

Package ‘CA3variants’

September 17, 2018

Type Package

Title Three-Way Correspondence Analysis Variants

Version 1.0

Date 2018-08-20

Author Rosaria Lombardo and Eric J Beh

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

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

Depends R (> 3.0.1), methods, tools, ggplot2, gridExtra, ggrepel

LazyData true

License GPL (> 2)

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

NeedsCompilation no

Repository CRAN

Date/Publication 2018-09-17 09:00:13 UTC

R topics documented:

ca3basic	2
CA3variants	4
chi3	5
chi3ordered	6
criter	7
criteria	8
emerson.poly	8
flatten	9
graph2poly	10
happy	11
init3	12
init3ordered	13
init3ordered1	14

init3ordered2	15
Kron	16
loss1.3	16
loss1.3ordered	17
loss2	18
margI	19
margJ	19
margK	20
newcomp3	21
newcomp3ordered	21
newcomp3ordered1	22
newcomp3ordered2	23
nsca3basic	23
oca3basic	25
olive	26
p.ext	27
plot.CA3variants	27
print.CA3variants	29
prod3	31
reconst3	32
rstand3	32
standtab	33
step.g3	34
step.g3ordered	35
stepi3	35
stepi3ordered	36
tau3	37
tucker	38
tuckerORDERED	39
Index	41

ca3basic

Three-way Symmetrical Correspondence Analysis

Description

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

Usage

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

Arguments

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

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.

Description

This function performs the three-way symmetrical (when `ca3type = "CA3"`) and non-symmetrical correspondence analysis (when `ca3type = "NSCA3"`) by using the Tucker3 decomposition and an ordered variant of three-way symmetrical correspondence analysis (when `ca3type = "OCA3"`) by using the Trivariate Moment Decomposition. The non-symmetrical variant considers 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.

Usage

```
CA3variants(Xtable, p = dim(Xtable)[[1]], q = dim(Xtable)[[2]], r = dim(Xtable)[[3]],
ca3type = "CA3", test = 10^-6, norder = 3)
```

Arguments

<code>Xtable</code>	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.
<code>p</code>	The number of components for the first mode. By default, <code>p = dim(Xtable)[[1]]</code> .
<code>q</code>	The number of components for the second mode. By default, <code>q = dim(Xtable)[[2]]</code> .
<code>r</code>	The number of components for the third mode. By default, <code>r = dim(Xtable)[[3]]</code> .
<code>ca3type</code>	The specification of the analysis to be performed. If <code>ca3type = "CA3"</code> , then a three-way (symmetrical) correspondence analysis will be performed (default analysis). If <code>ca3type = "NSCA3"</code> , then three-way non-symmetrical correspondence analysis will be performed. If <code>ca3type = "OCA3"</code> , then ordered three-way correspondence analysis will be performed.
<code>test</code>	Threshold used in the algorithm for stopping it after the convergence of the solutions.
<code>norder</code>	The input parameter for specifying the number of ordered variable when <code>ca3type = "OCA3"</code> .

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"`

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

Examples

```
data(happy)
CA3variants(happy,p=2,q=2,r=2, ca3type = "CA3")
```

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

Examples

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

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

Description

When three categorical variables are symmetrically related, we can analyse the strength of the association using the three-way Pearson mean square contingency coefficient, named the chi-squared index. The function chi3 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 components of inertia, the percentage of explained inertia, the degrees of freedom and p-value of each term of the partition.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

Examples

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

criter	<i>The stopping criteria for the Tucker3 algorithm</i>
--------	--

Description

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

Usage

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

Arguments

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

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh.

References

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

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

Description

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

Usage

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

Arguments

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

Author(s)

Rosaria Lombardo, Eric J Beh

References

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

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

Description

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

Usage

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

Arguments

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

Value

Describe the value returned

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

Note

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

Author(s)

Rosaria Lombardo and Eric J Beh.

References

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

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

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

Examples

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

flatten	<i>Flattened table</i>
---------	------------------------

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.

graph2poly

Two-dimensional correspondence plot or biplot.

Description

This function produces a two-dimensional correspondence biplot, given the set of computed coordinates. In case of ordered correspondence analysis, the biplot uses polynomial coordinates.

Usage

```
graph2poly(f, g, ni, nj, nk, a1 = 1, a2 = 2, prop = 0.8, cex = 1,
length = 0.01, cex.lab = 0.8, cols = c(1, 4), mar = c(5, 4, 4, 2) + 0.1,
arrow = F, type = "l", pos = 2)
```

Arguments

f	The polynomial coordinates of the row categories.
g	The polynomial coordinates of the column-tube categories.
ni	The category number of row variable.
nj	The category number of column variable.
nk	The category number of tube variable.
a1	The dimension reflected along the horizontal principal axis in the graphical display. By default n1=1.
a2	The dimension reflected along the vertical principal axis in the graphical display. By default n1=2.
prop	The scaling parameter for changing the area of plotting region. By default, prop = 1.
cex	The parameter that specifies the size of character labels of points in graphical displays. By default, it is equal to 0.5.
length	The length of the arrows whenever graphically depicted.
cex.lab	The parameter for setting the size of character labels of axes in graphical displays.
cols	The parameter for setting the column size of data table.
mar	The parameter for setting the size of plot area in graphical displays.

arrow	The flag parameter to depict arrows (T or F), if T the distances between the column-tube categories are represented as arrows.
type	The input parameter for specifying the type of line to draw between coordinates. By default, it is equal to 1 a line.
pos	The input parameter for specifying the label position. By default, it is equal to 2.

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	<i>Three-way contingency table</i>
-------	------------------------------------

Description

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

Usage

```
data(happy)
```

Format

The format is:

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

References

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

Examples

```

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

```

init3

Initial components from the Tucker3 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	<i>Initial components from the Trivariate Moment Decomposition algorithm</i>
--------------	--

Description

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

Usage

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

Arguments

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

Value

The initial components for each of the three categorical variables.

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

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

Description

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

Usage

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

Arguments

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

Value

The initial components for each of the three categorical variables.

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

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

Description

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

Usage

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

Arguments

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.

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

Description

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

Usage

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

Arguments

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

Details

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

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

Description

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

Usage

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


Arguments

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

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

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

Description

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

Usage

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

Arguments

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

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

loss2

Difference between two successive components

Description

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

Usage

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

Arguments

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

Value

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

Author(s)

Rosaria Lombardo and Eric J Beh.

References

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

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

Description

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

Usage

```
margI(m)
```

Arguments

m The three-way contingency table.

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

margJ	<i>Column marginals of a three-way contingency table</i>
-------	--

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.

newcomp3	<i>Updated component matrices</i>
----------	-----------------------------------

Description

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

Usage

```
newcomp3(param)
```

Arguments

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

Details

It is utilised from the function `tucker`.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

newcomp3ordered	<i>Updated component matrices</i>
-----------------	-----------------------------------

Description

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

Usage

```
newcomp3ordered(param)
```

Arguments

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

Details

It is utilised from the function tuckerORDERED.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

newcomp3ordered1 *Updated component matrices*

Description

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

Usage

```
newcomp3ordered1(param)
```

Arguments

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

Details

It is utilised from the function tuckerORDERED.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

newcomp3ordered2 *Updated component matrices*

Description

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

Usage

```
newcomp3ordered2(param)
```

Arguments

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

Details

It is utilised from the function `tuckerORDERED`.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

nsca3basic *Three-way Non-Symmetrical Correspondence Analysis*

Description

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

Usage

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

Arguments

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.

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, Ida Camminatiello.

References

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

Description

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

Usage

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

Arguments

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

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh.

References

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

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

p.ext *The external product in Tuckals3.*

Description

The computation of external product between the principal components.

Usage

```
p.ext(x,y)
```

Arguments

x	x matrix IxS
y	y matrix JxS

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

plot.CA3variants *Graphical display resulting from CAvariants3*

Description

The function plot.CA3variants allows the analyst to graphically display the biplot from their analysis. When the input parameter is biptype = "column-tube", the function displays the column-tube interactive biplot. When the input parameter is biptype = "row", the function displays the row interactive biplot. By default, biptype = "column-tube".

Usage

```
## S3 method for class 'CA3variants'
plot(x, firstaxis = 1, lastaxis = 2, cex = 0.8,
      plottype = "biplot", biptype="column-tube", prop = 1, scaleplot = 1, arrow = T, pos=1,
      size=2, size2=3,...)
```

Arguments

<code>x</code>	The output parameters of the main function <code>CA3variants</code> .
<code>firstaxis</code>	The dimension reflected along the horizontal axis.
<code>lastaxis</code>	The dimension reflected along the vertical axis.
<code>cex</code>	The parameter that specifies the size of character labels of points in graphical displays. By default, it is equal to 0.8.
<code>plottype</code>	The input parameter for specifying what kind of graphical display is requested. By default, it is equal to <code>biplot</code> .
<code>biptype</code>	The input parameter for specifying what kind of biplot is requested. By default, it is equal to <code>column-tube</code> , but could be <code>row</code> .
<code>prop</code>	The scaling parameter for changing the limits of the plotting area. By default, it is equal to 1.
<code>scaleplot</code>	The scaling parameter for <code>pos=1,size=2,cols=c(1,4)</code> . By default, it is equal to 1.
<code>arrow</code>	The flag parameter to display arrows (T or F), if T the distances between the column-tube categories are represented as arrows.
<code>pos</code>	The input parameter for changing the label position. By default, it is equal to 1.
<code>size</code>	The input parameter for specifying the size of points. By default, it is equal to 2.
<code>size2</code>	The input parameter for specifying the label size. By default, it is equal to 3.
<code>...</code>	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, Ida Camminatiello.

References

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

Examples

```
data(happy)
ris.ca3<-CA3variants(happy,p=2,q=2,r=2, ca3type = "CA3")
plot(ris.ca3)
ris.n sca3<-CA3variants(happy,p=2,q=2,r=2, ca3type = "NSCA3")
plot(ris.n sca3)
```

```
ris.oca3<-CA3variants(happy,p=2,q=2,r=2, ca3type = "OCA3",norder=3)
plot(ris.oca3)
```

```
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. 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, digits = 3,...)
```

Arguments

x	the name of the output of the main function CAvariants.
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.
DataMatrix	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 reconstructed by the three component matrices (of dimension Ixp, Jxq, and Kxr) and the core array.
nxhat2	The sum of squares of the xhat table elements.

<code>prp</code>	The ratio between the sum of squares of the <code>xhat</code> table elements and of the <code>xs</code> table elements.
<code>fi</code>	The principal row coordinates.
<code>fiC</code>	The standard row coordinates.
<code>gjk</code>	The principal column-tube coordinates.
<code>rows</code>	The row marginals of the three-way data table.
<code>cols</code>	The column marginals of the three-way data table.
<code>tubes</code>	The tube marginals of the three-way data table.
<code>flabels</code>	The row category labels.
<code>glabels</code>	The column category labels.
<code>maxaxes</code>	The maximum dimension to consider.
<code>inertia</code>	The total inertia of three-way symmetrical or non-symmetrical correspondence analysis.
<code>inertiapc0</code>	The percentage contribution of the three components to the total variation.
<code>inertiapc</code>	The vector of the percentage contributions of the three components to the total inertia.
<code>iproduct</code>	The inner product between the standard row coordinates (<code>fi</code>) and the column-tube principal coordinates (<code>gjk</code>).
<code>g</code>	The core array (i.e. the generalized singular values) calculated by using the Tuckals3 algorithm.
<code>index3</code>	When <code>ca3type = "CA3"</code> , the <code>index3</code> 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 <code>ca3type = "NSCA3"</code> <code>index3</code> 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.
<code>ca3type</code>	The specification of the analysis to be performed. When <code>ca3type = "CA3"</code> , then a three-way symmetrical correspondence analysis will be performed (default analysis). If <code>ca3type = "NSCA3"</code> , then three-way non-symmetrical correspondence analysis will be performed, 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 <code>ca3type = "OCA3"</code> , then an ordered three-way symmetrical correspondence analysis will be performed. Differently, this analysis uses a new method of decomposition called Trivariate Moment Decomposition.
<code>iteration</code>	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

Examples

```
data(happy)
ris.ca3<-CA3variants(happy,p=2,q=2,r=2, ca3type = "CA3")
print(ris.ca3)
ris.n sca3<-CA3variants(happy,p=2,q=2,r=2, ca3type = "NSCA3")
print(ris.n sca3)
ris.o ca3<-CA3variants(happy,p=2,q=2,r=2, ca3type = "OCA3",norder=3)
print(ris.o ca3)
```

prod3

Products among arrays

Description

The function calculates the products among arrays.

Usage

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

Arguments

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

Details

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

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

 reconst3

Reconstruction of the three-way centred profile table

Description

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

Usage

```
reconst3(param)
```

Arguments

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

Value

The three-way reconstructed table of centred profiles.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

 rstand3

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.

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

Description

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

Usage

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

Arguments

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

step.g3

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 general element can be interpreted as a generalized singular value.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

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

Description

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

Usage

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

Arguments

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

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

stepi3	<i>Component matrices from the Tucker3 decomposition</i>
--------	--

Description

The function computes the component matrices from the Tuckals3 algorithm.

Usage

```
stepi3(param)
```

Arguments

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

Details

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

Value

Component matrices from the Tucker3 decomposition.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

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

Description

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

Usage

```
stepi3ordered(param)
```

Arguments

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

Details

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

Value

Component matrices from the Trivariate Moment Decomposition decomposition.

Author(s)

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References

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

tau3	<i>Partition of the Marcotorchino three-way index</i>
------	---

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)
```

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

Description

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

Usage

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

Arguments

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

Details

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

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh.

References

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

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

Description

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

Usage

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

Arguments

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

Details

The function tuckerORDERED is utilised from the function oca3basic.

Value

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

Author(s)

Rosaria Lombardo, Eric J Beh.

References

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

Index

- *Topic **Three-way correspondence analysis**
 - ca3basic, 2
- *Topic **Trivariate Moment Decomposition**
 - init3ordered, 13
 - init3ordered1, 14
- *Topic **Tuckals3**
 - stepi3, 35
- *Topic **Tucker3 components**
 - newcomp3, 21
 - newcomp3ordered, 21
 - newcomp3ordered1, 22
 - newcomp3ordered2, 23
- *Topic **Tucker3**
 - init3, 12
 - init3ordered, 13
 - init3ordered1, 14
 - init3ordered2, 15
- *Topic **Tucklas3**
 - init3, 12
 - init3ordered2, 15
 - stepi3ordered, 36
 - tucker, 38
 - tuckerORDERED, 39
- *Topic **array**
 - margI, 19
 - margJ, 19
 - margK, 20
- *Topic **biplot**
 - graph2poly, 10
- *Topic **component**
 - criteria, 8
- *Topic **core**
 - step.g3, 34
 - step.g3ordered, 35
- *Topic **criterion**
 - criter, 7
 - loss1.3, 16
 - loss1.3ordered, 17
- *Topic **datasets**
 - happy, 11
 - olive, 26
- *Topic **graphical display**
 - plot.CA3variants, 27
- *Topic **index**
 - tau3, 37
- *Topic **interactive biplot**
 - plot.CA3variants, 27
- *Topic **interactive plot**
 - plot.CA3variants, 27
- *Topic **loss**
 - loss2, 18
- *Topic **marginal**
 - margI, 19
 - margJ, 19
 - margK, 20
- *Topic **multivariate**
 - flatten, 9
 - Kron, 16
 - loss1.3, 16
 - loss1.3ordered, 17
 - newcomp3, 21
 - newcomp3ordered, 21
 - newcomp3ordered1, 22
 - newcomp3ordered2, 23
 - nsca3basic, 23
 - print.CA3variants, 29
 - reconst3, 32
 - rstand3, 32
 - standtab, 33
 - step.g3, 34
 - step.g3ordered, 35
 - stepi3, 35
 - stepi3ordered, 36
 - tucker, 38
 - tuckerORDERED, 39
- *Topic **nonparametric**

- emerson.poly, 8
 - *Topic **ordered**
 - oca3basic, 25
 - *Topic **partition**
 - print.CA3variants, 29
 - tau3, 37
 - *Topic **plot**
 - graph2poly, 10
 - *Topic **product**
 - Kron, 16
 - p.ext, 27
 - prod3, 31
 - *Topic **reconstruction**
 - reconst3, 32
 - *Topic **three way index**
 - chi3ordered, 6
 - *Topic **three-way index**
 - chi3, 5
 - *Topic **three-way non-symmetric
correspondece analysis**
 - nsca3basic, 23
 - *Topic **three-way non-symmetrical
correspondence analysis.**
 - CA3variants, 4
 - *Topic **three-way symmetrical
correspondence analysis.**
 - CA3variants, 4
 - *Topic **tuckals3**
 - p.ext, 27
 - *Topic **tucker**
 - criteria, 8
 - *Topic **weighted array**
 - prod3, 31
 - *Topic **weight**
 - rstand3, 32
-
- ca3basic, 2
 - CA3variants, 4
 - chi3, 5
 - chi3ordered, 6
 - criter, 7
 - criteria, 8
 - emerson.poly, 8
 - flatten, 9
 - graph2poly, 10
 - happy, 11
 - init3, 12
 - init3ordered, 13
 - init3ordered1, 14
 - init3ordered2, 15
 - Kron, 16
 - loss1.3, 16
 - loss1.3ordered, 17
 - loss2, 18
 - margI, 19
 - margJ, 19
 - margK, 20
 - newcomp3, 21
 - newcomp3ordered, 21
 - newcomp3ordered1, 22
 - newcomp3ordered2, 23
 - nsca3basic, 23
 - oca3basic, 25
 - olive, 26
 - p.ext, 27
 - plot.CA3variants, 27
 - print.CA3variants, 29
 - prod3, 31
 - reconst3, 32
 - rstand3, 32
 - standtab, 33
 - step.g3, 34
 - step.g3ordered, 35
 - stepi3, 35
 - stepi3ordered, 36
 - tau3, 37
 - tucker, 38
 - tuckerORDERED, 39