

Package ‘CA3variants’

December 9, 2020

Type Package

Title Three-Way Correspondence Analysis Variants

Version 3.0

Date 2020-12-09

Author Rosaria Lombardo, Eric J Beh and Michel van de Velden

Maintainer Rosaria Lombardo <rosaria.lombardo@unicampania.it>

Description Provides four variants of three-way correspondence analysis (ca):
three-way symmetrical ca, three-way non-symmetrical ca, three-way ordered symmetrical ca
and three-way ordered non-symmetrical ca.

Depends R (> 3.0.1), methods, tools, ggforce, gridExtra, ggrepel,
multichull, utils

Imports ggplot2, plotly, checkmate

LazyData true

License GPL (> 2)

URL <https://www.R-project.org>

NeedsCompilation no

Repository CRAN

Date/Publication 2020-12-09 15:20:02 UTC

R topics documented:

ca3basic	3
ca3plot	4
CA3variants	5
caplot3d	7
chi3	8
chi3ordered	9
coord	10
coordnsc3	10
criter	11
criteria	12

emerson.poly	12
flatten	13
happy	14
init3	15
init3ordered	16
init3ordered1	17
init3ordered2	18
Kron	19
loss1.3	19
loss1.3ordered	20
loss2	21
makeindicator	22
margI	22
margJ	23
margK	24
museum	24
newcomp3	26
newcomp3ordered	27
newcomp3ordered1	27
newcomp3ordered2	28
nsca3basic	29
oca3basic	30
olive	31
onsca3basic	32
p.ext	33
plot.CA3variants	34
print.CA3variants	35
prod3	38
reconst3	39
rstand3	40
simulabootsimple	41
simulabootstrat	41
standtab	42
step.g3	43
step.g3ordered	43
stepi3	44
stepi3ordered	45
tau3	46
tau3ordered	47
threewayboot	48
tucker	48
tuckerORDERED	49
tunelocal	51

ca3basic

*Three-way Symmetrical Correspondence Analysis***Description**

This function is used in the main function CA3variants when the input parameter is ca3type="CA3". It performs the three-way symmetrical correspondence analysis by TUCKALS3 algorithm.

Usage

```
ca3basic(x, p, q, r, test = 10^-6, ctr = T, std = T)
```

Arguments

x	The three-way contingency table.
p	The number of components of the first mode.
q	The number of components of the second mode.
r	The number of components of the third mode.
test	The treshold used in the algorithm TUCKALS3.
ctr	The flag parameter (T or F), if F the analysis is not centered.
std	The flag parameter (T or F) if F the analysis is not standardized.

Value

x	The original three-way contingency table.
xs	The weighted three-way contingency table.
xhat	Three-way contingency table reconstructed after Tuckals3 by principal components and core array.
nxhat2	The inertia of three-way symmetric correspondence analysis (Three-way Pearson ratio).
prp	The proportion of inertia reconstructed using the p, q, r principal components and the core array to the total inertia. To select the model dimensions (number of principal components), we examine the inertia explained by the p, q, r principal components with respect to the overall fit.
a	The row principal components.
b	The column principal coordinates.
cc	The tube principal coordinates.
g	The core array calculated by using the Tuckals3 algorithm and can be interpreted as generalised singular value table. They help to explain the strength of the association between the three principal components.
iteration	The number of iterations that are required for the TUCKALS3 algorithm to converge.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

 ca3plot

Row isometric biplot or Column isometric biplot

Description

This function is used in the main plot function when the plot type parameter is `plottype = "biplot"` and the variants of three-way CA are not ordered. It can produce a row or a column biplot.

Usage

```
ca3plot(frows, gcols, firstaxis, lastaxis, inertia pc, size1, size2, biptype, addlines)
```

Arguments

<code>frows</code>	The row principal or standard coordinates.
<code>gcols</code>	The column principal or standard coordinates.
<code>firstaxis</code>	The first axis number.
<code>lastaxis</code>	The second axis number.
<code>inertiapc</code>	The percentage of the explained inertia by each dimension.
<code>size1</code>	The size of the plotted symbol for categories in biplots.
<code>size2</code>	The size of the plotted text for categories in biplots.
<code>biptype</code>	The input parameter for specifying what kind of biplots is requested. By default, it is equal to <code>column-tube</code> , but could be <code>row</code> .
<code>addlines</code>	The input parameter for plotting lines in biplot. By default, it is equal to <code>addlines = TRUE</code> .

Note

This function depends on the R library `plotly`.

Author(s)

Rosaria Lombardo, Eric J. Beh and Michel van de Velden.

References

- Beh EJ and Lombardo R 2014 Correspondence Analysis: Theory, Practice and New Strategies. Wiley.
- Lombardo R Beh EJ 2016 Variants of Simple Correspondence Analysis. *The R Journal*, 8 (2), 167–184.
- Van de Velden M, Iodice D’Enza A, Palumbo F 2017. Cluster Correspondence Analysis. *Psychometrika*, 82, 158–185.
- Gower JC, Lubbe SG, and Le Roux, NJ 2011. Understanding biplots. New York: Wiley.

 CA3variants

Correspondence Analysis variants for three-way contingency tables

Description

This function performs four variants of three-way correspondence analysis (CA). It does the three-way symmetrical CA, when `ca3type = "CA3"`, and three-way non-symmetrical CA, when `ca3type = "NSCA3"`, by using the Tucker3 decomposition. It also performs ordered three-way symmetrical CA, when `ca3type = "OCA3"`, and ordered three-way non-symmetrical CA, when `ca3type = "ONSCA3"`, by using the Trivariate Moment Decomposition. The non-symmetrical variants consider the three variables asymmetrically related, such that one of the variables is the response to be predicted given the other two variables. It calculates the coordinates and inertia values of the chosen analyses. Furthermore, it allows to look at the index (Pearson’s chi-squared or Marcotorchino’s tau) partition.

Usage

```
CA3variants(Xdata, dims=c(p,q,r), ca3type = "CA3", test = 10^-6, resp = "row", norder = 3)
```

Arguments

- | | |
|----------------------|---|
| <code>Xdata</code> | The three-way data. It can be a R object array or raw data (n individuals by three categorical variables, for an example, see museum data). When a three-way non-symmetrical variant is performed, by default, the response variable is the row variable when an array is given, or the first of three columns when a raw data set is given. For changing, consider the parameter <code>resp = "col"</code> or <code>resp = "tube"</code> . |
| <code>dims</code> | The number of components for the first, second and third mode. By default, no <code>dims</code> is given. When using an ordered variant of three-way CA recall to consider the complete dimension, i.e. the number of components for the first, second and third mode must be equal to the number of rows, columns and tubes, respectively. |
| <code>ca3type</code> | The specification of the analysis to be performed. If <code>ca3type = "CA3"</code> , then a three-way symmetrical correspondence analysis will be performed (default analysis). If <code>ca3type = "NSCA3"</code> , then three-way non-symmetrical correspondence analysis will be performed. If <code>ca3type = "OCA3"</code> , then ordered three-way symmetrical correspondence analysis will be performed. If <code>ca3type = "ONSCA3"</code> , then ordered three-way non-symmetrical correspondence analysis will be performed. |

test	Threshold used in the algorithm for stopping it after the convergence of the solutions.
resp	The input parameter for specifying in non-symmetrical three-way correspondence analysis variants (<code>ca3type = "NSCA3"</code> and <code>ca3type = "ONSCA3"</code>) what is the response variable (logically antecedent to the others). By default, <code>resp = "row"</code> , but it could be <code>resp = "col"</code> or <code>resp = "tube"</code> .
norder	The input parameter for specifying the number of ordered variable when <code>ca3type = "OCA3"</code> or <code>ca3type = "ONSCA3"</code> . By default, all three variables are ordered <code>norder = 3</code> . When <code>norder = 1</code> , you assume that the ordered variable is the column variable. When <code>norder = 2</code> , you assume that the ordered variables are the row and column variable.

Details

This function recall internally many other functions, depending on the setting of the input parameters. After performing three-way symmetric or non-symmetric correspondence analysis, it recall two functions for printing and plotting the results. These two important functions are `print.CA3variants` and `plot.CA3variants`.

Value

The value of output returned depends on the kind of analysis performed. For a detailed description of the output one can see:
the output value of `ca3basic` if the input parameter is `ca3type="CA3"`; the output value of `nsca3basic` if the input parameter is `ca3type="NSCA3"`; the output value of `oca3basic` if the input parameter is `ca3type="OCA3"` the output value of `onsca3basic` if the input parameter is `ca3type="ONSCA3"`

Author(s)

Rosaria Lombardo, Eric J Beh and Michel van de Velden.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.
Kroonenberg PM (1994). The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.
Lombardo R, Beh EJ and Kroonenberg PM (2020-preprint) Symmetrical and Non-Symmetrical Variants of Three-Way Correspondence Analysis for Ordered Variables. Statistical Science Journal, p. 1-33.

Examples

```
data(museum)
CA3variants(Xdata = museum, dims = c(p=2,q=2,r=2), ca3type = "CA3")
data(happy)
CA3variants(Xdata = happy, dims = c(p=3,q=5,r=4), ca3type = "OCA3")
CA3variants(Xdata = happy, dims = c(p=2,q=2,r=2), ca3type = "NSCA3")
CA3variants(Xdata = happy, dims = c(p=3,q=5,r=4), ca3type = "ONSCA3")
```

caplot3d	<i>Three dimensional correspondence plot</i>
----------	--

Description

This function is used in the plot function plot.CAvariants when the logical parameter is plot3d = TRUE. It produces a 3-dimensional visualization of the association.

Usage

```
caplot3d(coordR, coordC, inertiaPer, firstaxis = 1, lastaxis = 2, thirddaxis = 3)
```

Arguments

coordR	The row principal or standard coordinates.
coordC	The column principal or standard coordinates.
inertiaPer	The percentage of the total inertia explained inertia by each dimension.
firstaxis	The first axis number. By default, firstaxis = 1.
lastaxis	The second axis number. By default, lastaxis = 2.
thirddaxis	The third axis number. By default, thirddaxis = 3.

Note

This function depends on the R library plotly.

Author(s)

Rosaria Lombardo and Eric J. Beh

References

Beh EJ and Lombardo R 2014 Correspondence Analysis: Theory, Practice and New Strategies. Wiley.
Lombardo R Beh EJ 2016 Variants of Simple Correspondence Analysis. The R Journal, 8 (2), 167–184.

`chi3`*The partition of the Pearson three-way index*

Description

When three categorical variables are symmetrically related, we can analyse the strength of the association using the three-way Pearson mean square contingency coefficient, named the chi-squared index. The function `chi3` partitions the Pearson phi-squared statistic when in `CA3variants` we set the parameter `ca3type = "CA3"`.

Usage

```
chi3(f3, digits = 3)
```

Arguments

<code>f3</code>	The three-way contingency array given as an input parameter in <code>CA3variants</code> .
<code>digits</code>	The number of decimal digits. By default <code>digits=3</code> .

Value

The partition of the Pearson index into three two-way association terms and one three-way association term. It also shows the explained inertia, the degrees of freedom and p-value of each term of the partition.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.
Carlier A and Kroonenberg P (1996). Decompositions and biplots in three-way correspondence analysis. *Psychometrika*, 61, 355-373.

Examples

```
data(happy)
chi3(f3=happy, digits=3)
```

chi3ordered	<i>The partition of the Pearson three-way index.</i>
-------------	--

Description

When three categorical variables are symmetrically related, we can analyse the strength of the symmetrical association using the three-way Pearson statistic. The function `chi3ordered` partitions the Pearson phi-squared statistic using orthogonal polynomials when, in `CA3variants`, we set the parameter `ca3type = "OCA3"`.

Usage

```
chi3ordered(f3, digits = 3)
```

Arguments

<code>f3</code>	The three-way contingency array given as an input parameter in <code>CA3variants</code> .
<code>digits</code>	The number of decimal digits. By default <code>digits=3</code> .

Value

The partition of the Pearson index into three two-way association terms and one three-way association term. It also shows the polynomial components of inertia, the percentage of explained inertia, the degrees of freedom and p-value of each term of the partition.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). *Correspondence Analysis, Theory, Practice and New Strategies*. John Wiley & Sons.

Examples

```
data(happy)
chi3ordered(f3=happy, digits=3)
```

 coord

The weighted components of the Tucker3 algorithm

Description

The function computes the weighted components from the Tucker3 algorithm (to take into account the different weight systems in row, column and tube spaces) for symmetrical three-way correspondence analysis.

Usage

```
coord(res, x)
```

Arguments

res	The component matrices resulting from the Tucker3 algorithm.
x	The original three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

 coordnsc3

The weighted components of the Tucker3 algorithm

Description

The function computes the weighted components from the Tucker3 algorithm (to take into account the different weight systems in row, column and tube spaces) for non-symmetrical three-way correspondence analysis.

Usage

```
coordnsc3(res, x)
```

Arguments

res	The component matrices resulting from the Tucker3 algorithm.
x	The original three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.\

criter

The stopping criteria for the Tucker3 algorithm

Description

The function computes the stopping criteria needed for the Tucker3 algorithm.

Usage

```
criter(x, xhat)
```

Arguments

x	The three-way contingency table
xhat	The reconstruction of the data array using the three component matrices and the core array.

Value

The criterion used in Tuckals3 is the sum of squares of the differences between the weighted data array and the reconstructed data array.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.\

criteria	<i>Criterion of the Tucker3 algorithm</i>
----------	---

Description

The function criteria is used to define the criterion of the Tucker3 algorithm.

Usage

```
criteria(aold, anew)
```

Arguments

aold	The old component of the first mode.
anew	The new component of the first mode.

Author(s)

Rosaria Lombardo, Eric J Beh

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

emerson.poly	<i>Orthogonal polynomials</i>
--------------	-------------------------------

Description

This function is called from the function oca3basic when in CA3variants we set ca3type = "OCA3". It allows the analyst to compute the orthogonal polynomials of the ordered categorical variable. The number of the polynomials is equal to the variable category less one. The function computes the polynomial transformation of the ordered categorical variable.

Usage

```
emerson.poly(mj, pj)
```

Arguments

mj	The ordered scores of an ordered variable. By default mj = NULL, the natural scores (1,2,...) are computed.
pj	The marginals, relative frequencies of the ordered variable.

Value

Describe the value returned

B The matrix of the orthogonal polynomials without the trivial polynomial.

Note

Note that the sum of the marginals of the ordered variables should be one.

Author(s)

Rosaria Lombardo and Eric J Beh.

References

Beh EJ and Lombardo R 2014 Correspondence Analysis: Theory, Practice and New Strategies. John Wiley & Sons.

Emerson PL 1968 Numerical construction of orthogonal polynomials from a general recurrence formula. *Biometrics*, 24 (3), 695-701.

Lombardo R Beh EJ Variants of Simple Correspondence Analysis. *The R Journal*, 8 (2), 167–184.

Examples

```
emerson.poly(c(1,2,3,4,5), as.vector(c(.1,.2,.3,.2,.2)))
```

 flatten

Flattened table

Description

The function flattens the three-way table into the concatenation of two-way matrices.

Usage

```
flatten(x)
```

Arguments

x The three-way contingency table.

Details

It is utilised by a number of functions: CA3variants, reconst3, newcomp3 and step.g3.

Value

x The flattened table of size I,JK where I, J and K are the number of the categories of rows, columns and tubes, respectively.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

happy

Three-way contingency table

Description

This three-way contingency table represents an historical data set found in Beh and Lombardo (2014).

Usage

data(happy)

Format

The format is:

row names [1:3] "H1", "H2", "H3" col names [1:5] "S1", "S2", "S3", "S4", "S5" tube names [1:4] "E1", "E2", "E3", "E4"

References

Beh EJ and Lombardo R 2014 Correspondence Analysis: Theory, Practice and New Strategies. John Wiley & Sons.

Examples

```
happy <-
structure(c(15, 17, 7, 34, 53, 20, 36, 70, 23, 22, 67, 16, 61,
79, 36, 31, 60, 5, 60, 96, 12, 46, 45, 11, 25, 40, 12, 26, 31,
7, 35, 63, 5, 45, 74, 10, 30, 39, 4, 13, 24, 4, 8, 7, 3, 18,
15, 2, 14, 15, 1, 3, 9, 2, 3, 2, 0, 4, 1, 1), .Dim = c(3L, 5L,
4L), .Dimnames = list(c("H1", "H2", "H3"), c("S1", "S2", "S3",
"S4", "S5"), c("E1", "E2", "E3", "E4")))
dim(happy)
```

init3	<i>Initial components from the Tuckals3 algorithm</i>
-------	---

Description

The function is utilised from the function tucker to compute the initial components for each of the three categorical variables.

Usage

```
init3(x, p, q, r)
```

Arguments

x	The three-way contingency table.
p	The number of components of the first mode.
q	The number of components of the second mode.
r	The number of components of the third mode.

Value

The initial components for each of the three categorical variables.

a	The initial component derived from the Tucker3 decomposition for the first mode.
b	The initial component derived from the Tucker3 decomposition for the second mode.
cc	The initial component derived from the Tucker3 decomposition for the third mode.
x	The three-way contingency table

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

init3ordered	<i>Initial components from the Trivariate Moment Decomposition algorithm</i>
--------------	--

Description

The function is utilised from the function tuckerordered to compute the initial components for each of the three ordered categorical variables.

Usage

```
init3ordered(x, p, q, r, x0)
```

Arguments

x	The three-way contingency table.
p	The number of components of the first mode.
q	The number of components of the second mode.
r	The number of components of the third mode.
x0	The original three-way contingency table.

Value

The initial components for each of the three categorical variables.

a	The initial component derived from the Trivariate Moment Decomposition for the first mode.
b	The initial component derived from the Trivariate Moment Decomposition for the second mode.
cc	The initial component derived from the Trivariate Moment Decomposition for the third mode.
x	The three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh E J and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

init3ordered1	<i>Initial components from the Trivariate Moment Decomposition algorithm</i>
---------------	--

Description

The function is utilised from the function tuckerORDERED to compute the initial components for the first ordered categorical variables.

Usage

```
init3ordered1(x, p, q, r, x0)
```

Arguments

x	The three-way contingency table.
p	The number of components of the first mode.
q	The number of components of the second mode.
r	The number of components of the third mode.
x0	The original three-way contingency table.

Value

The initial components for each of the three categorical variables.

a	The initial component derived from the Trivariate Moment Decomposition for the first mode.
b	The initial component derived from the Trivariate Moment Decomposition for the second mode.
cc	The initial component derived from the Trivariate Moment Decomposition for the third mode.
x	The three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

init3ordered2	<i>Initial components from the Trivariate Moment Decomposition algorithm</i>
---------------	--

Description

The function is utilised from the function `tuckerordered` to compute the initial components for each of the two ordered categorical variables.

Usage

```
init3ordered2(x, p, q, r, x0)
```

Arguments

<code>x</code>	The three-way contingency table.
<code>p</code>	The number of components of the first mode.
<code>q</code>	The number of components of the second mode.
<code>r</code>	The number of components of the third mode.
<code>x0</code>	The original three-way contingency table.

Value

The initial components for each of the three categorical variables.

<code>a</code>	The initial component derived from the Trivariate Moment Decomposition for the first mode.
<code>b</code>	The initial component derived from the Trivariate Moment Decomposition for the second mode.
<code>cc</code>	The initial component derived from the Trivariate Moment Decomposition for the third mode.
<code>x</code>	The three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). *Correspondence Analysis, Theory, Practice and New Strategies*. John Wiley & Sons.

Kron	<i>Kronecker product</i>
------	--------------------------

Description

The function performs the Kronecker product. Starting from two matrices of dimension $I \times P$ and $J \times Q$ the resulting matrix will be of dimension $I \times J, P \times Q$.

Usage

```
Kron(a, b)
```

Arguments

a	The first matrix of dimension $I \times P$ involved in the kronecker product.
b	The second matrix of dimension $J \times Q$ involved in the kronecker product.

Details

This function is utilised from several other functions like `CA3variants`, `newcomp3`, `step.g3` and `reconst3`.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

loss1.3	<i>General loss criterion</i>
---------	-------------------------------

Description

This function represents the general loss function on which is based `Tuckals3` and calculates the difference between two arrays, `x` and `xhat`, where `x` is the three-way contingency table and `xhat` is the reconstruction of this table by means of components and core array.

Usage

```
loss1.3(param, comp.old)
```

Arguments

param	The matrices of the row, column and tube components derived via the Tucker3 model.
comp.old	The matrices of the row, column and tube components derived in the foregoing iteration of the Tuckals3 algorithm.

Value

The difference between three-way contingency table and its reconstruction from the Tucker3 model.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

loss1.3ordered	<i>General loss criterion</i>
----------------	-------------------------------

Description

This function represents the general loss function on which is based the Trivariate Moment Decomposition and calculates the difference between two arrays, x and $xhat$, where x is the three-way contingency table and $xhat$ is the reconstruction of this table by means of components and core array.

Usage

```
loss1.3ordered(param, comp.old)
```

Arguments

param	The matrices of the row, column and tube components derived via the Trivariate Moment Decomposition model.
comp.old	The matrices of the row, column and tube components derived in the foregoing iteration of the Trivariate Moment Decomposition algorithm.

Value

The difference between three-way contingency table and its reconstruction from the Trivariate Moment Decomposition model.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

loss2

Difference between two successive components

Description

The function computes the difference between two successive components in the iteration of the Tuckals3 algorithm.

Usage

```
loss2(param, comp.old)
```

Arguments

param	The matrices of the row, column and tube components derived via the Tucker3 model.
comp.old	The matrices of the row, column and tube components derived in the foregoing iteration of the Tuckals3 algorithm.

Value

The difference between two successive components in the iteration of the Tuckals3 algorithm.

Author(s)

Rosaria Lombardo and Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

makeindicator	<i>Make an Indicator matrix</i>
---------------	---------------------------------

Description

From a three-way contingency table (as can be used in CA3variants), it gives the N x total number of categories (rows+cols+tubs) indicator matrix

Usage

```
makeindicator(X)
```

Arguments

X The three-way data array. It must be an R object array.

Value

Z Output: the N x total number of categories (rows+cols+tubs) indicator matrix

Author(s)

Rosaria Lombardo, Michel van de Velden, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.\

Examples

```
data(happy)
makeindicator(happy)
```

margI	<i>Row marginals of a three-way contingency table</i>
-------	---

Description

This function computes the row marginals of the three-way contingency table specified by the input parameter.

Usage

```
margI(m)
```

Arguments

m The three-way contingency table.

Value

The row marginals of the considered three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

margJ *Column marginals of a three-way contingency table*

Description

The function computes the column marginals of the three-way contingency table specified by the input parameter.

Usage

margJ(m)

Arguments

m The three-way contingency table.

Value

The column marginals of the considered three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

margK

Tube marginals of a three-way contingency table

Description

The function computes the tube marginals of the three-way contingency table specified by the input parameter.

Usage

```
margK(m)
```

Arguments

`m` The three-way contingency table.

Value

The tube marginals of the considered three-way contingency table.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

museum

Raw data: Three variables from a survey

Description

This raw data table represents a possible data set selected from a large survey on customer satisfaction during museum visiting. The rows are individuals. The first column concerns the number of visits, the second column concerns if they like it, and the third column their satisfaction.

Usage

```
data(museum)
```

Format

The format is: num [1:223, 1:3] "often" "much" "excellent" ...


```

"good", "suff", "unsuff"), class = "factor")), class = "data.frame", row.names = c("1",
"2", "3", "5", "6", "8", "9", "10", "12", "13", "14", "16", "17",
"18", "19", "20", "21", "22", "23", "24", "25", "27", "30", "31",
"32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42",
"43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "54",
"55", "56", "57", "58", "59", "60", "61", "64", "65", "66", "67",
"68", "69", "70", "71", "72", "73", "74", "75", "78", "80", "81",
"82", "84", "85", "86", "87", "88", "89", "90", "91", "92", "95",
"96", "97", "98", "99", "100", "101", "102", "104", "105", "106",
"107", "108", "109", "110", "111", "112", "113", "115", "116",
"117", "118", "119", "120", "121", "122", "123", "124", "125",
"126", "127", "128", "129", "130", "131", "132", "133", "136",
"138", "139", "140", "142", "143", "144", "145", "146", "147",
"148", "149", "150", "151", "153", "154", "155", "156", "157",
"158", "159", "160", "162", "163", "165", "166", "167", "168",
"169", "170", "171", "173", "174", "175", "176", "177", "178",
"179", "180", "181", "182", "183", "184", "185", "186", "187",
"189", "190", "191", "192", "193", "194", "195", "196", "197",
"198", "200", "201", "202", "203", "204", "205", "206", "207",
"208", "209", "210", "211", "212", "213", "214", "215", "217",
"218", "219", "220", "221", "222", "223", "224", "225", "227",
"228", "229", "230", "231", "232", "233", "234", "235", "236",
"237", "238", "239", "240", "241", "242", "243", "244", "245",
"246", "247", "248", "249", "250", "251", "252", "253"))
dim(museum)
data(museum)

```

newcomp3

Updated component matrices

Description

The function computes the updated component matrices of the Tucker3 decomposition.

Usage

```
newcomp3(param)
```

Arguments

param	The initial matrices of the row, column and tube components derived via the <code>init3</code> function.
-------	--

Details

It is utilised from the function `tucker`.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

newcomp3ordered *Updated component matrices*

Description

The function computes the updated component matrices of the Trivariate Moment Decomposition. It is supposed that the number of the ordered categorical variables is equal to 3.

Usage

```
newcomp3ordered(param)
```

Arguments

param	The initial matrices of the row, column and tube components derived via the <code>init3</code> function.
-------	--

Details

It is utilised from the function `tuckerORDERED`.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

newcomp3ordered1 *Updated component matrices*

Description

The function computes the updated component matrices of the Trivariate Moment Decomposition. It is supposed that the number of the ordered categorical variables is equal to 1.

Usage

```
newcomp3ordered1(param)
```

Arguments

param The initial matrices of the row, column and tube components derived via the `init3` function.

Details

It is utilised from the function `tuckerORDERED`.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). *Correspondence Analysis, Theory, Practice and New Strategies*. John Wiley & Sons.

newcomp3ordered2 *Updated component matrices*

Description

The function computes the updated component matrices of the Trivariate Moment Decomposition. It is supposed that the number of the ordered categorical variables is equal to 2.

Usage

```
newcomp3ordered2(param)
```

Arguments

param The initial matrices of the row, column and tube components derived via the `init3` function.

Details

It is utilised from the function `tuckerORDERED`.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). *Correspondence Analysis, Theory, Practice and New Strategies*. John Wiley & Sons.

nsca3basic

*Three-way Non-Symmetrical Correspondence Analysis***Description**

This function is used in the main function `CA3variants` when the input parameter is `catype="NSCA3"`. It decomposes the Marcotorchino index, computes principal axes, coordinates, weights of rows and columns, total inertia (equal to the Marcotorchino index) and the rank of the matrix.

Usage

```
nsca3basic(x, p, q, r, test = 10^-6, ctr = T, std = T)
```

Arguments

<code>x</code>	The three-way contingency table.
<code>p</code>	The number of components of the first mode.
<code>q</code>	The number of components of the second mode.
<code>r</code>	The number of components of the third mode.
<code>test</code>	The threshold used in the algorithm.
<code>ctr</code>	The flag parameter to center the data (T or F), if F the data are not centered.
<code>std</code>	The flag parameter to weight the data (T or F), if F the data are not weighted.

Value

<code>x</code>	The original three-way contingency table.
<code>xs</code>	The weighted three-way contingency table.
<code>xhat</code>	The three-way contingency table reconstructed after Tuckals3 by means of the principal components and core array.
<code>nxhat2</code>	The inertia of the three-way non-symmetrical correspondence analysis for one response (the three-way Marcotorchino index).
<code>prp</code>	The proportion of inertia reconstructed using the principal components and the core array to the total inertia. To select the model dimensions (number of principal components), we examine the inertia explained by the <code>p</code> , <code>q</code> , <code>r</code> principal components with respect to the overall fit.
<code>a</code>	The row principal components.
<code>b</code>	The column principal components.
<code>cc</code>	The tube principal components.
<code>g</code>	The core array (generalized singular values) calculated by using the Tuckals3 algorithm. They help to explain the strength of the association among the three principal components.

`iteration` The number of iterations that are required for the TUCKALS3 algorithm to converge.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

oca3basic

Three-way Ordered Symmetrical Correspondence Analysis

Description

This function is used in the main function CAvariants when the input parameter is `ca3type="OCA3"`. It performs the three-way symmetric correspondence analysis by TUCKALS3.

Usage

```
oca3basic(x, p, q, r, test = 10^-6, ctr = T, std = T, norder=3)
```

Arguments

<code>x</code>	The three-way contingency table.
<code>p</code>	The number of components of the first mode.
<code>q</code>	The number of components of the second mode.
<code>r</code>	The number of components of the third mode.
<code>test</code>	The threshold used in the algorithm TUCKALS3.
<code>ctr</code>	The flag parameter (T or F), if F the analysis is not centered.
<code>std</code>	The flag parameter (T or F) if F the analysis is not standardized.
<code>norder</code>	The number of ordered variables considered.

Value

<code>x</code>	The original three-way contingency table.
<code>xs</code>	The weighted three-way contingency table.
<code>xhat</code>	Three-way contingency table reconstructed after Tuckals3 by principal components and core array
<code>nxhat2</code>	The inertia of three-way symmetric correspondence analysis (Three-way Pearson ratio).

prp	The proportion of inertia reconstructed using the p, q, r principal components and the core array to the total inertia. To select the model dimensions (number of principal components), we examine the inertia explained by the p, q, r principal components with respect to the overall fit.
a	The row principal components.
b	The column principal coordinates.
cc	The tube principal coordinates.
g	The core array calculated by using the Tuckals3 algorithm and can be interpreted as generalised singular value table. They help to explain the strength of the association between the three principal components.
iteration	The number of iterations that are required for the TUCKALS3 algorithm to converge.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

olive *Three-way contingency table*

Description

This three-way contingency table represents an historical data set found in Agresti (1990).

Usage

```
data(olive)
```

Format

The format is:

```
row names [1:6] "A", "B", "C", "D", "E", "F" col names [1:3] "NW", "NE", "SW" tube names [1:2]
"urban", "rural"
```

References

Beh EJ and Lombardo R 2014 Correspondence Analysis: Theory, Practice and New Strategies. John Wiley & Sons.

Examples

```
olive <-structure(c(20, 15, 12, 17, 16, 28, 18, 17, 18, 18,
6, 25, 12, 9, 23, 21, 19, 30, 30, 22, 21, 17, 8,
12, 23, 18, 20, 18, 10, 15, 11, 9, 26, 19, 17, 24
), .Dim = c(6L, 3L, 2L), .Dimnames = list(c("A", "B", "C", "D",
"E", "F"), c("NW", "NE", "SW"), c("urban", "rural")))
dim(olive)
data(olive)
```

onsca3basic

Three-way Ordered Non-Symmetrical Correspondence Analysis

Description

This function is used in the main function CAvariants when the input parameter is ca3type="ONSCA3". It performs the three-way symmetric correspondence analysis by TUCKALS3.

Usage

```
onsca3basic(x, p, q, r, test = 10^-6, ctr = T, std = T, norder=3)
```

Arguments

x	The three-way contingency table.
p	The number of components of the first mode.
q	The number of components of the second mode.
r	The number of components of the third mode.
test	The threshold used in the algorithm TUCKALS3.
ctr	The flag parameter (T or F), if F the analysis is not centered.
std	The flag parameter (T or F) if F the analysis is not standardized.
norder	The number of ordered variables considered.

Value

x	The original three-way contingency table.
xs	The weighted three-way contingency table.
xhat	Three-way contingency table reconstructed after Tuckals3 by principal components and core array
nxhat2	The inertia of three-way symmetric correspondence analysis (Three-way Pearson ratio).
prp	The proportion of inertia reconstructed using the p, q, r principal components and the core array to the total inertia. To select the model dimensions (number of principal components), we examine the inertia explained by the p, q, r principal components with respect to the overall fit.

a	The row principal components.
b	The column principal coordinates.
cc	The tube principal coordinates.
g	The core array calculated by using the Tuckals3 algorithm and can be interpreted as generalised singular value table. They help to explain the strength of the association between the three principal components.
iteration	The number of iterations that are required for the TUCKALS3 algorithm to converge.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

p.ext

The external product in Tuckals3.

Description

The computation of external product between the principal components.

Usage

p.ext(x,y)

Arguments

x	x matrix IxS
y	y matrix JxS

Value

resultant matrix (IxJ),S with elements xis per yis

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

plot.CA3variants *Graphical display resulting from CAvariants3*

Description

The function `plot.CA3variants` allows the analyst to graphically display six types of biplots from their analysis. When the input parameter is `biptype = "column-tube"` (or `biptype = "col-tube"`), the function displays the column-tube interactive biplot. When the input parameter is `biptype = "row"`, the function displays the row biplot. When the input parameter is `biptype = "col"`, the function displays the column biplot. When the input parameter is `biptype = "row-tube"`, the function displays the row-tube biplot. When the input parameter is `biptype = "tube"`, the function displays the tube biplot. When the input parameter is `biptype = "row-column"` (or `biptype = "row-col"`), the function displays the row-column interactive biplot.

By default, `biptype = "column-tube"`.

Usage

```
## S3 method for class 'CA3variants'
plot(x, firstaxis = 1, lastaxis = 2, thirdaxis = 3, cex = 0.8,
     biptype="column-tube", scaleplot = NULL, plot3d = FALSE, pos = 1,
     size1 = 1, size2 = 2, addlines = TRUE,...)
```

Arguments

<code>x</code>	The output parameters of the main function <code>CA3variants</code> .
<code>firstaxis</code>	The dimension reflected along the horizontal axis.
<code>lastaxis</code>	The dimension reflected along the vertical axis.
<code>thirdaxis</code>	The dimension reflected along the third axis when <code>plot3d = TRUE</code> .
<code>cex</code>	The parameter that specifies the size of character labels of points in graphical displays. By default, it is equal to 0.8.
<code>biptype</code>	The input parameter for specifying what kind of biplot is requested. By default, it is equal to <code>column-tube</code> , but could be <code>row-tube</code> , <code>row-column</code> , <code>row</code> , <code>column</code> and <code>tube</code> .
<code>scaleplot</code>	The scaling parameter for biplots to pull points away from the origin (see gamma biplot in Gower et al 2011). By default, it is equal to the overall average for the sum of squares of the two sets of coordinates (principal and standard ones), because of the average sum of squares for the two sets of points is the same (see Van de Velden et al 2017).
<code>plot3d</code>	The logical parameter specifies whether a 3D plot is to be included in the output or not. By default, <code>plot3d = FALSE</code> .
<code>pos</code>	The input parameter for changing the label position. By default, it is equal to 1.
<code>size1</code>	The input parameter for specifying the size of pointers. By default, it is equal to 1.
<code>size2</code>	The input parameter for specifying the label size. By default, it is equal to 2.

addlines The input parameter for plotting lines in biplots (the points in standard coordinates are represented using lines). By default, it is equal to addlines = TRUE.

... Further arguments passed to or from other methods.

Details

It is utilised by the main function `CA3variants` and uses the secondary graphical function `graph2poly`.

Value

Graphical displays of three-way correspondence analysis variants. Interactive plots or biplots are the graphical results of this function.

Author(s)

Rosaria Lombardo, Eric J Beh and Michel van de Velden.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

Van de Velden M, Iodice D'Enza A, Palumbo F 2017. Cluster Correspondence Analysis. *Psychometrika*, 82, 158–185.

Gower JC, Lubbe SG, and Le Roux, NJ 2011. Understanding biplots. New York: Wiley.

Lombardo R, Beh EJ and Kroonenberg PM (2020-preprint) Symmetrical and Non-Symmetrical Variants of Three-Way Correspondence Analysis for Ordered Variables. *Statistical Science Journal*, p. 1-33.

Examples

```
data(happy)
ris.ca3<-CA3variants(happy, dims = c(p = 2, q = 2, r = 2), ca3type = "CA3")
plot(ris.ca3)
ris.nsca3<-CA3variants(happy, dims = c(p = 2, q = 2, r = 2), ca3type = "NSCA3")
plot(ris.nsca3, biptype = "row", plot3d = TRUE)
ris.oca3<-CA3variants(happy, dims = c(p = 3, q = 5, r = 4), ca3type = "OCA3", norder = 3)
plot(ris.oca3, biptype = "tube", firstaxis=4, lastaxis=7)
ris.onsca3<-CA3variants(happy, dims = c(p = 3, q = 5, r = 4), ca3type = "ONSCA3", norder = 3)
plot(ris.onsca3, biptype="row", firstaxis=6, lastaxis=7)
```

print.CA3variants *Print of three-way correspondence analysis results*

Description

This function prints the results of three-way symmetrical or non-symmetrical correspondence analysis. If the input parameter, in `CA3variants`, is `ca3type="CA3"`, the function prints the results of three-way symmetrical correspondence analysis. If the input parameter, in `CA3variants`, is `ca3type="NSCA3"`, the function prints the results of three-way non-symmetrical correspondence analysis. If the input parameter, in `CA3variants`, is `ca3type="OCA3"`, the function prints the results of ordered three-way symmetrical correspondence analysis. If the input parameter, in `CA3variants`, is `ca3type="ONSCA3"`, the function prints the results of ordered three-way non-symmetrical correspondence analysis. When the input parameter, in `print.CA3variants`, is `digits = 3`, the function prints all the results using three digital numbers.

Usage

```
## S3 method for class 'CA3variants'
print(x, printall= FALSE, digits = 3,...)
```

Arguments

<code>x</code>	The name of the output of the main function <code>CAvariants</code> .
<code>printall</code>	The logical parameter that specifies if to print all the results or some of them. By default, <code>printall = FALSE</code> .
<code>digits</code>	The input parameter specifying the digital number. By default, <code>digits = 3</code> .
<code>...</code>	Further arguments passed to or from other methods.

Value

The value of output returned depends on the kind of three-way correspondence analysis variant performed. It also gives the number of the iteration of the algorithm to reach the convergence of the solution. Depending on the variant of three-way correspondence analysis performed, it gives the related weighted contingency table, the reconstructed table by the components and core array, the explained inertia, the total inertia, the inertia in percentage, the proportion of explained inertia given the defined number of the components, the row standard and principal coordinates, the interactive column-tube standard and principal coordinates, the inner-product matrix of coordinates, the core array and index partitioning. In detail:

<code>CA3variants</code>	The output of the kind of three-way correspondence analysis analysis considered.
<code>Data</code>	The original three-way contingency table.
<code>xs</code>	The centred and weighted three-way contingency table when the input parameters are <code>ctr=T</code> and <code>std=T</code> .
<code>xhat</code>	The three-way contingency table approximated (reconstructed) by the three component matrices (of dimension $I \times p$, $J \times q$, and $K \times r$) and the core array.
<code>nxhat2</code>	The sum of squares of the approximated contingency table.
<code>prp</code>	The ratio between the inertia of the complete contingency table and the inertia of the approximated contingency table.
<code>fi</code>	The principal row coordinates.

fiStandard	The standard row coordinates.
gjk	The principal column-tube coordinates.
gjkStandard	The standard column-tube coordinates.
fj	The principal column coordinates.
fjStandard	The standard column coordinates.
gik	The principal row-tube coordinates.
gikStandard	The standard row-tube coordinates.
fk	The principal tube coordinates.
fkStandard	The standard tube coordinates.
gij	The principal row-column coordinates.
gijStandard	The standard row-column coordinates.
rows	The row marginals of the three-way data table.
cols	The column marginals of the three-way data table.
tubes	The tube marginals of the three-way data table.
flabels	The row category labels.
glabels	The column category labels.
maxaxes	The maximum dimension to consider.
inertia	The total inertia of a variant of three-way correspondence analysis.
inertiaRSS	The residual inertia of a variant of three-way correspondence analysis.
inertiapc	The percentage contribution of the three components to the total variation.
inertiacoltub	The vector of the percentage contributions of the interactively coded column-tube components to the total inertia, useful for making interactively coded biplots.
inertiarow	The vector of the percentage contributions of the row components to the total inertia, useful for making response biplots.
iproduct	The inner product between the standard row coordinates (fi) and the column-tube principal coordinates (gjk).
g	The core array (i.e. the generalized singular values) calculated by using the Tuckals3 algorithm.
index3	When ca3type = "CA3" or ca3type = "OCA3", the index3 represents the partition of the Pearson index into three two-way association terms and one three-way association term. It also shows the C statistic of each term, its degrees of freedom and p-value. If ca3type = "NSCA3" or ca3type = "ONSCA3", the index3 returns the partition of the Marcotorchino index into three two-way association terms and one three-way association term. It also shows the C statistic of each term, its degrees of freedom and p-value.
ca3type	The specification of the analysis to be performed. When ca3type = "CA3", then a three-way symmetrical correspondence analysis will be performed (default analysis). If ca3type = "NSCA3", then three-way non-symmetrical correspondence analysis will be performed, where one of the variables is the response to be predicted given the other two variables. These two three-way variants use the Tucker3 method of decomposition. When ca3type = "OCA3" or ca3type

= "ONSCA3", then an ordered three-way symmetrical or non-symmetrical correspondence analysis will be performed. Differently, these analysis use a new method of decomposition called Trivariate Moment Decomposition.

`iteration` The number of iterations that are required for the Tuckals3 algorithm to converge.

Author(s)

Rosaria Lombardo, Eric J Beh and Michel van de Velden.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons

Examples

```
data(happy)
ris.ca3<-CA3variants(happy, dims= c(p=2,q=2,r=2), ca3type = "CA3")
print(ris.ca3)
ris.nsca3<-CA3variants(happy, dims = c(p=2,q=2,r=2), ca3type = "NSCA3")
print(ris.nsca3)
ris.oca3<-CA3variants(happy, dims = c(p=3,q=5,r=4), ca3type = "OCA3",norder=3)
print(ris.oca3)
ris.onsca3<-CA3variants(happy, dims = c(p=3,q=5,r=4), ca3type = "ONSCA3",norder=3)
print(ris.onsca3)
```

prod3

Products among arrays

Description

The function calculates the products among arrays.

Usage

```
prod3(m, a1, a2, a3)
```

Arguments

`m` The original three-way contingency table.
`a1` The weight matrix related to the rows of the table.
`a2` The weight matrix related to the columns of the table.
`a3` The weight matrix related to the tubes of the table.

Details

It is utilised in `standtab`, `rstand3` and `rstand3delta` in order to weight the contingency table with respect to the three weight matrices defined in the row, column and tube spaces differently for the three variants of three-way correspondence analysis.

Value

The three-way contingency table weighted with respect the suitable weight matrices (depending on the analysis variants).

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). *Correspondence Analysis, Theory, Practice and New Strategies*. John Wiley & Sons

reconst3

Reconstruction of the three-way centred profile table

Description

The function reconstructs the three-way centred profile table using the component matrices from the Tucker3 decomposition and the core array.

Usage

```
reconst3(param)
```

Arguments

`param` The matrices of the row, column and tube components derived via the Tucker3 model.

Value

The three-way reconstructed table of centred profiles.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh E J and Lombardo R (2014). *Correspondence Analysis, Theory, Practice and New Strategies*. John Wiley & Sons.

rstand3	<i>Weighted centred three-way table for three-way non-symmetric correspondence analysis</i>
---------	---

Description

The function computes the three-way weighted centred contingency table to perform three-way non-symmetric correspondence analysis with one response and two predictors.

Usage

```
rstand3(x, std = T, ctr = T)
```

Arguments

x	The original three-way contingency table.
std	The flag parameter for weighting the original table. If std=F the original contingency table is not weighted.
ctr	The flag parameter for centering the original table. If ctr=F the original array is not centered.

Value

xs	The weighted array with respect to the three associated metrics. It is used when CA3variants="NSCA" and represents the three-way weighted and centred column profile table.
----	---

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

simulabootsimple	<i>Generation of parametric bootstrap samples</i>
------------------	---

Description

This function allows to generate parametric bootstrap samples in order to check the optimal dimension number of three-way correspondence analysis. The bootstrap samples have the same marginal proportions and the total number of the original table. The adopted sampling scheme is simple.

Usage

```
simulabootsimple(Xtable,nboots=100,resamptype=1)
```

Arguments

Xtable	The three-way data array. It must be an R object array. When non-symmetrical analysis for one response variable is performed, the response mode is the row variable.
nboots	The number of bootstrap samples to generate when boots = T. By default nboots = 0.
resamptype	Set value of resamptype according to two methods: resamptype=1 corresponds to multinomial distribution and resamptype=2 to Poisson distribution.

Author(s)

Michel van de Velden, Rosaria Lombardo and Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. Wiley.

simulabootstrat	<i>Generation of parametric bootstrap samples</i>
-----------------	---

Description

This function allows to generate parametric bootstrap samples in order to check the optimal dimension number of three-way correspondence analysis. The bootstrap samples have the same marginal proportions and total number of the original table. The adopted sampling scheme is stratified.

Usage

```
simulabootstrat(Xtable,nboots=100,resamptype=1)
```

Arguments

xtable	The three-way data array. It must be an R object array. When non-symmetrical analysis for one response variable is performed, the response mode is the row variable.
nboots	The number of bootstrap samples to generate when boots = T. By default nboots = 0.
resamptype	Set value of resamptype according to two methods: resamptype=1 corresponds to multinomial distribution and resamptype=2 to Poisson distribution.

Author(s)

Rosaria Lombardo, Michel van de Velden, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. Wiley.

standtab	<i>Three-way centred column profile table for the three-way symmetric correspondence analysis</i>
----------	---

Description

The function computes the three-way centred column profile table to perform three-way symmetric correspondence analysis.

Usage

```
standtab(x, std = T, ctr = T)
```

Arguments

x	The original three-way contingency table.
std	The flag parameter for weighting the original table. If F the original contingency table is not weighted.
ctr	The flag parameter for centering the original table. If F the original array is not centered.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

step.g3	<i>The core array derived via the Tucker3 model.</i>
---------	--

Description

The Tucker3 model involves the computation of principal components, which are derived for each of the three categorical variables, and of the core array which is akin to the generalised correlations between these components. The function `step.g3` computes the core array.

Usage

```
step.g3(param)
```

Arguments

param	The weighted three-way table and the matrices of the row, column and tube components derived via the Tuckals3 algorithm.
-------	--

Value

The core matrix whose the general element can be interpreted as a generalized singular value.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

step.g3ordered	<i>The core array derived via the Trivariate Moment Decomposition model.</i>
----------------	--

Description

The Trivariate Moment Decomposition model involves the computation of principal polynomial components, which are derived for each of the three categorical variables, and of the polynomial core array which is akin to the generalised correlations between these components. The function `step.g3ordered` computes the core array.

Usage

```
step.g3ordered(param)
```

Arguments

param The weighted three-way table and the matrices of the row, column and tube components derived via the Trivariate Moment Decomposition algorithm.

Value

The core matrix whose the general element can be interpreted as a generalized singular value.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

stepi3

Component matrices from the Tucker3 decomposition

Description

The function computes the component matrices from the Tuckals3 algorithm.

Usage

stepi3(param)

Arguments

param The weighted contingency table and the matrices of the row, column and tube components derived via the Tucker3 model.

Details

The functions newcomp3, stepi3, init3 and step.g3 compute the component matrices and core array in the iterative steps of Tuckals3. They are all utilised from the function tucker.

Value

Component matrices from the Tucker3 decomposition.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

stepi3ordered	<i>Component matrices from the Trivariate Moment Decomposition decomposition</i>
---------------	--

Description

The function computes the polynomial component matrices from the Emerson's recurrence formula for the ordered categorical variables of the analysis.

Usage

```
stepi3ordered(param)
```

Arguments

param	The weighted contingency table and the matrices of the row, column and tube components derived via the Trivariate Moment Decomposition model.
-------	---

Details

The functions `newcomp3ordered`, `stepi3ordered`, `init3ordered` and `step.g3ordered` compute the polynomial component matrices and core array in the Trivariate Moment Decomposition. They are all utilised from the function `tuckerORDERED`.

Value

Component matrices from the Trivariate Moment Decomposition decomposition.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

`tau3`*Partition of the Marcotorchino three-way index*

Description

When the association among three categorical variables is asymmetric such that one variable is a logical response variable to the other variables, we recommend calculating the non-symmetrical three-way measure of predictability such as the Marcotorchino index (Marcotorchino, 1985). The function `tau3` partitions the Marcotorchino statistic when, in `CA3variants`, we set the parameter `ca3type = "NSCA3"`.

Usage

```
tau3(f3, digits = 3)
```

Arguments

<code>f3</code>	Three-way contingency array given as an input parameter in <code>CA3variants</code> .
<code>digits</code>	Number of decimal digits. By default <code>digits=3</code> .

Value

<code>z</code>	The partition of the Marcotorchino index into three two-way association terms and one three-way association term. It also shows the C statistic of each term, its degrees of freedom and p-value.
<code>CM</code>	the C statistic of the Marcotorchino index.
<code>devt</code>	The denominator of the Marcotorchino index.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

Examples

```
data(happy)
tau3(happy, digits=3)
```

tau3ordered	<i>The partition of the Marcotorchino three-way index.</i>
-------------	--

Description

When three categorical variables are symmetrically related, we can analyse the strength of the association using the three-way Marcotorchino index. The function `chi3` partitions the Marcotorchino statistic using orthogonal polynomials when, in `CA3variants`, we set the parameter `ca3type = "ONSCA3"`.

Usage

```
tau3ordered(f3, digits = 3)
```

Arguments

<code>f3</code>	The three-way contingency array given as an input parameter in <code>CA3variants</code> .
<code>digits</code>	The number of decimal digits. By default <code>digits=3</code> .

Value

The partition of the Marcotorchino index into three two-way non-symmetrical association terms and one three-way association term. It also shows the polynomial components of inertia, the percentage of explained inertia, the degrees of freedom and p-value of each term of the partition.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). *Correspondence Analysis, Theory, Practice and New Strategies*. John Wiley & Sons.

Examples

```
data(olive)
tau3ordered(f3=olive, digits=3)
```

threewayboot	<i>Generation of non-parametric bootstrap samples</i>
--------------	---

Description

This function allows to generate non-parametric bootstrap samples in order to check the optimal dimension number of three-way correspondence analysis. The bootstrap samples have the same marginal proportions and the total number of the original table. Do nboots bootstrap on the indicator matrix X (observations x (rows+cols+tubs) categories). From a three-way contingency table, it makes the indicator using makeindicator. The output is a list of three-way tables.

Usage

```
threewayboot(Xdata,nboots=100)
```

Arguments

Xdata	The three-way contingency array. It must be an R object array.
nboots	The number of bootstrap samples to generate when boots = T. By default nboots = 0.

Author(s)

Rosaria Lombardo, Michel van de Velden, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

tucker	<i>Tucker3 decomposition of the three-way table.</i>
--------	--

Description

The Tucker3 model, originally proposed by psychologist Ledyard R. Tucker, involves the computation of principal components, which are derived for each of the three categorical variables, and of the core array which is akin to the generalised correlations between these components. The function represents the heart of the Tuckals3 algorithm to perform the Tucker3 decomposition of the three-way array x.

Usage

```
tucker(x, p, q, r, test = 10^-6)
```


Arguments

x	The three-way contingency table.
p	The number of components of the first mode.
q	The number of components of the second mode.
r	The number of components of the third mode.
test	The treshold used in the algorithm.

Details

The function tucker is utilised from the functions ca3basic, nsca3basic and oca3basic.

Value

a	The final component derived from the Tucker3 decomposition for the first mode.
b	The final component derived from the Tucker3 decomposition for the second mode.
cc	The final component derived from the Tucker3 decomposition for the third mode.
g	The core array.
x	The three-way contingency table.
cont	The number of iterations that are required for the Tucker3 algorithm to converge.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.

tuckerORDERED	<i>Trivariate moment decomposition of the three-way table.</i>
---------------	--

Description

The Trivariate moment decomposition (TMD) represents the heart of a new algorithm to perform the decomposition of the three-way ordered contingency tables. It is based on the orthogonal polynomials (Emerson 1968) computed for each categorical ordered variable.

Usage

```
tuckerORDERED(x, p, q, r, test = 10^-6, xi, norder=3)
```

Arguments

x	The three-way contingency table.
p	The number of components of the first mode.
q	The number of components of the second mode.
r	The number of components of the third mode.
test	The treshold used in the algorithm.
xi	The original three-way contingency table.
norder	The number of ordered variables.

Details

The function tuckerORDERED is utilised from the function oca3basic.

Value

a	The final component derived from the TMD decomposition for the first mode.
b	The final component derived from the TMD decomposition for the second mode.
cc	The final component derived from the TMD decomposition for the third mode.
g	The core array.
x	The three-way contingency table.
cont	The number of iterations that are required for the TMD algorithm to converge. If all variables are ordered, the convergence is reached in one step, differently if we have mixed variables. Indeed, the decomposition will become hybrid, a mix of TMD algorithm and Tuckals3 algorithm.

Author(s)

Rosaria Lombardo, Eric J Beh.

References

- Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.
- Emerson PL 1968 Numerical construction of orthogonal polynomials from a general recurrence formula. *Biometrics*, 24 (3), 695-701.
- Lombardo R Beh EJ Variants of Simple Correspondence Analysis. *The R Journal*, 8 (2), 167–184.
- Lombardo R Beh EJ and Kroonenberg PM (2016) Modelling Trends in Ordered Correspondence Analysis Using Orthogonal Polynomials. *Psychometrika*, 325–349.

tunelocal	<i>Dimension selection for three-dimensional correspondence biplot using convex hull.</i>
-----------	---

Description

This function allows to select the optimal dimension number for correspondence biplot, given the set of possible dimension combination of the original data. For exploring, it is also possible to check the optimal model dimension by using bootstrap samples which have the same marginal proportions and the total number of the original table. When the input parameter `boots = T`, it does bootstrap sampling. There are three kinds of possible bootstrap sampling. When `boottype = "bootnp"` it performs a non parametric bootstrap sampling. When `boottype = "bootpsimple"` it performs a parametric simple bootstrap sampling. When `boottype = "bootpstrat"`, it performs a parametric stratified bootstrap sampling. In particular in case of parametric bootstrap types, when `resamptype=1` it considers a multinomial distribution, and when `resamptype = 2` it considers a poisson distribution.

Usage

```
tunelocal(Xdata, ca3type = "CA3", resp = "row", norder = 3, digits = 3, boots = FALSE,
          nboots = 0, boottype= "bootpsimple", resamptype = 1)
```

Arguments

<code>Xdata</code>	The three-way data. It can be a R object array or raw data (n individuals by three categorical variables, for an example, see museum data). When a three-way non-symmetrical variant is performed, by default, the response variable is the row variable when an array is given, or the first of three columns when a raw data set is given. For changing, consider the parameter <code>resp = "col"</code> or <code>resp = "tube"</code> .
<code>ca3type</code>	The specification of the analysis to be performed. If <code>ca3type = "CA3"</code> , then a three-way (symmetrical) correspondence analysis will be performed (default analysis). If <code>ca3type = "NSCA3"</code> , then three-way non-symmetrical correspondence analysis will be performed. If <code>ca3type = "OCA3"</code> , then ordered three-way symmetrical correspondence analysis will be performed. If <code>ca3type = "ONSCA3"</code> , then ordered three-way non-symmetrical correspondence analysis will be performed.
<code>resp</code>	The input parameter for specifying in non-symmetrical three-way correspondence analysis variants (<code>ca3type = "NSCA3"</code> and <code>ca3type = "ONSCA3"</code>) what is the response variable (logically antecedent to the others). By default <code>resp = "row"</code> , but it could be the column variable <code>resp = "col"</code> or the tube variable <code>resp = "tube"</code> .
<code>norder</code>	The input parameter for specifying the number of ordered variable when <code>ca3type = "OCA3"</code> or <code>ca3type = "ONSCA3"</code> .
<code>digits</code>	The input parameter specifying the digital number. By default, <code>digits = 3</code> .
<code>boots</code>	The flag parameter to perform the search of optimal dimensions using bootstrap samples. By defaults, <code>boots = FALSE</code> .

nboots	The number of bootstrap samples to generate when boots = TRUE. By default nboots = 0.
boottype	The specification of the kind of bootstrap sampling to be performed. If boottype = "bootpsimple", then a parametric bootstrap using a simple sampling scheme will be performed (default sampling). If boottype = "bootpstrat", then a parametric bootstrap using a stratified sampling scheme will be performed. If boottype = "bootnp", then a non-parametric bootstrap using a simple sampling scheme will be performed.
resamptype	When the kind of bootstrap is parametric you can set the data distribution using the input parameter resamptype according to two distribution: resamptype=1 corresponds to multinomial distribution and resamptype=2 to Poisson distribution.

Value

XG	The list of tables on which is performed the three-way CA variant. It consists of the original array and (when boots=T) bootstrapped arrays.
output1	Chi-square criterion and df of models on the convex hull when using the original array.
output2	Chi-square criterion and df of models on the convex hull when using bootstrapped arrays.
output3	Badness of fit criterion and df of models on the convex hull when using the original array.
output4	Badness of fit criterion and df of models on the convex hull when using bootstrapped arrays.
output5	Goodness of fit criterion and df of models on the convex hull when using the original array.
output6	Goodness of fit criterion and df of models on the convex hull when using bootstrapped arrays.

Author(s)

Rosaria Lombardo, Michel van de Velden, Eric J Beh.

References

Beh EJ and Lombardo R (2014). Correspondence Analysis, Theory, Practice and New Strategies. John Wiley & Sons.
 Wilderjans T F, Ceulemans E, and Meers K (2013). CHull: A generic convex hull based model selection method. Behavior Research Methods, 45, 1-15.
 Ceulemans E, and Kiers H A L (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. British Journal of Mathematical & Statistical Psychology, 59, 133-150.

Examples

```
data(happy)
tunelocal(Xdata = happy, ca3type = "CA3", boots = FALSE)
```


Index

- * **Three-way correspondence analysis**
 - ca3basic, 3
- * **Trivariate Moment Decomposition**
 - init3ordered, 16
 - init3ordered1, 17
- * **Tuckals3**
 - stepi3, 44
- * **Tucker3 components**
 - newcomp3, 26
 - newcomp3ordered, 27
 - newcomp3ordered1, 27
 - newcomp3ordered2, 28
- * **Tucker3**
 - init3, 15
 - init3ordered, 16
 - init3ordered1, 17
 - init3ordered2, 18
- * **Tucklas3**
 - init3, 15
 - init3ordered2, 18
 - stepi3ordered, 45
 - tucker, 48
 - tuckerORDERED, 49
- * **array**
 - margI, 22
 - margJ, 23
 - margK, 24
- * **biplot**
 - ca3plot, 4
- * **bootstrap**
 - simulabootsimple, 41
 - simulabootstrat, 41
 - threewayboot, 48
 - tunelocal, 51
- * **component**
 - criteria, 12
- * **core**
 - step.g3, 43
 - step.g3ordered, 43
- * **criterion**
 - coord, 10
 - coordnsc3, 10
 - criter, 11
 - loss1.3, 19
 - loss1.3ordered, 20
- * **datasets**
 - happy, 14
 - museum, 24
 - olive, 31
- * **dimension**
 - simulabootsimple, 41
 - simulabootstrat, 41
 - threewayboot, 48
 - tunelocal, 51
- * **graphical display**
 - plot.CA3variants, 34
- * **index**
 - tau3, 46
- * **indicator**
 - makeindicator, 22
- * **interactive biplot**
 - plot.CA3variants, 34
- * **interactive plot**
 - plot.CA3variants, 34
- * **loss**
 - loss2, 21
- * **marginal**
 - margI, 22
 - margJ, 23
 - margK, 24
- * **matrix**
 - makeindicator, 22
- * **multivariate**
 - flatten, 13
 - Kron, 19
 - loss1.3, 19
 - loss1.3ordered, 20
 - newcomp3, 26

- newcomp3ordered, 27
- newcomp3ordered1, 27
- newcomp3ordered2, 28
- nsca3basic, 29
- print.CA3variants, 35
- reconst3, 39
- rstand3, 40
- standtab, 42
- step.g3, 43
- step.g3ordered, 43
- stepi3, 44
- stepi3ordered, 45
- tucker, 48
- tuckerORDERED, 49
- * **nonparametric**
 - emerson.poly, 12
- * **ordered**
 - oca3basic, 30
 - onsca3basic, 32
- * **partition**
 - print.CA3variants, 35
 - tau3, 46
- * **plot**
 - caplot3d, 7
- * **product**
 - Kron, 19
 - p.ext, 33
 - prod3, 38
- * **reconstruction**
 - reconst3, 39
- * **three way index**
 - chi3ordered, 9
 - tau3ordered, 47
- * **three-way index**
 - chi3, 8
- * **three-way non-symmetric correspondece analysis**
 - nsca3basic, 29
- * **three-way non-symmetrical correspondence analysis.**
 - CA3variants, 5
- * **three-way symmetrical correspondence analysis.**
 - CA3variants, 5
- * **tuckals3**
 - p.ext, 33
- * **tucker**
 - criteria, 12
- * **weighted array**
 - prod3, 38
- * **weight**
 - rstand3, 40
- ca3basic, 3
- ca3plot, 4
- CA3variants, 5
- caplot3d, 7
- chi3, 8
- chi3ordered, 9
- coord, 10
- coordnsc3, 10
- criter, 11
- criteria, 12
- emerson.poly, 12
- flatten, 13
- happy, 14
- init3, 15
- init3ordered, 16
- init3ordered1, 17
- init3ordered2, 18
- Kron, 19
- loss1.3, 19
- loss1.3ordered, 20
- loss2, 21
- makeindicator, 22
- margI, 22
- margJ, 23
- margK, 24
- museum, 24
- newcomp3, 26
- newcomp3ordered, 27
- newcomp3ordered1, 27
- newcomp3ordered2, 28
- nsca3basic, 29
- oca3basic, 30
- olive, 31
- onsca3basic, 32
- p.ext, 33
- plot.CA3variants, 34

print.CA3variants, 35
prod3, 38

reconst3, 39
rstand3, 40

simulabootsimple, 41
simulabootstrat, 41
standtab, 42
step.g3, 43
step.g3ordered, 43
stepi3, 44
stepi3ordered, 45

tau3, 46
tau3ordered, 47
threewayboot, 48
tucker, 48
tuckerORDERED, 49
tunelocal, 51