as.data.frame(x, row.names = NULL,
optional = FALSE, dim.names = FALSE, ...)
```

Arguments

x	A vector or a 'dist' object.
size	Integer, the number of observations in the dataset (see <code>dist</code>).
labels, diag, upper, call, method	Logical, setting attributes of the distance object (see <code>dist</code>).
row.names	Row names for the resulting data frame.
optional	Argument not yet used.
dim.names	Logical, if TRUE, names stored in the <code>Labels</code> attribute of the <code>dist</code> object are returned as factors instead of row/column indices.
...	Other arguments passed to functions.

Value

`vec2dist` returns an object of class `dist`.

`is.dist` returns a logical statement about the class of the object.

`as.data.frame.dist` returns a data frame with columns `row` for row id's, `col` for column id's and `dist` for the distance values.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

Examples

```
vec2dist(1:3, 3)
vec2dist(1:3, 3, diag = TRUE)
vec2dist(1:3, 3, diag = TRUE, upper = TRUE)
vec2dist(1:3, 3, labels=letters[1:3])
x <- dist(1:5)
```

```
attr(x, "Labels") <- LETTERS[1:5]
x
is.dist(x)
as.data.frame(x)
as.data.frame(x, dim.names = TRUE)
```

as.mefa

Conversion Among Various Object Classes

Description

These functions coerce into class 'mefa' or 'stcs'; or converts 'mefa' or 'stcs' objects into a 'data.frame', 'matrix', 'array', 'list'.

Usage

```
as.stcs(x, ...)
## Default S3 method:
as.stcs(x, ...)
as.mefa(x, ...)
## Default S3 method:
as.mefa(x, samp, taxa, ...)
## S3 method for class 'array':
as.mefa(x, ...)
## S3 method for class 'list':
as.mefa(x, ...)
## S3 method for class 'mefa':
as.matrix(x, ...)
## S3 method for class 'mefa':
as.array(x, ...)
## S3 method for class 'mefa':
as.list(x, ...)
## S3 method for class 'stcs':
as.data.frame(x, ...)
## S3 method for class 'mefa':
as.data.frame(x, ..., fun, name, make.unique = FALSE)
mss(x, name, make.unique = FALSE, ...)
msr(x, name, make.unique = FALSE, ...)
mts(x, name, make.unique = FALSE, ...)
mtr(x, name, make.unique = FALSE, ...)
```

Arguments

x	an object of class 'mefa'.
samp	a data frame containing rows for samples, or NULL. See argument <code>xtab.fixed</code> and details for sample matching options.

taxa	a data frame containing rows for taxa, or NULL. See argument <code>xtab.fixed</code> and details for taxa matching options.
fun	a function to determine what to return, see details. It has no default, because the choice is not trivial.
name	optional character vector, names to return if not all possible elements are needed, see details.
make.unique	logical, useful to use <code>make.unique = TRUE</code> if names to return are non-unique, see details.
...	other arguments passed to the generic function <code>as.data.frame</code> .

Details

The usage of most of these coercion methods is trivial. Arrays and lists can be coerced into 'mefa' objects as if those were the `segm` element of it (either nested, or non nested). The reverse is true, when coercing an object of class 'mefa' into list or array, the `segm` element is used.

Coercing among object classes 'matrix' and 'mefa' is done via using the `xtab` element.

By `as.data.frame.mefa`, four kinds of data frames can be returned, depending on the function used as the `fun` argument: `mss` returns summary statistics for samples (richness and abundance) and `x$samp`; `msr` returns raw data (`x$xtab`) and `x$samp`; `mts` returns summary statistics for taxa (occurrence and abundance) and `x$taxa`; `mtr` returns raw data (`t(x$xtab)`) and `x$taxa`.

The `name` can be used if only a subset of the summary statistics, or raw data should be returned. The character vector should contain names of elements to return (see examples).

It might (rarely) occur, that names of the summary statistics, or the raw data and the column names of the sample/taxa table overlap (contains names that are common in the two set). In this case, the `make.unique = TRUE` can be used to resolve non-uniqueness and avoid the error message produced otherwise.

The functions supplied as the `fun` argument can be used separately. Although the usage of the `as.data.frame` method is more transparent.

Value

An object of class 'stcs', 'mefa', 'matrix', 'array', 'list' or 'data.frame', depending on the method used.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

See Also

[mefa](#), [stcs](#), [as.matrix](#), [as.list](#), [as.array](#), [as.data.frame](#)

Examples

```
data(dol.count, dol.samp, dol.taxa)
x <- mefa(stcs(dol.count), dol.samp, dol.taxa)
## These two are equivalent
as.data.frame(x, fun = mss)
mss(x)
## Return only two species
as.data.frame(x, fun = msr, c("iiso", "ppyg"))
## Taxa table and summary
as.data.frame(x, fun = mts)
## Taxa table and raw data transpose
as.data.frame(x, fun = mtr)
## Why is it useful?
## Instead of
glm(x$xtab[, "amin"] ~ microhab + method, data = x$samp, family = poisson)
## it is more convenient to use
glm(amin ~ microhab + method, data = msr(x), family = poisson)
```

dolina

The Dolina Dataset

Description

Data were collected in dolines (sinkholes, karstic depressions) of Northeast Hungary.

Usage

```
data(dol.count)
data(dol.samp)
data(dol.taxa)
```

Format

The `dol.count` data set is a data frame with 297 observations on the following 4 variables.

`samp` a factor with levels for samples.
`taxa` a factor with levels for the species.
`count` a numeric vector with count data.
`segm` a factor with levels for segments.

The `dol.samp` data set is a data frame with 24 observations on the following 2 variables.

`microhab` a factor with levels `dead.wood` `litter` `live.wood` `rock` for microhabitats.
`method` a factor with levels `time` `quadrat` for the sampling method applied.

The `dol.taxa` data set is a data frame with 121 observations on the following 4 variables.

`species` a factor with levels the species names.

`author` a factor with levels for authors of the species and description dates (accents are removed).

`familia` a factor with levels for family level classification.

`size` a numeric vector for major shell dimension of the adult specimen.

Details

The data sets are results of the 'DOLINA 2007' expedition (Vilisics et al. 2008). Species characteristics are based on Kerney et al. (1983). The full dataset (S'olymos & Kemencei 2008) from which this example was derived is available at the Dataverse Network ([hdl:1902.1/12060](https://doi.org/10.1155/2010/12060)) with the code to make the subsetting.

Source

S'olymos, P., Kemencei, Z. (2008) Methodological study data set of land snails from the Dolina 2007 project. Available at the Dataverse Network: [hdl:1902.1/12060](https://doi.org/10.1155/2010/12060).

S'olymos, P., Kemencei, Z., P'all-Gergely, B., Farkas, R., Vilisics, F., Hornung, E. (2009) Does shell accumulation matter in micro scale land snail surveys? *Malacologia*, **51**, 389–393.

Vilisics, F., Nagy, A., S'olymos, P., Farkas, R., Kemencei, Z., P'all-Gergely, B., Kisfali, M., Hornung, E. (2008) Data on the terrestrial isopoda fauna of the Als'o-hegy, Aggtelek National Park, Hungary. *Folia Faunistica Slovaca* **13**, 9–12. <http://zoology.fns.uniba.sk/ffs/interface/00053-Vilisics-et-al-2008.pdf>.

Kerney, M. P., Cameron, R. A. D., Jungbluth, J. H. (1983) *Die Landschnecken Nord- und Mitteleuropas*. P. Parey, Hamburg-Berlin, 384 p.

Examples

```
data(dol.count)
str(dol.count)
data(dol.samp)
str(dol.samp)
data(dol.taxa)
str(dol.taxa)
```

fill.na

Replace <NA> Values with Nearest Above Values

Description

The function replaces <NA> values with the nearest values above them in the same column.

Usage

```
fill.na(x)
```

Arguments

`x` a vector, a matrix or a data frame.

Details

This function is useful when converting a 'notebook style' data into an 'stcs' object. The 'notebook style' data means that observations are registered in a non-redundant way, e.g. when typing in a spreadsheet.

Value

A matrix or a data frame.

Note

This function is not vectorized, so can be a bit slow for large data sets.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

See Also

[stcs](#)

Examples

```
x <- data.frame(
  samp = c("s1", NA, "s2", NA, "s3", "s4"),
  taxa = c("t1", NA, "s1", "s2", "s3", NA),
  count = c(1, 2, 10, 3, 4, 1),
  segm = c("male", "female", "male", "female", "male", "male"))
x
fill.na(x)
fill.na(c(1:4, NA))
```

label	<i>Assign Label to an Object and Return It</i>
-------	--

Description

This function assigns a label to any kind of object. The label later can be retrieved and reused.

Usage

```
label(x)
label(x) <- value
```

Arguments

value	any expression.
x	any object.

Details

The label can be any expression assigned to any object. It is stored as a character attribute. Inheritance is not implemented, so by any transformation of the object into another one, the label is lost.

Value

Print out the label value.

Author(s)

Peter Solyomos, <solyomos@ualberta.ca>

References

Solyomos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solyomos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

Examples

```
x <- 1:10
x
## Assign a character
label(x) <- "x is a vector"
x
## Get back the label
label(x)
```

```
## Note the effect of deparsing
label(x) <- 1:10
label(x)
```

mat2list

Matrix Manipulation Tools

Description

Converts a matrix to a list, calculates matrix fill, or exclude rows/columns with certain row/column sums.

Usage

```
mat2list(x, MARGIN = 1)
mfill(x)
rdrop(x, cutoff = 0)
cdrop(x, cutoff = 0)
```

Arguments

<code>x</code>	A vector or matrix to be converted.
<code>MARGIN</code>	Margin in <code>c(1, 2)</code> , indicating respectively rows or columns to be listed.
<code>cutoff</code>	Numeric, exclude rows/columns with sums \leq <code>cutoff</code> .

Value

`mat2list` returns a list of length `nrow(x)` or `ncol(x)` depending on `MARGIN`. Elements are rows/columns of `x`. Dimnames are retained as names.

`mfill` returns a single number with the matrix fill value.

`rdrop` and `cdrop` return the original matrix but excluding rows (`rdrop`), or columns (`cdrop`) that has sum \leq `cutoff`. Excluded row/column numbers and specifications are stored in the "exclude" attribute.

Author(s)

Peter Solyomos, <solyomos@ualberta.ca>

References

Solyomos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solyomos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

Examples

```
x <- matrix(1:12, 3, 4)
rownames(x) <- letters[1:3]
colnames(x) <- LETTERS[1:4]
mat2list(x, 1)
mat2list(x, 2)
x[2, ] <- 0
x[, 2] <- 0
mfill(x)
mfill(x) == sum(x > 0) / prod(dim(x))
rdrop(x)
cdrop(x)
cdrop(x, cutoff = 5)
cdrop(x, cutoff = 22) # be careful
```

mefa

Create an Object of Class 'mefa'

Description

An object of class 'mefa' is a compendium of a crosstabulated taxa-by-samples (count) data, and optionally (count) data for segments, and linked tables for samples and taxa. The 'mefa' term stand for an acronym of 'metafaunistics', indicating that data processing is a critical and often time consuming step before data analysis. The 'mefa' package aims to help in this respect.

Usage

```
mefa(xtab, samp = NULL, taxa = NULL, id.samp = NULL, id.taxa = NULL,
      segment = TRUE, nested = FALSE, drop.zero = FALSE, drop.index = FALSE,
      xtab.fixed = TRUE)
## S3 method for class 'mefa':
is(x)
## S3 method for class 'mefa':
print(x, nlist = 10, ...)
## S3 method for class 'mefa':
summary(object, nlist = 10, ...)
## S3 method for class 'summary.mefa':
print(x, nlist, ...)
## S3 method for class 'mefa':
dim(x)
## S3 method for class 'mefa':
dimnames(x)
```

Arguments

`xtab` data. Can be an object of class 'stcs', or a matrix or data frame. Can be count (integer) or non-integer.

<code>samp</code>	a data frame containing rows for samples, or <code>NULL</code> . See argument <code>xtab.fixed</code> and details for sample matching options.
<code>taxa</code>	a data frame containing rows for taxa, or <code>NULL</code> . See argument <code>xtab.fixed</code> and details for taxa matching options.
<code>id.samp</code>	<code>NULL</code> if sample names for matching are the rownames of <code>samp</code> , otherwise, the index refers to the column of <code>samp</code> where the names are located.
<code>id.taxa</code>	<code>NULL</code> if taxa names for matching are the rownames of <code>taxa</code> , otherwise, the index refers to the column of <code>taxa</code> where the names are located.
<code>segment</code>	logical, whether information in the <code>segments</code> column of the 'stcs' object should be used (<code>TRUE</code> , default). It has no effect when <code>xtab</code> is matrix or data frame.
<code>nested</code>	logical, whether segments should be nested within each other from the first to the last (<code>TRUE</code>) or not (<code>FALSE</code> , default).
<code>drop.zero</code>	logical, if <code>TRUE</code> empty samples are removed from the result (can be useful before multivariate analyses). Otherwise, empty rows are left intact (<code>FALSE</code> , default).
<code>drop.index</code>	logical, whether columns of the samples and taxa tables containing the names used for matching should be removed (<code>TRUE</code>) to reduce redundancy or not (<code>FALSE</code> , default). This argument has effect only if <code>id.samp</code> and <code>id.taxa</code> are not <code>NULL</code> .
<code>xtab.fixed</code>	logical, if <code>TRUE</code> (default) the <code>samp</code> and <code>taxa</code> tables are subsetted and ordered according to the names of <code>xtab</code> . If <code>FALSE</code> , <code>xtab</code> is also subsetted according to the intersect of the names. Row and column ordering is determined by <code>xtab</code> in either cases.
<code>x, object</code>	an object of class 'mefa'.
<code>nlist</code>	number of elements in the list of segment names to print. List of segment names is truncated at 10 by default.
<code>...</code>	other arguments passed to functions.

Details

Main goals of the `mefa` function are to (1) convert a long formatted object of 'stcs' into a crosstabulated and optionally segmented 'mefa' object and (2) link this crosstabulation with tables of samples and taxa. The segments can be nested within each other. This feature can be useful, if segments represent e.g. subsequent sampling periods (years) and the the aim is to detect effects of data accumulation over time.

A `mefa` object can be viewed as a project oriented compendium of the data. It contains 3 dimensional (samples, taxa, segments) representation of the count data and data tables for samples and taxa. Segments contain information on some particular internal division of the data (sub populations, method specific partitions, repeated measures), but it has technical significance. Thus data tables for segments are not supported.

Input values may contain non-integer values as well.

If the input `xtab` is a matrix, `dimnames` are necessary if either `samp` or `taxa` tables are provided. In other cases, `dimnames` are not necessary. In this way, simple statistics can be returned for the matrix.

The print method returns basic information, instead of a long structural representation.

The summary contains statistics calculated from the 'mefa' object (species richness, number of individuals, occupancy, abundance, total sum, matrix fill, etc., see 'Value' section). The list can be recalled by e.g. `unclass(summary(x))`, elements of the list e.g. `summary(x)$s.rich`.

Methods for extracting, subsetting, plotting and reporting 'mefa' objects are discussed elsewhere (see links in 'See also' section). On how the 'mefa' objects can be used in further analyses, see 'Examples' and the vignette (`vignette("mefa")`).

When the `as.mefa` function is used with a 'mefa' object as argumentum, samples and taxa tables are used when `samp` and `taxa` arguments are missing. If the aim is to redefine these tables, those also can be given.

Value

The `mefa` function returns an object of class 'mefa'. It is a list with 5 elements:

<code>call</code>	the function call.
<code>xtab</code>	cross-tabulated count data, rows are samples, columns are taxa.
<code>segm</code>	list of matrices if segments are used, otherwise NULL.
<code>samp</code>	a data frame for sample related data (rows correspond to samples) if provided, otherwise NULL.
<code>taxa</code>	a data frame for taxa related data (rows correspond to taxa) if provided, otherwise NULL.

The `summary.mefa` function returns a list invisibly, with elements: `s.rich`: vector with species richness values for the samples, `s.abu`: vector with number of individuals values for the samples, `t.occ`: number of sites occupied by each species, `t.abu`: abundances of each species, `ntot`: total number of individuals, `mfill`: matrix fill, `nsamp`: number of samples, `ntaxa`: number of taxa, `nsegm`: number of segments, `segment`: vector of segment names, `call`: the function call from the 'mefa' object, `nested`, `drop.zero`, `xtab.fixed`: attributes of the 'mefa' object.

The method `dim` returns a vector of length 3 with values for number of samples, number of taxa and number of segments in the 'mefa' object. The third value is 1 in cases, when `s$segm` is NULL (because it is 1 undefined segment essentially identical to the matrix `x$xtab`, thus no need for a replicate).

The `dimnames` method returns a list of three character vectors for names of samples, taxa and segments. These can be NULL as well.

Note

The `mefa` function alone can be used instead of the combination of object classes `xcount` and `xorder` used in older (< 2.0) versions of the **mefa** package. Further it is also extended by previously undocumented features.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

S'olymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

S'olymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

Further methods are discussed on separate help pages: see `[.mefa` and `aggregate.mefa` for extracting and aggregating the data, `melt.mefa` for redefining segments or melting data into long format, `report.mefa` for generating report into file, and `plot.mefa` `boxplot.mefa` and `image.mefa` for graphical display.

See `as.mefa` for coercion methods.

The simple 2 dimensional wide arrangement of long formatted data can be done by the function `cast` in the **reshape** package of Hadley Wickham, by the `mama` function in the **simba** package by Gerald Jurasinski, and by the function `matrify` in the **labdsv** package of David W. Roberts. These are, however, not useful either for 3 dimensional (segmented) crosstabulation (see `table` and `xtabs` functions), or linking data tables for sample and taxa with the count data.

Examples

```
data(dol.count, dol.samp, dol.taxa)
## Input is stcs
x1 <- mefa(stcs(dol.count))
x1
## Input is matrix
x2 <- mefa(x1$xtab)
x2
## Attach data frame for samples
x3 <- mefa(stcs(dol.count), dol.samp)
x3
## Attach data frame for samples and taxa
x4 <- mefa(stcs(dol.count), dol.samp, dol.taxa)
x4
## Methods
## (chapter 'See also' provides
## links for further methods)
summary(x4)
## Descriptives inside the summary
unclass(summary(x4))
## Testing mefa objects
is(x4, "mefa")
## Removing empty samples
as.mefa(x4, drop.zero = TRUE)
## Dimensions
dim(x4)
## Dimnames
dimnames(x4)
```

```
## Simple examples how to use mefa objects in analysis
## GLM on sample abundances
mod <- glm(summary(x4)$s.abu ~ .^2,
  data = x4$samp, family = quasipoisson)
summary(mod)
## See the demo and vignette for more examples
# demo(mefa)
# vignette("mefa")
## Cluster analysis of community matrix
x5 <- aggregate(x4, c("method", "microhab"))
h <- hclust(dist(x5$xtab))
plot(h)
```

mefaCheck

Check and Compare Object(s) of Class 'mefa'

Description

The functions evaluates the internal structure of a 'mefa' object.

Usage

```
mefaCheck(x)
mefaCompare(x1, x2, strict = FALSE)
```

Arguments

`x`, `x1`, `x2` an object of class 'mefa'.

`strict` logical, if TRUE the function checks if ordering of dimnames are the same, otherwise it compares only matching.

Value

`mefaCheck` returns a list, first element is logical with TRUE indicating that all requirements are met. If this is FALSE, following elements of the list describe the encountered problems as characters. `mefaCheck` is called by the `is.mefa` function.

`mefaCompare` compares two mefa objects. If all dimnames, and the corresponding values in the `xtab` and `segm` parts are identical, the result is TRUE, and otherwise FALSE. Different ordering of dimnames allowed, the comparison is based on the numeric values determined by dimnames.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

SVolyms P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

SVolyms P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

[mefa](#), [is.mefa](#), [all.equal.list](#)

Examples

```
x <- matrix(rpois(20,1), 5, 4)
m <- mefa(x)
mefaCheck(m)
data(dol.count)
x1 <- mefa(stcs(dol.count))
x2 <- mefa(melt(x1))
mefaCompare(x1, x2)
mefaCompare(x1, x2, strict = TRUE)
```

mefaCrosstab

Cross Tabulate an Object of Class 'stcs'

Description

The function makes cross tabulation of an object of class 'stcs'. Similar to the `table` function but handles specialties of the 'stcs' object class (i.e. removes 'pseudo species' created to indicate empty samples). This is used internally by the function `mefa` to make 3 dimensional representation of the data.

Usage

```
mefaCrosstab(x, segment = FALSE, nested = FALSE, drop.zero = FALSE)
```

Arguments

<code>x</code>	an object of class 'stcs'.
<code>segment</code>	logical, if TRUE, the crosstabulation is made in 2 dimensions by the <code>samp</code> (sample) and <code>taxa</code> columns of the <code>x</code> . Otherwise, crosstabulation is made in 3 dimensions by the <code>samp</code> , <code>taxa</code> and <code>segm</code> (segment) columns.
<code>nested</code>	logical, if FALSE count values of each segments are independent, if TRUE crosstabulated matrices are summed up in a nested fashion from segment level 1 to the number of segments (it is indicated in the segment levels as well).
<code>drop.zero</code>	argument passed to the function <code>stcs</code> , indicating whether empty samples (with zero sum) should be removed prior to cross tabulation (TRUE), or not (FALSE).

Details

The cross tabulation of an object of class 'stcs' is made by the function `xtabs`, which accepts non-integer values too.

Value

It returns a matrix if `segment = FALSE` or a list with matrices for the segments if `segment = TRUE`.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

See Also

[stcs](#), [mefa](#), [rep.data.frame](#)

Examples

```
x <- data.frame(sample = LETTERS[c(1,1,2,2,3,4)],
  species = c(letters[c(5,5,5,6,7)], "zero.pseudo"),
  count = c(1,2,10,3,4,0), segment = letters[c(8,9,8,9,8,8)])
y <- stcs(x)
mefaCrosstab(y)
mefaCrosstab(y, segment = TRUE)
mefaCrosstab(y, segment = TRUE, nested = TRUE)
## Examples for non-integer data
y[1:5, 3] <- y[1:5, 3] + 0.123
mefaCrosstab(y)
mefaCrosstab(y, segment = TRUE)
```

Description

Display package documentation using `pager` or `pdfviewer` defined in `options`.

Usage

```
mefadocs(doc = c("ChangeLog", "SampleReport"))
```

Arguments

doc The name of the document (partial match, case sensitive).

Details

"ChangeLog" contains version history, "SampleReport" shows an example LaTeX document how to use [report.mefa](#) to prepare [Sweave](#) reports based on faunistic data.

Author(s)

Peter Solymos (<solymos@ualberta.ca>) adapted Jari Oksanen's [vegandocs](#) function in the **vegan** package.

Examples

```
## Not run:
mefadocs("Change")
mefadocs("Sample")

## End(Not run)
```

mefalogo

Schematic Plot of an Object of Class 'mefa'

Description

Plots a schematic representation of an object of class 'mefa', or alternatively plots a colored logo of the **mefa** package.

Usage

```
mefalogo(type = c("cont", "fill"), labels = c("str", "r"),
adj=c(0,0), scale=c(1,1), new = TRUE, cex = 1, ann=FALSE, axes=FALSE,
xlim=c(0,10)*scale[1]+adj[1], ylim=c(0,10)*scale[2]+adj[2], ...)
```

Arguments

type character, "cont" is for drawing only black contours, "fill" is for filled polygons.

labels character, "str" for labelling according to the general structure of a 'mefa' object, "r" places a big R letter in the middle (for logo).

adj numeric values to adjust the dimensions of the polygons.

scale a numeric value to rescale the dimensions of the polygons.

<code>new</code>	logical, if a new plot (TRUE) should be drawn, or just adding the polygons and labels to an existing one (FALSE).
<code>cex</code>	numeric, argument for scaling the labels.
<code>ann</code>	logical, annotations of the plot (see <code>par</code>).
<code>axes</code>	logical, drawing axes or not (see <code>par</code>).
<code>xlim, ylim</code>	extreme values for the x and y axes.
<code>...</code>	other arguments passed to the function <code>plot</code> , see <code>par</code> . Only effective if <code>new = TRUE</code> .

Details

This function was developed for depicting the general structure of the 'mefa' objects. The colored logo version can be used in presentations as well.

Author(s)

Péter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

Examples

```
mefalogo()
mefalogo("fill", "r")
```

`mefaNestedless` *Make Non-Nested 'mefa' Object*

Description

The functions makes a non-nested 'mefa' object.

Usage

```
mefaNestedless(x)
```

Arguments

`x` an objects of class 'mefa' with nested segments.

Value

Returns an object of class 'mefa', with non-nested segments. This function is internally used by the `melt.mefa` function.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

`mefa`, `is.mefa`, `melt.mefa`

Examples

```
data(dol.count)
x <- mefa(stcs(dol.count), nested = TRUE)
x
y <- mefaNestedless(x)
y
```

mefaTables

Subsetting and Ordering of Related Data Tables

Description

This function is called by the `mefa` function to subset the community data matrix and the related data frames, but can be applied more generally for subsetting and ordering.

Usage

```
mefaTables(xtab, dframe, margin, index = NULL,
drop.index = FALSE, xtab.fixed = TRUE)
```

Arguments

<code>xtab</code>	a data matrix.
<code>dframe</code>	a data frame.
<code>margin</code>	margin of the <code>xtab</code> matrix to use row (<code>margin = 1</code>) or column (<code>margin = 2</code>) names in comparison with the rownames of <code>dframe</code> .
<code>index</code>	if NULL rownames are used directly, else it can be used to set which column of <code>dframe</code> should be used as rowname and for comparison.
<code>drop.index</code>	logical, should the <code>index</code> column of <code>dframe</code> be deleted (if <code>index != NULL</code>).
<code>xtab.fixed</code>	logical, if TRUE the <code>xtab</code> matrix is left intact, and <code>dframe</code> is subsetted accordingly. If FALSE, both <code>xtab</code> and <code>dframe</code> are subsetted, and returned values are based on the the intersect of the compared names.

Value

Returns a list with elements `xtab` (matrix) and `dframe` (data frame), corresponding to the subsetted `xtab` and `dframe` data sets, respectively. Original column and row orderings in the input matrix `xtab` are preserved.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

[mefa](#)

Examples

```
x <- matrix(rpois(20,1), 5, 4)
## Note the reverse alphabetical names
rownames(x) <- letters[5:1]
x
f <- data.frame(matrix(rnorm(30), 10, 3))
rownames(f) <- letters[1:10]
f
## Reverse alphabetical names preserved
mefaTables(x, f, 1)
## Now result is the intersect
rownames(f) <- letters[3:12]
mefaTables(x, f, 1, xtab.fixed = FALSE)
```

`mefaweb`*Open 'mefa' Website*

Description

Opens the 'mefa' website in the default navigator.

Usage

```
mefaweb()
```

Details

The website contains resources for the 'mefa' package, please check out for more. Thank you!

Author(s)

P'eter S'olymos, <solymos@ualberta.ca>

References

S'olymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

S'olymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

Examples

```
## Not run:  
mefaweb()  
  
## End (Not run)
```

`melt`*Melt (Convert) an Object of Class 'mefa' Into 'stcs'*

Description

The function converts a wide formatted 'mefa' object into a long formatted 'stcs' object. The resulting segment column can take various formats.

Usage

```
melt(x, ...)
## S3 method for class 'mefa':
melt(x, segm.var = NULL, by.samp = TRUE,
     raw.out = FALSE, drop.zero = FALSE, ...)
```

Arguments

<code>x</code>	an object of class 'mefa'.
<code>segm.var</code>	NULL (default) or column indices of samples (if <code>by.samp = TRUE</code>) or taxa tables (if <code>by.samp = FALSE</code>). If it refers to more than one column, their interaction is taken. If NULL, the resulting segment column is filled with "undefined".
<code>by.samp</code>	logical, if TRUE (default) <code>segm.var</code> is taken from the samples table (<code>x\$samp</code>), if FALSE <code>segm.var</code> is taken from the taxa table (<code>x\$taxa</code>).
<code>raw.out</code>	logical, whether the result should contain all the zeros that is in the crosstabulated (wide) matrix (TRUE) or not (FALSE, default). Removing the zeros is straightforward for sparse matrices. But empty samples can be retained if <code>drop.zero = FALSE</code> .
<code>drop.zero</code>	logical, whether samples with zero total count should be leaved out (TRUE) or not (FALSE, default).
<code>...</code>	other arguments passed to the function <code>stcs</code> .

Details

If a 'mefa' object has original segments and `segm.var = NULL`, the function returns original segments. In this case, the `raw.out = TRUE` cannot be applied (because segments have different zero samples, so the option is meaningless). If the 'mefa' object contains only a data matrix without dimnames, rows and columns are numbered before melting.

Besides the goal to convert from wide format to long format, this 'mefa' specific implementation is also useful, if a segment is not coded along with the long formatted database, but can be found in a linked table of samples or species. In this way, a 'mefa' object with new segments can be defined easily.

If melting is done with a vector that is not part of the tables inside the 'mefa' object, it must have a class attribute to be recognised, use e.g. the `as.factor` function (see 'Examples').

Value

An object of class 'stsc' if `raw = FALSE`, pure 'data.frame' otherwise (because the zero count flag cannot be applied).

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

SVolymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

SVolymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

See Also

The generic function `melt` can also be found in the **reshape** package of Hadley Wickham, along with methods for data frames, arrays and lists (`melt.default`, `melt.data.frame`, `melt.array`).

Similar function in the **Hmisc** package by Frank Harrel is the `reShape`.

Similar function in the **simba** package by Gerald Jurasinski is the `liste`.

Examples

```
data(dol.count, dol.samp, dol.taxa)
x <- mefa(stcs(dol.count), dol.samp, dol.taxa)
## Long format with original segments
melt(x)
## Long format with undefined segment
melt(as.mefa(x, segment = FALSE))
## These two are identical
melt(x, "microhab")
melt(x, x$samp$microhab) ## this is a factor
## Using a simple vector as factor
melt(x, as.factor(rbinom(dim(x)[1], 1, 0.5)))
## Interaction
melt(x, c("method", "microhab"))
## From taxa table
melt(x, "familia", by.samp = FALSE)
## Example for non-integer data
y <- x$xtab
y[y > 0] <- y[y > 0] + 0.123
z <- mefa(y)
melt(z)
```

plot.mefa

Plotting Methods for Object of Class 'mefa' and 'stcs'

Description

Various methods for plotting objects of class 'mefa'.

Usage

```

## S3 method for class 'mefa':
plot(x, stat = 1:4, type = c("bar", "rank"),
trafo = c("none", "log", "ratio"), show = TRUE, ylab, xlab, ...)
## S3 method for class 'mefa':
boxplot(x, stat = 1:4, all = TRUE, show = TRUE, ylab, xlab, ...)
## S3 method for class 'mefa':
image(x, segm=NULL, trafo=c("none", "log", "bins", "prab"),
probs = seq(0, 1, 0.05), ordering = TRUE, reverse = TRUE, names = FALSE,
show = TRUE, ylab, xlab, ...)
## S3 method for class 'stcs':
plot(x, stat = 1:4, type = c("bar", "rank"),
trafo = c("none", "log", "ratio"), show = TRUE, ylab, xlab, ...)
## S3 method for class 'stcs':
boxplot(x, stat = 1:4, all = TRUE, show = TRUE, ylab, xlab, ...)
## S3 method for class 'stcs':
image(x, segm=NULL, trafo=c("none", "log", "bins", "prab"),
probs = seq(0, 1, 0.05), ordering = TRUE, reverse = TRUE, names = FALSE,
show = TRUE, ylab, xlab, ...)

```

Arguments

<code>x</code>	an object of class 'mefa' or 'stcs'.
<code>stat</code>	numeric, to determine which characteristic to plot. 1: number of species in samples (default), 2: total number of individuals in samples, 3: number of occurrences per taxa, 4 number of individuals per taxa.
<code>type</code>	character, "bar" produces barchart (default), "rank" ranked curves based on the characteristic defined by <code>stat</code> .
<code>trafo</code>	character, transformation of the plotted variable. "none": no transformation (default), "log": logarithmic transformation (base 10), "ratio": normalizes values by the maximum and rescales to the [0, 1] interval (useful for plotting multiple rank abundance curves), "bins": recodes the values according to quantiles based on <code>probs</code> , "prab": presence absence transformation of count data.
<code>all</code>	logical, if TRUE values of <code>stat</code> is plotted for all segments too on the boxplot.
<code>ylab, xlab</code>	character to overwrite default label for the y and x axes. If NULL, than default labels are returned on the plot.
<code>segm</code>	if NULL the <code>x\$xtab</code> matrix is used for plotting. Otherways, this defines the segment (one element in <code>x\$segm</code>) for plotting (can be numeric or character with the name of the segment).
<code>probs</code>	numeric vector of probabilities with values in [0, 1] (passed internally to <code>qvector</code>).
<code>ordering</code>	logical, if TRUE (default) the samples-by-taxa matrix is ordered by row and columns sums, if FALSE row and columns are not rearranged.
<code>reverse</code>	logical, if the values to plot should be reversed (TRUE, default, original zero values are lightly, while higher values are strongly coloured) or not (FALSE).

	This is related to <code>col</code> argument of the generic function <code>image</code> . Currently, <code>heat.colors</code> is the default color scheme.
<code>names</code>	logical, it labels samples and taxa in the plot using names in <code>x</code> . If it is a logical vector of length 2, sample and taxa names are returned accordingly.
<code>show</code>	logical, produce a plot (TRUE) or not (FALSE). FALSE can be useful, if the returned plotted values are reused (e.g. the matrix returned invisibly <code>image</code> can be used by <code>filled.contour</code> , or multiple values are used in one plot).
<code>...</code>	further arguments to pass to plotting functions. See especially <code>zlim</code> and <code>col</code> arguments of the generic function <code>image</code> , and arguments for the generic function <code>boxplot</code> .

Details

`plot` returns barchart, or ranked curve of summary statistics (number of species, individuals in samples, number of occurrences or abundance of taxa) based on the `x$xtab` matrix of the 'mefa' objects. These values are basically returned by `summary.mefa`.

`boxplot` returns box-and-whiskers plots for the summary statistics based on matrices for each segments in `x$segm`.

`image` creates a grid of colored rectangles with colors corresponding to the values in the segment defined by the argument `segm`. If `ordering = TRUE`, the ordering of the segment will be based on the `x$xtab` matrix and not on the matrix for the segment itself. This is due to better comparison among segments.

All graphical display methods for objects of class 'stcs' are based on the conversion of the object into 'mefa', and than the respective plotting method is applied. The conversion is made based on the default `mefa` settings (e.g. with segments). If more control is needed over the object structure, use the `mefa` function to coerce to a more appropriate class for this.

Value

All methods produce a plot if `show = TRUE`, and return the plotted values invisibly, or visibly if `show = FALSE`.

Author(s)

Péter Solyómos, <solyomos@ualberta.ca>

References

Solyómos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solyómos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

`image`, `plot`, `boxplot`, `filled.contour`, `par`, `heat.colors`, `topo.colors`, `terrain.colors`, `rainbow`

Examples

```

data(dol.count, dol.samp, dol.taxa)
x <- mefa(stcs(dol.count), dol.samp, dol.taxa)
## Frequency distributions
opar <- par(mfrow=c(2,2))
plot(x, 1)
plot(x, 2)
plot(x, 3)
plot(x, 4)
par(opar)
## Ranked curves
opar <- par(mfrow=c(2,2))
plot(x, 1, type="rank")
plot(x, 2, type="rank")
plot(x, 3, type="rank")
plot(x, 4, type="rank")
par(opar)
## Boxplot for segments
opar <- par(mfrow=c(2,2))
boxplot(x, 1)
boxplot(x, 2)
boxplot(x, 3)
boxplot(x, 4)
par(opar)
## Image (levelplot)
## comparing all and the segments
opar <- par(mfrow=c(1,3))
image(x, trafo = "bins", main = "All segments")
image(x, segm = 1, trafo = "bins", main = dimnames(x)$segm[1])
image(x, segm = 2, trafo = "bins", main = dimnames(x)$segm[2])
par(opar)
## For black and white, with names
image(x, col = grey(seq(0, 1, 0.1)), names = TRUE)
## For nice colors other than default
opar <- par(mfrow=c(1,2))
image(x[, "fresh"], col = topo.colors(10),
main = "Fresh")
image(x[, "broken"], col = terrain.colors(10),
main = "Broken")
par(opar)

```

qvector

*Ordinal Transformation of a Vector***Description**

Converts a numeric vector into an ordinal one based on quantiles.

Usage

```
qvector(x, probs = seq(0, 1, 0.25), na.rm = FALSE, ...)
```

Arguments

<code>x</code>	a numeric vector.
<code>probs</code>	numeric vector of probabilities with values in [0, 1] used by quantile .
<code>na.rm</code>	logical, if true, any NA and NaNs are removed from <code>x</code> before the quantiles are computed.
<code>...</code>	further arguments passed to the function quantile .

Value

A numeric vector of length the same as the input vector, containing the respective (upper) quantile values.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

[quantile](#)

Examples

```
x <- rnorm(20)
x
y <- qvector(x)
y
rbind(x, y)
plot(x, y, type="h")
```

rep.data.frame

Inflate (repeat) a Data Frame

Description

The function inflates a data frame based on count values in a vector corresponding the rows.

Usage

```
## S3 method for class 'data.frame':
rep(x, ...)
```

Arguments

`x` a matrix or data frame, but can be a vector.

`...` arguments passed to the function `rep`, i.e. `times`, `length.out` and `each` (see explanation there).

Details

Each rows of the data frame `x` will be repeated as many times as the corresponding value in e.g. the vector `times`. This function is called internally by the function `stcs`.

Value

A data frame with columns as in the original (`x`) and e.g. `sum(times)` number of rows.

Author(s)

Peter Solymos, <solymos@ualberta.ca>

References

Solymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Solymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

`stcs`

Examples

```
x <- data.frame(sample = LETTERS[c(1,1,2,2,3)],
               species = letters[c(5,5,5,6,7)],
               count = c(1,2,10,3,4), segment = letters[c(8,9,8,9,8)])
x
rep(x[,c(1,2,4)], times = x[,3])
rep(x[,c(1,2,4)], each = 2)
```

 report.mefa

 Write Report of an Object of Class 'mefa' into File

Description

The data of the 'mefa' object is written into a file in a standard format used by most local journals for publishing distribution data of organisms. The aim is to provide an exporting facility that can directly be used for generating reports, parts of research papers, or supplementary materials.

Usage

```
report(x, ...)
## S3 method for class 'mefa':
report(x, filename, segment = FALSE, n = NULL,
       by.taxa = TRUE, samp.var = NULL, drop.redundant = NULL,
       collapse = TRUE, taxa.name = NULL, author.name = NULL,
       taxa.order = NULL, grouping = FALSE, tex = FALSE, binary = FALSE,
       tex.control = list(ital.taxa = TRUE, noindent = TRUE,
                          bold.sect = TRUE, bold.lst = TRUE, vspace1 = 0.5, vspace2 = 0.2),
       sep = c(" ", ":", "(", ":", " ", ")", ";"), dir = getwd(), ...)
```

Arguments

x	an object of class 'mefa'.
filename	character, the name of the file to be written into the working directory or a directory defined by the <code>outdir</code> argument.
segment	logical, if <code>NULL</code> (default) segments are not distinguished in the output. If <code>TRUE</code> all (<code>n = NULL</code>) or some of the segments are used and distinguished.
n	segments to be used. If <code>NULL</code> all segments are used, if not <code>NULL</code> , give names of the segments to be used (only if <code>segment = TRUE</code>).
by.taxa	logical, whether the main sectioning should be made according to taxa (<code>TRUE</code> , default) or samples (<code>FALSE</code> , currently not implemented).
samp.var	variables in the samples table (<code>x\$samp</code>) to be used in the report. Variables appear in the order given here.
drop.redundant	logical, whether redundant elements should be dropped (an integer up to <code>length(samp.var) - 1</code>) or not (<code>NULL</code> , default).
collapse	logical, if <code>TRUE</code> samples with identical values in <code>samp.var</code> for a given species are collapsed to avoid redundancy.
taxa.name	column in the taxa table (<code>x\$ntaxa</code>) where the taxa names are stored. By default (<code>NULL</code>) <code>dimnames(x)\$ntaxa</code> is used.
author.name	column in the taxa table (<code>x\$ntaxa</code>) where the authors' names and dates of descriptions are stored. By default it is not used (<code>NULL</code>). If specified, these follow the species names in the report.

taxa.order	column in the taxa table (<code>x\$taxa</code>) where the variable for ordering is stored. By default (NULL) <code>taxa.name</code> is used for ordering.
grouping	logical, whether all records of a given species should be collapsed into single paragraph without grouping (FALSE, default), or grouped into separate paragraphs (TRUE).
tex	logical, whether to generate formatted LaTeX output (TRUE) or plain text file (FALSE).
binary	logical, if TRUE the result will contains count data.
tex.control	a list with LaTeX formatting options (if <code>tex = TRUE</code>). The <code>ital.taxa</code> controls italicised taxa names, <code>noindent</code> controls indentation of paragraphs, <code>bold.sect</code> controls boldface type of paragraph openings, <code>bold.1st</code> controls the boldface type of first elements of each record of a given species, <code>vspace1</code> controls the space between species blocks, <code>vspace2</code> controls the space between paragraphs of the same species (if <code>grouping = FALSE</code>).
sep	character vector of length 7. The 1st separates values in <code>samp.var</code> , 2nd is placed between values of <code>samp.var</code> if redundant elements were dropped, 3rd is the opening character before count data, 4th stands after the segment name, 5th separates values of segments, 6th is the closing character after the count data, 7th separates the records.
dir	character, full path of the directory where the file should be written (working directory is then restored). If NULL, the current working directoty is used.
...	further arguments potentially passed (currently there are none).

Details

The resulting file can be recycled by copy-and-pasting into a document (if `tex = FALSE`) or be included into a LaTeX report (e.g. via the package `Sweave`, see `mefadocs("SampleReport")`). Contrary to the many arguments, the default values do not need much modifications in most of the cases (see examples).

Value

Writes a file into the working (or a specified) directory.

Note

The `by.taxa = FALSE` option is not yet implemented.

Author(s)

P\eter S\olymos, <solymos@ualberta.ca>

References

S\olymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Svolymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29**(8), 1–28. <http://www.jstatsoft.org/v29/i08/>

<http://mefa.r-forge.r-project.org/>

See Also

[mefa, write](#)

Examples

```
data(dol.count, dol.samp, dol.taxa)
x <- mefa(stcs(dol.count), dol.samp, dol.taxa)
## Not run:
## Plain text
## Count values from x$xtab
report(x, "report-all.txt")
## Count values for each segments
report(x, "report-segm-all.txt",
       samp.var = c("method", "microhab"), segment = TRUE)
## LaTeX formatting
report(x, "report-all.tex", tex=TRUE)
## For how to include into a TeX or Rnw file, see:
mefadocs("SampleReport")

## End(Not run)
```

stcs

Create an Object of Class 'stcs'

Description

This function creates object of class 'stcs' from a data frame. This is the primary format for long (database style) data sets in the **mefa** package. The 'stcs' object can be converted into an object of class 'mefa' with the function `mefa`. The function name 'stcs' refers to first letters of column names of the resulting object: **s**amples, **t**axa, **c**ounts, **s**ebments.

Usage

```
stcs(dframe, expand = FALSE, drop.zero = FALSE, zero.pseudo = "zero.pseudo")
## S3 method for class 'stcs':
is(x)
## S3 method for class 'stcs':
summary(object, ...)
```

Arguments

<code>dframe</code>	a data frame with 2-4 columns. If 2 columns are provided, it is assumed that first column contains sample, while second taxa names. If 3 columns are provided, the first two is treated as sample and taxa names, while the third is treated as count if numeric (either interer or non-integer values can be supplied), and segment if character or factor. If 4 columns are provided, those are assumed to be in the samples, taxa, count, segment order.
<code>expand</code>	logical, whether the object should be (<code>TRUE</code>) expanded (via the function <code>rep.data.frame</code>) or not (<code>FALSE</code> , default). If expanded, each rows correspond to one individual observation (value of the <code>count</code> column is 1) or a zero sample. <code>expand = TRUE</code> works only with integer values.
<code>drop.zero</code>	logical, whether samples with zero total count should be left out (<code>TRUE</code>) or not (<code>FALSE</code> , default).
<code>zero.pseudo</code>	character, value in the <code>taxa</code> column that indicates zero samples. If it is a vector of length 2, first element is used for taxa names, second for segment names to indicate zero samples.
<code>x, object</code>	an object of class 'stcs'.
<code>...</code>	further arguments passed to the function <code>stcs</code> .

Details

If the data are in a long (database style) format, the `stcs` function prepares the ground for the `mefa` function to make the cross tabulation. But if only a crosstabulated matrix is needed, the `table` function can be applies as well (in this case be sure to set `expand = TRUE`, and this is only for integer data), or alternatively see the `xtabs` or `mefaCrosstab` functions (these accepts non-integer data). For subsetting, simply use extraction methods available for data frames (`[.data.frame]`).

The summary contains statistics calculated after coercing the object into the class 'mefa'.

Methods for extracting, subsetting are the same as for data frame objects. Plotting 'stcs' objects are discussed elsewhere (see links in 'See also' section).

Value

Returns an object of class 'stcs' with 4 columns. The column names are converted consistently into `samp` for samples, `taxa` for taxa names, `count` for counts and `segm` for segment names.

The class has its own methods (i.e. `is`, `as.stcs`, `summary` and `plot`), but also has a `data.frame` class attribute. Thus other methods that available for data frames are applicable to an object of class 'stcs' (e.g. `print`, `str`).

Note

The `stcs` function nearly equivalent to the `sscount` function in older (< 2.0) versions of the `mefa` package.

Author(s)

P\eter S\olyms, <solymos@ualberta.ca>

References

Sivolymos P. (2008) mefa: an R package for handling and reporting count data. *Community Ecology* **9**, 125–127.

Sivolymos P. (2009) Processing ecological data in R with the mefa package. *Journal of Statistical Software* **29(8)**, 1–28. <http://www.jstatsoft.org/v29/i08/>
<http://mefa.r-forge.r-project.org/>

See Also

[data.frame](#), [summary.data.frame](#), [str](#), [table](#)

See [plot.stcs](#) [boxplot.stcs](#) and [image.stcs](#) for graphical display options.

See [as.stcs](#) for coercion methods.

Similar standardized count data formats can be found in the **untb** package by Robin K. S. Hankin: [count](#), [census](#).

Examples

```
## General long format data
x <- data.frame(
  sample = paste("Sample", c(1,1,2,2,3,4), sep="."),
  species = c(paste("Species", c(1,1,1,2,3), sep="."), "zero.pseudo"),
  count = c(1,2,10,3,4,0),
  segment = letters[c(6,13,6,13,6,6)])
x
## Long format as stcs
y <- stcs(x)
y
## Methods
as.stcs(x)
is(y, "stcs")
is(y, "data.frame")
## Effects of arguments
stcs(x, expand = TRUE)
stcs(x, drop.zero = TRUE)
stcs(x, zero.pseudo = "pseudo.species")
## Input options
stcs(x[,1:2])
stcs(x[,1:3])
stcs(x[,c(1:2,4)])
## The Dolina dataset
data(dol.count)
dol <- stcs(dol.count)
dol
plot(dol)
```

Index

- *Topic **IO**
 - report.mefa, 31
 - *Topic **classes**
 - mefa, 12
 - stcs, 34
 - *Topic **datasets**
 - dolina, 7
 - *Topic **documentation**
 - mefadocs, 19
 - mefaweb, 23
 - *Topic **hplot**
 - mefalogo, 20
 - plot.mefa, 26
 - *Topic **manip**
 - aggregate.mefa, 2
 - as.data.frame.dist, 4
 - as.mefa, 5
 - fill.na, 9
 - label, 10
 - mat2list, 11
 - mefa, 12
 - mefaCheck, 16
 - mefaCrosstab, 18
 - mefaNestedless, 21
 - mefaTables, 22
 - melt, 24
 - qvector, 29
 - rep.data.frame, 30
 - stcs, 34
 - *Topic **methods**
 - aggregate.mefa, 2
 - as.mefa, 5
 - mefa, 12
 - melt, 24
 - plot.mefa, 26
 - report.mefa, 31
 - stcs, 34
 - [.data.frame, 34
 - [.mefa, 15
 - [.mefa (aggregate.mefa), 2
 - aggregate.mefa, 2, 15
 - all.equal.list, 17
 - as.array, 7
 - as.array.mefa (as.mefa), 5
 - as.data.frame, 6, 7
 - as.data.frame.dist, 4
 - as.data.frame.mefa (as.mefa), 5
 - as.data.frame.stcs (as.mefa), 5
 - as.factor, 3, 25
 - as.list, 7
 - as.list.mefa (as.mefa), 5
 - as.matrix, 7
 - as.matrix.mefa (as.mefa), 5
 - as.mefa, 5, 15
 - as.stcs, 35
 - as.stcs (as.mefa), 5
 - boxplot, 27, 28
 - boxplot.mefa, 15
 - boxplot.mefa (plot.mefa), 26
 - boxplot.stcs, 35
 - boxplot.stcs (plot.mefa), 26
- cast, 15
- cdrop (mat2list), 11
- census, 35
- count, 35
- data.frame, 35
- dim.mefa (mefa), 12
- dimnames.mefa (mefa), 12
- dist, 4
- dol.count (dolina), 7
- dol.samp (dolina), 7
- dol.taxa (dolina), 7
- dolina, 7
- fill.na, 9
- filled.contour, 27, 28

`heat.colors`, 27, 28
`image`, 27, 28
`image.mefa`, 15
`image.mefa` (`plot.mefa`), 26
`image.stcs`, 35
`image.stcs` (`plot.mefa`), 26
`is.dist` (`as.data.frame.dist`), 4
`is.mefa`, 17, 21
`is.mefa` (`mefa`), 12
`is.stcs` (`stcs`), 34

`label`, 10
`label<-` (`label`), 10
`liste`, 25

`mama`, 15
`mat2list`, 11
`matrify`, 15
`mefa`, 2, 3, 7, 12, 17–19, 21–23, 33
`mefa-package` (`mefa`), 12
`mefaCheck`, 16
`mefaCompare` (`mefaCheck`), 16
`mefaCrosstab`, 18, 34
`mefadocs`, 19
`mefalogo`, 20
`mefaNestedless`, 21
`mefaTables`, 22
`mefaweb`, 23
`melt`, 24
`melt.array`, 25
`melt.data.frame`, 25
`melt.default`, 25
`melt.mefa`, 15, 21
`mfill` (`mat2list`), 11
`msr` (`as.mefa`), 5
`mss` (`as.mefa`), 5
`mtr` (`as.mefa`), 5
`mts` (`as.mefa`), 5

`options`, 19

`par`, 20, 28
`plot`, 28
`plot.mefa`, 15, 26
`plot.stcs`, 35
`plot.stcs` (`plot.mefa`), 26
`print.mefa` (`mefa`), 12
`print.summary.mefa` (`mefa`), 12

`quantile`, 29, 30
`qvector`, 27, 29

`rainbow`, 28
`rdrop` (`mat2list`), 11
`rep`, 30
`rep.data.frame`, 19, 30, 34
`report` (`report.mefa`), 31
`report.mefa`, 15, 19, 31
`reShape`, 25

`stcs`, 7, 10, 18, 19, 24, 30, 31, 34
`str`, 35
`summary.data.frame`, 35
`summary.mefa`, 27
`summary.mefa` (`mefa`), 12
`summary.stcs` (`stcs`), 34
`Sweave`, 19

`table`, 15, 18, 34, 35
`terrain.colors`, 28
`topo.colors`, 28

`vec2dist` (`as.data.frame.dist`), 4
`vegandocs`, 19

`write`, 33

`xtabs`, 15, 18, 34