Package 'CA3variants'

September 20, 2022

Type Package

Title Three-Way Correspondence Analysis Variants

Version 3.3

Date 2022-09-20

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)

Imports ggplot2, plotly, tools, ggforce, gridExtra, ggrepel, multichull, utils

LazyData true

License GPL (> 2)

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

NeedsCompilation no

Repository CRAN

Date/Publication 2022-09-20 16:10:01 UTC

R topics documented:

`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, sign = TRUE)
```

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.
`sign`	The input parameter for changing the sign to the components according to the core sign.

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, inertiapc, size1, size2, biptype, addlines)
```

Arguments

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

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. John Wiley & Sons.
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, sign = TRUE)
```

Arguments

Xdata	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 resp = "col" or resp = "tube".

`dims` The number of components for the first, second and third mode. By default, no `dims` 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.

`ca3type` The specification of the analysis to be performed. If `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. If `ca3type = "OCA3"`, then ordered three-way symmetrical correspondence analysis will be performed. If `ca3type = "ONSCA3"`, 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 (`ca3type = "NSCA3"` and `ca3type = "ONSCA3"`) what is the response variable (logically antecendent to the others). By default, `resp = "row"`, but it could be `resp = "col"` or `resp = "tube"`.

`norder` The input parameter for specifying the number of ordered variable when `ca3type = "OCA3"` or `ca3type = "ONSCA3"`. By default, all three variables are ordered `norder = 3`. When `norder = 1`, you assume that the ordered variable is the column variable. When `norder = 2`, you assume that the ordered variables are the row and column variable.

`sign` The input parameter for changing the sign to the components according to the core sign.

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(ratrank)
CA3variants(Xdata = ratrank, dims = c(p=2,q=2,r=1), ca3type = "CA3")
data(happy)
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 = "OCA3")
CA3variants(Xdata = happy, dims = c(p=3,q=5,r=4), ca3type = "ONSCA3")
```

---

caplot3d — *Three dimensional correspondence plot*

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, thirdaxis = 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`.
`thirdaxis`	The third axis number. By default, `thirdaxis = 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. John Wiley & Sons.
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

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

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 PM (1996) Decompositions and biplots in three-way correspondence analysis. Psychometrika, 61, 355-373.

Examples

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

---

chi3ordered *The partition of the Pearson three-way index.*

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

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

Value

The partition of the Pearson index into three two-way association terms and one three-way association term. It also shows the polynomial componets 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)
```

chkneg — *Check the sign of component values*

Description

This function is called from `signscore`. It checks the negativity of the column of an array AND the positivity of the columns of an array If NegPtr = 1 then there is an entirely negative component If PosPtr = 1 then there is an entirely positive component If BigPtr = 1 then maximum neg. abs > max pos

Usage

```
chkneg(comp, nr, nc)
```

Arguments

`comp`	One of three component matrix.
`nr`	The row number of the component matrix.
`nc`	The column number of the component matrix.

Author(s)

Rosaria Lombardo and Pieter M Kroonenberg.

References

Kroonenberg PM (1994) The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.

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.

critera *Criterion of the Tucker3 algorithm*

Description

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

Usage

```
critera(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.

---

crptrs — *Pointing to the columns of the component matrices.*

Description

Given ICORE, i.e. the pointer to an element in CORE(p,q,r), this subroutine calculates the IA, IB and IC, pointing to the columns of the component matrices A, B and C that are responsible for the value in CORE(ICORE).

Usage

```
crptrs(icore, p, q, r)
```

Arguments

`icore`	The pointer to the core elements whose sign should be reversed.
`p`	The dimension number of the first mode.
`q`	The dimension number of the second mode.
`r`	The dimension number of the third mode.

Value

`IA`	The pointer to the columns of the first component matrix, given the pointer to an element in core.
`IB`	The pointer to the columns of the second component matrix, given the pointer to an element in core.
`IC`	The pointer to the columns of the third component matrix, given the pointer to an element in core.

Author(s)

Rosaria Lombardo and Pieter M Kroonenberg.

References

Kroonenberg PM (1994) The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.

`emerson.poly` *Orthogonal polynomials*

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 (2016) Variants of Simple Correspondence Analysis. The R Journal, 8 (2), 167–184.

`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 was generated from the database of the European Social Survey 2016. The variables that we selected for our analysis are Education, Households and Happiness.

Usage

```
data(happy)
```

Format

The format is:

row names [1:4] "ED1", "ED2", "ED3", "ED45" col names [1:6] "HS1", "HS2", "HS3", "HS4", "HS5", "HS>5" tube names [1:4] "low", "middle", "high", "very-high"

References

Lombardo R, van de Velden M and Beh E J (2022) Three-way Correspondence Analysis in R. (submitted)

Examples

```
happy <-
structure(c(325, 411, 793, 602, 239, 374, 827, 583, 63,
181, 361, 303, 42, 129, 229, 224, 16, 49, 89, 54,
11, 37, 31, 21, 357, 477, 1049, 929, 327, 610, 1447,
1446, 115, 303, 763, 832, 64, 250, 591, 638, 35, 105,
183, 185, 15, 56, 99, 71, 265, 327, 769, 928, 342,
565, 1461, 1808, 104, 314, 768, 1006, 69, 312, 729,
977, 21, 122, 215, 362, 14, 57, 126, 129, 214, 241,
554, 660, 419, 561, 1467, 1861, 130, 290, 786, 938,
89, 319, 741, 1022, 36, 121, 289, 408, 35, 87, 153,
171), .Dim = c(4, 6, 4), .Dimnames = list(c("ED1",
"ED2", "ED3", "ED45"), c("HS1", "HS2", "HS3", "HS4", "HS5", "HS>5"
), c("low", "middle", "high", "very-high")))
dim(happy)
```

`happyNL` *Raw data: Three variables from a Dutch survey on happiness*

Description

This raw data table represents a possible data set selected from a large survey on happiness. The rows are individuals. The first column concerns four level of happiness, the second column concerns the number of households in a family, and the third column their level of Education.

Usage

```
data(museum)
```

Format

The format is: row names [1:4] "low", "middle", "high", "very-high" col names [1:5] "HS1", "HS2", "HS3", "HS4", ">HS5" tube names [1:4] "ED1", "ED2", "ED3", "ED45"

References

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

Examples

```
happyNL<-structure(c(11L, 12L, 15L, 7L, 2L, 6L, 17L, 13L, 0L, 2L, 4L,
6L, 0L, 5L, 7L, 3L, 0L, 3L, 3L, 1L, 14L, 56L, 52L, 22L, 11L,
39L, 70L, 65L, 1L, 14L, 19L, 14L, 5L, 12L, 16L, 20L, 2L, 3L,
10L, 4L, 14L, 44L, 44L, 15L, 6L, 27L, 79L, 47L, 4L, 17L, 40L,
27L, 2L, 25L, 49L, 38L, 1L, 12L, 12L, 11L, 10L, 41L, 66L, 24L,
4L, 32L, 100L, 90L, 1L, 8L, 40L, 28L, 3L, 15L, 49L, 35L, 1L,
4L, 23L, 15L), .Dim = c(4L, 5L, 4L), .Dimnames = list(happy = c("low",
"middle", "high", "very-high"), hhmmb = c("HS1", "HS2", "HS3",
"HS4", ">HS5"), edulvla = c("ED1", "ED2", "ED3", "ED45")), class = "table")
dim(happyNL)
data(happyNL)
```

`init3` *Initial components from the Tuckals3 algorithm*

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 — *Initial components from the Trivariate Moment Decomposition algorithm*

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` *Initial components from the Trivariate Moment Decomposition algorithm*

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 — *Initial components from the Trivriate Moment Decomposition algorithm*

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

`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 Trivriate Moment Decomposition for the first mode.
`b`	The initial component derived from the Trivriate Moment Decomposition for the second mode.
`cc`	The initial component derived from the Trivriate 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.

invcmp — *Changing component sign*

Description

This function changes the sign of the elements in column of the component matrix.

Usage

```
invcmp(comp, nr, nc, chgcomp)
```

Arguments

`comp`	One of the three component matrices.
`nr`	The row number of the component matrix.
`nc`	The column number of the component matrix.
`chgcomp`	The pointers to the columns of the component matrix that are responsible for the value in the ordered core.

Author(s)

Rosaria Lombardo and Pieter M Kroonenberg.

References

Kroonenberg PM (1994) The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.

invcor — *Changing core sign*

Description

This function is used from the function `signscore`. It changes the sign of the elements of core slice

Usage

```
invcor(core, p, q, r, chgmode, chgcomp)
```

Arguments

`core`	The core array.
`p`	The dimension of the first mode.
`q`	The dimension of the second mode.
`r`	The dimension of the third mode.
`chgmode`	One of the three mode to change.
`chgcomp`	One of the three component to change.

Author(s)

Rosaria Lombardo and Pieter M Kroonenberg.

References

Kroonenberg PM (1994) The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.

Kron — *Kronecker product*

Description

The function performs the Kronecker product. Starting from two matrices of dimension IxP and JxQ the resulting matrix will be of dimension IxJ,PxQ.

Usage

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

Arguments

a	The first matrix of dimension IxP involved in the kronecker product.
b	The second matrix of dimension JxQ 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 — *General loss criterion*

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 — *General loss criterion*

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 Trivriate 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` *Make an Indicator matrix*

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.

`margI` *Row marginals of a three-way contingency table*

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 satisfacyion 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" ...

References

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

Examples

```
museum<-structure(list(nvis = structure(c(2L, 2L, 4L, 4L, 1L, 3L, 3L,
2L, 4L, 1L, 3L, 3L, 4L, 2L, 4L, 3L, 4L, 2L, 2L, 3L, 4L, 4L, 2L,
4L, 3L, 4L, 2L, 2L, 4L, 1L, 2L, 2L, 4L, 1L, 4L, 2L, 2L, 2L, 4L,
1L, 1L, 1L, 1L, 2L, 2L, 3L, 2L, 3L, 4L, 4L, 1L, 3L, 2L, 2L, 3L,
3L, 3L, 2L, 4L, 3L, 2L, 4L, 2L, 3L, 3L, 3L, 3L, 3L, 3L, 2L, 3L,
3L, 3L, 3L, 2L, 2L, 4L, 4L, 4L, 4L, 3L, 2L, 3L, 3L, 3L, 4L, 2L,
2L, 2L, 4L, 1L, 1L, 1L, 1L, 2L, 2L, 3L, 2L, 3L, 4L, 4L, 1L, 3L,
3L, 2L, 4L, 3L, 2L, 4L, 3L, 2L, 4L, 2L, 3L, 3L, 2L, 2L, 2L, 3L,
2L, 3L, 2L, 2L, 3L, 2L, 2L, 4L, 4L, 4L, 4L, 4L, 3L, 2L, 3L, 3L,
3L, 4L, 4L, 1L, 3L, 3L, 2L, 1L, 1L, 1L, 1L, 3L, 4L, 2L, 4L, 3L,
4L, 2L, 2L, 3L, 4L, 2L, 3L, 3L, 3L, 4L, 2L, 2L, 2L, 4L, 1L, 3L,
1L, 1L, 2L, 2L, 3L, 2L, 3L, 3L, 3L, 1L, 3L, 2L, 2L, 2L, 1L, 1L,
2L, 2L, 2L, 1L, 3L, 2L, 3L, 4L, 4L, 1L, 3L, 2L, 2L, 2L, 3L, 2L,
3L, 4L, 4L, 1L, 3L, 3L, 3L, 2L, 1L, 4L, 1L, 3L, 4L, 3L, 4L, 2L,
4L, 3L, 4L, 2L, 2L, 3L, 3L, 4L), .Label = c("no", "often", "some",
"voften"), class = "factor"), like = structure(c(2L, 2L, 2L,
2L, 2L, 3L, 3L, 2L, 2L, 2L, 3L, 3L, 2L, 3L, 1L, 3L, 2L, 3L, 3L,
1L, 3L, 2L, 3L, 2L, 3L, 2L, 2L, 3L, 2L, 3L, 3L, 3L, 3L, 3L, 2L,
2L, 2L, 2L, 2L, 2L, 2L, 2L, 3L, 2L, 2L, 3L, 2L, 3L, 2L, 2L, 2L,
2L, 2L, 2L, 1L, 2L, 2L, 2L, 1L, 3L, 3L, 2L, 3L, 3L, 2L, 3L, 2L,
3L, 2L, 2L, 3L, 2L, 3L, 2L, 3L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 3L, 3L, 2L, 3L, 3L, 3L,
2L, 2L, 2L, 2L, 2L, 2L, 2L, 3L, 3L, 2L, 3L, 3L, 2L, 3L, 3L, 2L,
3L, 2L, 3L, 3L, 2L, 3L, 2L, 3L, 2L, 2L, 2L, 2L, 3L, 3L, 2L, 2L,
2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 3L, 3L, 2L, 1L, 2L, 2L, 3L, 3L,
2L, 3L, 1L, 2L, 2L, 3L, 3L, 1L, 3L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 3L, 2L, 2L, 3L, 2L, 2L, 3L, 2L, 3L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 2L, 2L, 2L, 3L, 2L, 2L, 3L, 2L, 3L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 1L, 2L, 2L, 2L, 2L, 2L, 2L, 3L, 3L, 2L, 1L, 2L, 2L, 3L,
3L, 3L, 2L, 3L, 1L, 3L, 2L, 3L, 3L, 1L, 3L, 3L), .Label = c("little",
"much", "some"), class = "factor"), satisfaction = structure(c(1L,
2L, 2L, 1L, 1L, 2L, 2L, 1L, 3L, 1L, 3L, 1L, 1L, 2L, 2L, 1L, 2L,
2L, 2L, 2L, 2L, 1L, 4L, 2L, 2L, 3L, 1L, 2L, 1L, 1L, 3L, 3L, 1L,
1L, 2L, 1L, 1L, 1L, 2L, 2L, 1L, 4L, 3L, 1L, 1L, 2L, 2L, 2L, 2L,
2L, 2L, 2L, 2L, 1L, 2L, 1L, 2L, 1L, 2L, 2L, 3L, 1L, 2L, 3L, 2L,
3L, 2L, 3L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 1L, 1L, 4L, 3L, 1L, 1L, 2L,
2L, 2L, 2L, 2L, 2L, 1L, 2L, 1L, 2L, 2L, 3L, 1L, 2L, 3L, 1L, 2L,
3L, 2L, 3L, 2L, 3L, 2L, 1L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 1L, 3L, 3L, 1L,
3L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 2L, 2L, 1L, 4L, 2L, 1L, 1L, 2L, 2L, 2L, 2L, 2L, 2L,
1L, 1L, 1L, 2L, 1L, 1L, 4L, 3L, 1L, 1L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 1L, 1L, 2L, 1L, 2L, 1L, 2L, 1L, 2L, 2L, 2L, 1L, 3L, 3L,
1L, 3L, 4L, 1L, 1L, 2L, 2L, 1L, 2L, 2L, 2L, 2L, 3L, 4L), .Label = c("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 `init3` 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 `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.

`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, sign = TRUE)
```

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.
`ctr`	The flag parameter to center the data (T or F), if F the data are not centered.
`std`	The flag parameter to weight the data (T or F), if F the data are not weighted.
`sign`	The input parameter for changing the sign to the components according to the core sign.

Value

`x`	The original three-way contingency table.
`xs`	The weighted three-way contingency table.
`xhat`	The three-way contingency table reconstructed after Tuckals3 by means of the principal components and core array.
`nxhat2`	The inertia of the three-way non-symmetrical correspondence analysis for one response (the three-way Marcotorchino index).
`prp`	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 p, q, r principal components with respect to the overall fit.
`a`	The row principal components.
`b`	The column principal components.
`cc`	The tube principal components.

`g`	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.

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 `CA3variants` 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, sign = TRUE)
```

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.
`norder`	The number of ordered variables considered.
`sign`	The input parameter for changing the sign to the components according to the core sign.

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.

`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 `CA3variants` 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, sign = TRUE)
```

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.
`norder`	The number of ordered variables considered.
`sign`	The input parameter for changing the sign to the components according to the core sign.

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* `CA3variants`

Description

The function `plot.CA3variants` allows the analyst to graphically display six types of biplots for symmetrical 3-way variants and two types of biplots for non-symmetrical 3-way variants. The six types of biplots for `CA3` and `OCA3` are the following. When the input parameter is `biptype = "column-tube"` (or `biptype = "col-tube"`), the function displays the column-tube interactive biplot, where the column and tube variables are coded interactively and have principal coordinates and the row variable has standard coordinates. When the input parameter is `biptype = "row"`, the function displays the row biplot, where the rows have principal coordinates. When the input parameter is `biptype = "col"`, the function displays the column biplot, where the columns have principal coordinates. When the input parameter is `biptype = "row-tube"`, the function displays the row-tube biplot, where the row-tubes have principal coordinates. When the input parameter is `biptype = "tube"`, the function displays the tube biplot, where the tubes have principal coordinates When the input parameter is `biptype = "row-column"` (or `biptype = "row-col"`), the function displays the row-column interactive biplot, where the row-columns have principal coordinates. The two types of biplots for `NSCA3` and `ONSCA3` are the following. When the input parameter is `biptype = "pred"`, the function displays the biplot where the predictors are coded interactively and have principal coordinates and the response has standard coordinates. When the input parameter is `biptype = "resp"`, the function displays the biplot where the response variable has principal coordinates and the predictors (interactively coded) have standard coordinates.

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 = 3, addlines = TRUE,...)
```

Arguments

`x`	The output parameters of the main function `CA3variants`.
`firstaxis`	The dimension reflected along the horizontal axis.
`lastaxis`	The dimension reflected along the vertical axis.
`thirdaxis`	The dimension reflected along the third axis when `plot3d = TRUE`.
`cex`	The parameter that specifies the size of character labels of points in graphical displays. By default, it is equal to 0.8.
`biptype`	The input parameter for specifying what kind of biplot is requested. By default, it is equal to `column-tube`, but could be `row-tube`, `row-column`, `row`, `column` and `tube`.
`scaleplot`	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).
`plot3d`	The logical parameter specifies whether a 3D plot is to be included in the output or not. By default, `plot3d = FALSE`.
`pos`	The input parameter for changing the label position. By default, it is equal to 1.
`size1`	The input parameter for specifying the size of pointers. By default, it is equal to 1.
`size2`	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 (2021) Symmetrical and Non-Symmetrical Variants of Three-Way Correspondence Analysis for Ordered Variables. Statistical Science Journal, p. 1-33.

Examples

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

---

plot.tunelocal — *Graphical display resulting from* tunelocal

---

Description

The function plot.tunelocal allows the analyst to graphically display the optimal model dimension using a convex hull.

Usage

```
## S3 method for class 'tunelocal'
plot(x,...)
```

Arguments

x	The results of the function tunelocal. It shows the models that are located on the boundary of the convex hull and selects an optimal model by means of the scree test values (st). When using boots=F, it gives the set of possible dimension combination of the original data using only the original data array. When using boots=T, it gives the set of possible dimension combination of the original data using bootstrapped data arrays.
...	Further arguments passed to or from other methods.

Value

Graphical displays of a convex hull computed using the original data and the bootstrapped data when in tunelocal the input parameter boot=TRUE computed bootstrapped data too.

Author(s)

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

References

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

```
res.tunelocal<-tunelocal(happy, ca3type = "CA3",boots = FALSE,
          nboots = 0)
plot(res.tunelocal)
```

`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

`x`	The name of the output of the main function `CA3variants`.
`printall`	The logical parameter that specifies if to print all the results or some of them. By default, `printall = FALSE`.
`digits`	The input parameter specifying the digital number. By default, `digits = 3`.
`...`	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:

`CA3variants`	The output of the kind of three-way correspondence analysis analysis considered.
`Data`	The original three-way contingency table.
`xs`	The centred and weighted three-way contingency table when the input parameters are `ctr=T` and `std=T`.
`xhat`	The three-way contingency table approximated (reconstructed) by the three component matrices (of dimension Ixp, Jxq, and Kxr) and the core array.
`nxhat2`	The sum of squares of the approximated contingency table.
`prp`	The ratio between the inertia of the complete contingency table and the inertia of the approximated contingency table.
`fi`	The principal row coordinates.
`fiStandard`	The standard row coordinates.
`gjk`	The principal colum-tube coordinates.
`gjkStandard`	The standard colum-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-colum coordinates.
`gijStandard`	The standard row-colum 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 colum-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.

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)
```

`print.tunelocal` *Print of tunelocal function results*

Description

This function prints the results of `tunelocal` for choosing the optimal model dimension of a variant of three-way correspondence analysis. When `boots = T` the number of different models that is assessed is based on the size of the original data being analysed.
For example, for a 4 x 5 x 4, there are 80 different models that are assessed.
When `boots = T`, the number of different models that is assessed is based on the size of all models obtained from the combination of dimensions of the bootstrapped data.
For example, for a 4 x 5 x 4 array, there are 800 different models that are assessed. By default`nboots = 100`,
you can change the parameter value in input of `tunelocal` function.

Usage

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

Arguments

`x`	The name of the output of the function `tunelocal`.
`digits`	The input parameter specifying the digital number. By default, `digits = 3`.
`...`	Further arguments passed to or from other methods.

Value

The value of output returned depends on the kind of sampling chosen. The sampling for making the convex hull can be based on the original data or on the bootstrapped data samples. In detail:

`XG`	The data samples used for assessing the optimal model dimension (original and/or bootstrapped).
`output1`	The results of `tunelocal`. It gives the goodness-of-fit criteria of models that are located on the boundary of the convex hull and selects the optimal model by means of the scree test values (st); see Ceulemans and Kiers (2006).
`ca3type`	It gives information about the kind of variant of three-way CA considered.
`boots`	The flag parameter to perform the search of optimal dimensions using bootstrap samples. By defaults, `boots = FALSE`.

Author(s)

Rosaria Lombardo, Michel van de Velden and 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

```
res.tunelocal<-tunelocal(happy, ca3type = "CA3")
print(res.tunelocal)
```

---

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 weigth 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

---

ratrank *Rating-ranking data a three-way contingency table*

Description

This three-way contingency table represents a known data set described in van Herk and van de Velden (2007). The three-way contingency table consists of nine rating values against nine ranking values given by the same participants across five European countries (France, Italy, Germany, UK and Spain).

Usage

```
data(ratrank)
```

Format

The format is:

row names [1:9] "1", "2", "3", "4", "5", "6","7","8","9" col names [1:9] "rank1", "rank2", "rank3", "rank4", "rank5", "rank6", "rank7", "rank8", "rank9" tube names [1:5] "F", "I","G","U","S"

References

van Herk H and van de Velden M (2007) Insight into the relative merits of rating and ranking in a cross-national context using three-way correspondence analysis. Food Quality and Preference, 18, 1096–1105.

Examples

```
ratrank<-structure(c(766L, 128L, 38L, 10L, 12L, 3L, 2L, 5L, 9L, 619L,
234L, 67L, 16L, 15L, 5L, 2L, 8L, 7L, 512L, 277L, 109L, 22L, 23L,
5L, 11L, 7L, 7L, 385L, 291L, 152L, 64L, 41L, 9L, 12L, 7L, 12L,
297L, 251L, 192L, 82L, 96L, 17L, 12L, 6L, 20L, 187L, 203L, 259L,
105L, 119L, 44L, 19L, 8L, 29L, 143L, 144L, 209L, 140L, 170L,
54L, 51L, 22L, 40L, 77L, 100L, 152L, 148L, 215L, 73L, 62L, 56L,
90L, 47L, 45L, 84L, 119L, 200L, 82L, 98L, 67L, 231L, 859L, 101L,
53L, 18L, 18L, 9L, 7L, 2L, 16L, 733L, 205L, 53L, 23L, 21L, 13L,
11L, 6L, 18L, 622L, 224L, 124L, 41L, 27L, 8L, 12L, 6L, 19L, 547L,
248L, 102L, 78L, 45L, 19L, 11L, 11L, 22L, 466L, 243L, 139L, 76L,
76L, 25L, 21L, 9L, 28L, 357L, 239L, 168L, 105L, 95L, 61L, 20L,
14L, 24L, 293L, 192L, 165L, 128L, 133L, 42L, 58L, 28L, 44L, 215L,
162L, 161L, 127L, 148L, 60L, 65L, 54L, 91L, 140L, 121L, 142L,
128L, 157L, 69L, 76L, 75L, 175L, 1219L, 193L, 29L, 13L, 3L, 4L,
2L, 6L, 3L, 651L, 504L, 111L, 30L, 19L, 8L, 5L, 8L, 6L, 476L,
```

```
335L, 230L, 35L, 13L, 8L, 5L, 4L, 6L, 346L, 324L, 201L, 136L,
30L, 5L, 5L, 6L, 5L, 239L, 299L, 234L, 101L, 170L, 22L, 14L,
4L, 6L, 166L, 246L, 265L, 116L, 96L, 71L, 27L, 11L, 16L, 124L,
179L, 215L, 163L, 139L, 52L, 80L, 20L, 32L, 80L, 114L, 172L,
148L, 168L, 80L, 84L, 96L, 70L, 63L, 48L, 101L, 115L, 183L, 92L,
123L, 131L, 292L, 916L, 99L, 40L, 12L, 7L, 3L, 3L, 13L, 42L,
578L, 224L, 65L, 11L, 15L, 6L, 3L, 18L, 30L, 486L, 207L, 140L,
34L, 14L, 10L, 7L, 16L, 29L, 405L, 207L, 149L, 64L, 30L, 7L,
14L, 21L, 19L, 304L, 256L, 157L, 60L, 83L, 9L, 17L, 20L, 21L,
239L, 222L, 195L, 95L, 55L, 34L, 20L, 18L, 18L, 204L, 169L, 213L,
113L, 89L, 23L, 45L, 16L, 15L, 165L, 148L, 184L, 128L, 121L,
46L, 38L, 51L, 23L, 89L, 94L, 147L, 141L, 181L, 70L, 57L, 32L,
82L, 1086L, 89L, 37L, 10L, 12L, 6L, 9L, 6L, 24L, 501L, 251L,
55L, 11L, 14L, 7L, 7L, 7L, 11L, 415L, 139L, 188L, 22L, 14L, 8L,
4L, 4L, 12L, 359L, 148L, 111L, 101L, 21L, 7L, 7L, 3L, 15L, 278L,
158L, 128L, 49L, 127L, 9L, 12L, 6L, 13L, 240L, 162L, 130L, 48L,
58L, 49L, 11L, 5L, 12L, 185L, 113L, 148L, 78L, 84L, 26L, 52L,
7L, 16L, 128L, 91L, 119L, 110L, 118L, 37L, 38L, 35L, 28L, 83L,
50L, 67L, 89L, 165L, 47L, 66L, 46L, 120L), .Dim = c(9L, 9L, 5L
), .Dimnames = list(c("1", "2", "3", "4", "5","6", "7",
                      "8", "9"), c("rank1", "rank2", "rank3",
                        "rank4", "rank5", "rank6",
                       "rank7", "rank8", "rank9"),
                      c("F", "I", "G", "U", "S")))
dim(ratrank)
data(ratrank)
```

`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 — *Weighted centred three-way table for three-way non-symmetric correspondence analysis*

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.

`selmod` *Selecting the mode.*

Description

Select the mode in which the column has to be sign reversed. Below is an heuristic algorithm but a fully rational choice is hard to come by. Maximal number of sign reversals = p+q+r-2, but this number can be much smaller. Sign reverse a component, determine which if any of p, q and r is available for reversal. If one of them is wholly positieve way never choose it, else if one is wholly negative choose that one from A, B, C respectively; else if there is a component with a largest absolute value which is negative choose that one, or the one from A,B,C in that order; else choose the column of the longest mode; end if FreeA, FreeB, FreeC = 0 component is not available; = 1 component is available.

Usage

```
selmod(aptr, bptr, cptr, posptrA, negptrA, bigptrA, posptrB, negptrB, bigptrB,
posptrC, negptrC, bigptrC, IA, IB, IC, I, J, K, p, q, r, longest)
```

Arguments

`aptr`	The pointer to the first component matrix.
`bptr`	The pointer to the second component matrix.
`cptr`	The pointer to the third component matrix.
`posptrA`	The pointer to the positive component of the first component matrix.
`negptrA`	The pointer to the negative component of the first component matrix.
`bigptrA`	The pointer to the biggest component (a larger negative positive value than a positive one) of the first component matrix.
`posptrB`	The pointer to the positive component of the first component matrix.
`negptrB`	The pointer to the negative component of the second component matrix.
`bigptrB`	The pointer to the biggest component (a larger negative positive value than a positive one) of the second component matrix.
`posptrC`	The pointer to the positive component of the third component matrix.
`negptrC`	The pointer to the negative component of the third component matrix.
`bigptrC`	The pointer to the biggest component (a larger negative positive value than a positive one) of the third component matrix.
`IA`	The pointer to the columns of the first component matrix, given the pointer to an element in core.
`IB`	The pointer to the columns of the second component matrix, given the pointer to an element in core.
`IC`	The pointer to the columns of the third component matrix, given the pointer to an element in core.

`I`	The row number of the three-way contingency table.
`J`	The column number of the three-way contingency table.
`K`	The tube number of the three-way contingency table.
`p`	The dimension number of the first mode.
`q`	The dimension number of the second mode.
`r`	The dimension number of the third mode.
`longest`	The component matrix of the longest mode to change sign (when no special reason for selection could be found).

Value

`success`	The flag variable to indicate if one of the components has to be sign reversed.
`chgmode`	Select the mode (1, 2 or 3) in which the column has to be sign reversed.
`chgcomp`	The pointer to the columns of the component matrices (A, B or C) that are responsible for the value in the ordered core array.

Author(s)

Rosaria Lombardo and Pieter M Kroonenberg.

References

Kroonenberg PM (1994) The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.

`signscore` *Changing the sign of negative core values.*

Description

This function makes the signs of the largest core elements positive
to facilitate interpretation. The appropriate columns of the component matrices for inversion
are determined and are reversed accordingly.

Usage

```
signscore(a, b, cc, I, J, K, p, q, r, core, IFIXA, IFIXB, IFIXC)
```

Arguments

`a`	The first component matrix A.
`b`	The second component matrix B.
`cc`	The third component matrix C.
`I`	The row number of the first mode of the three-way contingency table.
`J`	The column number of the second mode of the three-way contingency table.
`K`	The tube number of the third mode of the three-way contingency table.
`p`	The dimension number of the first mode.
`q`	The dimension number of the second mode.
`r`	The dimension number of the third mode.
`core`	The core array (generalized singular values).
`IFIXA`	The flag parameter to indicate whether the first component (A) belongs to a fixed mode. Fixed modes should not have their signs changed.
`IFIXB`	The flag parameter to indicate whether the second component (B) belongs to a fixed mode. Fixed modes should not have their signs changed.
`IFIXC`	The flag parameter to indicate whether the third component (C) belongs to a fixed mode. Fixed modes should not have their signs changed.

Value

`g`	The core array.
`gord`	the core array ordered with respect to the largest values (descending order).
`a`	The first matrix of components.
`aord`	The ordered first matrix of the components.
`b`	The second matrix of the components.
`bord`	The ordered second matrix of the components.
`cc`	The third matrix of the components.
`ccord`	The ordered third matrix of the components.

Author(s)

Rosaria Lombardo and Pieter M Kroonenberg.

References

Kroonenberg PM (1994) The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.

simulabootsimple *Generation of parametric bootstrap samples*

Description

This function allows to generate parametric bootstrap samples in order to check the optimal dimension number of three-way correspondence analysis. The boostrap 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. John Wiley & Sons.

simulabootstrat *Generation of parametric bootstrap samples*

Description

This function allows to generate parametric bootstrap samples in order to check the optimal dimension number of three-way correspondence analysis. The boostrap samples have the same marginal proportions and total number of the original table. Te 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. John Wiley & Sons.

`srtcor` *Sort the core array*

Description

This function sorts (a copy of) the core matrix and returns CORPTR, an p*q*r integer array holding the pointers to the greatest absolute values in CORE.

Usage

```
srtcor(core, p, q, r)
```

Arguments

`core`	The core array (generalized singular values).
`p`	The dimension number of the first mode.
`q`	The dimension number of the second mode.
`r`	The dimension number of the third mode.

Value

`coreptr`	The pointer to the ordered largest value of the core array.

Author(s)

Rosaria Lombardo and Pieter M Kroonenberg.

References

Kroonenberg PM (1994) The TUCKALS line: a suite of programs for three-way data analysis. Computational Statistics and Data Analysis, 18, 73–96.

`standtab` *Three-way centred column profile table for the three-way symmetric correspondence analysis*

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 — *The core array derived via the Tucker3 model.*

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 — *The core array derived via the Trivariate Moment Decomposition model.*

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 *Component matrices from the Trivariate Moment Decomposition decomposition*

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.

`summary.CA3variants` *Summary of three-way correspondence analysis results*

Description

This function prints the summary of the results of three-way symmetrical or non-symmetrical correspondence analysis. In particular it gives information on core and squared core and on the explained inertia when reducing dimensions.

Usage

```
## S3 method for class 'CA3variants'
summary(object, digits=3, ...)
```

Arguments

`object`	The name of the output of the main function `CA3variants`.
`digits`	The input parameter specifying the digital number. By default, `digits = 3`.
`...`	Further arguments passed to or from other methods.

Value

The value of output returned in short depends on the kind of three-way correspondence analysis variant performed. It gives the core table, the squared core table, the explained inertia, the total inertia and its proportion.

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

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

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

Value

`z`	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.
`CM`	the C statistic of the Marcotorchino index.
`devt`	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` *The partition of the Marcotorchino three-way index.*

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

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

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 componets 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 — *Generation of non-parametric bootstrap samples*

Description

This function allows to generate non-parametric bootstrap samples in order to check the optimal dimension number of three-way correspondence analysis. The boostrap 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 — *Tucker3 decomposition of the three-way table.*

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 *Trivariate moment decomposition of the three-way table.*

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 decmposition 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 (2016) 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, 81(2), 325–349.

tunelocal — *Dimension selection for three-dimensional correspondence biplot using convex hull.*

Description

This function allows to select the optimal dimension number for correspondence biplot, given the set of possible dimension combination of the original data. It determines the models that are located on the boundary of the convex hull and selects an optimal model by means of the scree test values (st). For exploring, it is also possible to check the optimal model dimension by using boostrap 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, PercentageFit = 0.01)
```

Arguments

`Xdata` 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 `resp = "col"` or `resp = "tube"`.

`ca3type` The specification of the analysis to be performed. If `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. If `ca3type = "OCA3"`, then ordered three-way symmetrical correspondence analysis will be performed. If `ca3type = "ONSCA3"`, then ordered three-way non-symmetrical correspondence analysis will be performed.

`resp` The input parameter for specifying in non-symmetrical three-way correspondence analysis variants (`ca3type = "NSCA3"` and `ca3type = "ONSCA3"`) what is the response variable (logically antecendent to the others). By default `resp = "row"`, but it could be the column variable `resp = "col"` or the tube variable `resp = "tube"`.

`norder` The input parameter for specifying the number of ordered variable when `ca3type = "OCA3"` or `ca3type = "ONSCA3"`.

`digits` The input parameter specifying the digital number. By default, `digits = 3`.

`boots` The flag parameter to perform the search of optimal dimensions using bootstrap samples. By defaults, `boots = FALSE`.

`nboots` The number of bootstrap samples to generate when `boots = TRUE`. Note that when `boots = FALSE`,
by default `nboots = 0`, but when `boots = TRUE`, by default `nboots = 100`.

`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.

`PercentageFit` Required proportion of increase in fit of a more complex model. By default, `PercentageFit = 0.01`.

Value

`output1` Chi-square criterion and df of models on the convex hull. It gives the criterion values of the models that are located on the boundary of the convex hull and selects the optimal model by means of the scree test values (st). When using `boots = FALSE`, it gives the set of possible dimension combination of the original data using only the original data array. When using `boots = TRUE`, it gives the set of possible dimension combination of the original data using bootstrapped data 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

```
tunelocal(Xdata = happy, ca3type = "CA3")
```

Index