Package ‘CA3variants’

April 28, 2020

Type Package
Title Three-Way Correspondence Analysis Variants
Version 2.5
Date 2020-04-27
Author Rosaria Lombardo, Eric J Beh and Michel van de Velden
Maintainer Rosaria Lombardo <rosaria.lombardo@unicampania.it>
Description Provides four variants of three-way correspondence analysis (ca):
three-way symmetrical ca, three-way non-symmetrical ca, three-way ordered symmetrical ca
and three-way ordered non-symmetrical ca.
Depends R (> 3.0.1), methods, tools, ggforce, gridExtra, ggrepel,
multichull
Imports ggplot2, plotly
LazyData true
License GPL (> 2)
URL https://www.R-project.org
NeedsCompilation no
Repository CRAN
Date/Publication 2020-04-28 19:00:02 UTC

R topics documented:

ca3basic ......................................................... 3
caplot .......................................................... 4
CA3variants ..................................................... 5
caplot3d ........................................................ 6
chi3 ............................................................ 7
chi3ordered ................................................... 8
criter ........................................................... 9
criteria ........................................................ 10
emerson.poly ............................................... 10
flatten ......................................................... 11
<table>
<thead>
<tr>
<th>R topics documented:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>happy</td>
<td>12</td>
</tr>
<tr>
<td>init3</td>
<td>13</td>
</tr>
<tr>
<td>init3ordered</td>
<td>14</td>
</tr>
<tr>
<td>init3ordered1</td>
<td>15</td>
</tr>
<tr>
<td>init3ordered2</td>
<td>16</td>
</tr>
<tr>
<td>Kron</td>
<td>17</td>
</tr>
<tr>
<td>loss1.3</td>
<td>17</td>
</tr>
<tr>
<td>loss1.3ordered</td>
<td>18</td>
</tr>
<tr>
<td>loss2</td>
<td>19</td>
</tr>
<tr>
<td>makeindicator</td>
<td>20</td>
</tr>
<tr>
<td>margI</td>
<td>20</td>
</tr>
<tr>
<td>margJ</td>
<td>21</td>
</tr>
<tr>
<td>margK</td>
<td>22</td>
</tr>
<tr>
<td>newcomp3</td>
<td>22</td>
</tr>
<tr>
<td>newcomp3ordered</td>
<td>23</td>
</tr>
<tr>
<td>newcomp3ordered1</td>
<td>24</td>
</tr>
<tr>
<td>newcomp3ordered2</td>
<td>24</td>
</tr>
<tr>
<td>nsca3basic</td>
<td>25</td>
</tr>
<tr>
<td>oca3basic</td>
<td>26</td>
</tr>
<tr>
<td>olive</td>
<td>27</td>
</tr>
<tr>
<td>onsca3basic</td>
<td>28</td>
</tr>
<tr>
<td>p.ext</td>
<td>29</td>
</tr>
<tr>
<td>plot.CA3variants</td>
<td>30</td>
</tr>
<tr>
<td>print.CA3variants</td>
<td>31</td>
</tr>
<tr>
<td>prod3</td>
<td>34</td>
</tr>
<tr>
<td>reconst3</td>
<td>35</td>
</tr>
<tr>
<td>rstand3</td>
<td>35</td>
</tr>
<tr>
<td>simulabootsimple</td>
<td>36</td>
</tr>
<tr>
<td>simulabootstrat</td>
<td>37</td>
</tr>
<tr>
<td>standtab</td>
<td>38</td>
</tr>
<tr>
<td>step.g3</td>
<td>38</td>
</tr>
<tr>
<td>step.g3ordered</td>
<td>39</td>
</tr>
<tr>
<td>stePi3</td>
<td>40</td>
</tr>
<tr>
<td>stePi3ordered</td>
<td>40</td>
</tr>
<tr>
<td>tau3</td>
<td>41</td>
</tr>
<tr>
<td>tau3ordered</td>
<td>42</td>
</tr>
<tr>
<td>threeewayboot</td>
<td>43</td>
</tr>
<tr>
<td>tucker</td>
<td>44</td>
</tr>
<tr>
<td>tuckerORDERED</td>
<td>45</td>
</tr>
<tr>
<td>tunelocal</td>
<td>46</td>
</tr>
</tbody>
</table>

Index 49
**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**

```r
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

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
```r
ca3plot(frows, gcols, firstaxis, lastaxis, nseg, inertiapc, thingseg, col1, col2, col3, size1, size2)
```

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.
- `nseg`: The vectors/arrows number where to project principal (or standard) coordinates.
- `inertiapc`: The percentage of the explained inertia by each dimension.
- `thingseg`: The principal or standard coordinates used to draw vectors (arrows).
- `col1`: The colour for the row variable labels.
- `col2`: The colour for the column variable labels.
- `col3`: The colour for the vectors (arrows) used in biplots.
- `size1`: The size of the plotted symbol for categories in biplot.
- `size2`: The size of the plotted text for categories in biplot.

Note
This function depends on the R library `plotly`.

Author(s)
Rosaria Lombardo and Eric J. Beh
CA3variants

References


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

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.
p The number of components for the first mode. By default, p = dim(Xtable)[[1]].
q The number of components for the second mode. By default, q = dim(Xtable)[[2]].
r The number of components for the third mode. By default, r = dim(Xtable)[[3]].
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.
norder The input parameter for specifying the number of ordered variable when ca3type = "OCA3".
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, Ida Camminatiello.

References


Examples

data(happy)
CA3variants(happy, p=2, q=2, r=2, ca3type = "CA3")
CA3variants(happy, p=2, q=2, r=2, ca3type = "OCA3")
CA3variants(happy, p=2, q=2, r=2, ca3type = "NSCA3")
CA3variants(happy, p=2, q=2, r=2, 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 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


---

**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

Examples
```r
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
```r
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
Examples

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

Description

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

Usage

```r
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

**critera**

*Criterion of the Tucker3 algorithm*

**Description**

The function critera 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**


---

**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**

derson.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.
**flatten**

**Value**
Describle 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**

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

---

**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 IJK 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


Description

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

Usage

data(happy)

Format

The format is:


References


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 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

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**

```r
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.

```r
a
b
c
```

- `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**

 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**

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**

**Kron**  
**Kronecker product**

**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**

---

**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 \( \text{xhat} \), where \( x \) is the three-way contingency table and \( \text{xhat} \) is the reconstruction of this table by means of components and core array.

**Usage**
loss1.3(param, comp.old)
loss1.3ordered

Arguments

  \texttt{param} \quad \text{The matrices of the row, column and tube components derived via the Tucker3 model.}

  \texttt{comp.old} \quad \text{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


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

\begin{verbatim}
loss1.3ordered(param, comp.old)
\end{verbatim}

Arguments

  \texttt{param} \quad \text{The matrices of the row, column and tube components derived via the Trivariate Moment Decomposition model.}

  \texttt{comp.old} \quad \text{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.
loss2

References


| 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

**Make an Indicator matrix**

**Description**

From a three-way contingency table (as can be used in CA3 variants), it gives the N x total number of categories (rows+cols+tubs) indicator matrix.

**Usage**

```r
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**


**Examples**

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

**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**

```r
margI(m)
```
The function computes the column marginals of the three-way contingency table specified by the input parameter.

Usage

\[ \text{margJ}(m) \]

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

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**


---

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 `tuckerORDERED`.

**Author(s)**

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

**References**


---

**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**

newcomp3ordered1

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

newcomp3ordered2

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


---

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 threshold 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).
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 con-
verge.

Author(s)
Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

References
John Wiley & Sons.

oca3basic Three-way Ordered Symmetrical Correspondence Analysis

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

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

Arguments

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.
### Value

x  The original three-way contingency table.
xs The weighted three-way contingency table.
xhat Three-way contingency table reconstructed after Tuckals 3 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.
c The tube principal coordinates.
g The core array calculated by using the Tuckals 3 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


---

### 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:

References


Examples

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

onsca3basic

Three-way Ordered Non-Symmetrical Correspondence Analysis

Description

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

Usage

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

Arguments

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

Value

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

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.

The row principal components.

The column principal coordinates.

The tube principal coordinates.

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.

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

Author(s)
Rosaria Lombardo, Eric J Beh.

References

---

**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 (I,J),S with elements xis per yis

**Author(s)**

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.
References


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,
biptype="column-tube", scaleplot = 1, plot3d = FALSE, pos=1,
size1=1, size2=2, invproj = FALSE, col1 = "blue", col2 = "red",...)

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.
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.
scaleplot The scaling parameter for pos=1, size=2, cols=c(1,4). By default, it is equal to 1.
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.
invproj The logical parameter specifies whether to portray standard coordinates as vectors and principal coordinates as points or vice-versa. By default, invproj = TRUE.
col1 The input parameter for specifying the row label colour. By default, it is blue.
col2 The input parameter for specifying the column label colour. By default, it is red.
... 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


Examples

data(happy)
ris.ca3<-CA3variants(happy, p = 2, q = 2, r = 2, ca3type = "CA3")
plot(ris.ca3, invproj=TRUE, col1="red", col2="green")
ris.nsca3<-CA3variants(happy, ca3type = "NSCA3")
plot(ris.nsca3, plot3d = TRUE)
ris.oca3<-CA3variants(happy, p = 2, q = 2, r = 2, ca3type = "OCA3", norder = 3)
plot(ris.oca3)
ris.onsca3<-CA3variants(happy, p = 2, q = 2, r = 2, ca3type = "ONSCA3", norder = 3)
plot(ris.onsca3)

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, digits = 3,...)
Arguments

x
digits
...

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
DataMatrix
xs
xhat
nxhat2
prp
fi
fiC
gjk
rows
cols
tubes
flabels
glabels
maxaxes
inertia
inertiaRSS
inertiapc
inertiacoltub
inertiarow

The output of the kind of three-way correspondence analysis analysis considered.
The original three-way contingency table.
The centred and weighted three-way contingency table when the input parameters are ct=1 and std=T.
The three-way contingency table approximated (reconstructed) by the three component matrices (of dimension Ixp, Jxq, and Kxr) and the core array.
The sum of squares of the approximated contingency table.
The ratio between the inertia of the complete contingency table and the inertia of the approximated contingency table.
The principal row coordinates.
The standard row coordinates.
The principal column-tube coordinates.
The row marginals of the three-way data table.
The column marginals of the three-way data table.
The tube marginals of the three-way data table.
The row category labels.
The column category labels.
The maximum dimension to consider.
The total inertia of a variant of three-way correspondence analysis.
The residual inertia of a variant of three-way correspondence analysis.
The percentage contribution of the three components to the total variation.
The vector of the percentage contributions of the interactively coded column-tube components to the total inertia, useful for making interactively coded biplots.
The vector of the percentage contributions of the row components to the total inertia, useful for making response biplots.
The inner product between the standard row coordinates ($f$) and the column-tube principal coordinates ($g_{ij}$).

The core array (i.e. the generalized singular values) calculated by using the Tuckals3 algorithm.

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.

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.

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

Author(s)
Rosaria Lombardo, Eric J Beh.

References

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

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

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)
simulabootsimple

**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 `CAvariants="NSCA"` and represents the three-way weighted and centred column profile table.

**Author(s)**

Rosaria Lombardo, Eric J Beh, Ida Camminatiello.

**References**


---

**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 bootstrap samples have the same marginal proportions and the total number of the original table. The adopted sampling scheme is simple.

**Usage**

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

**Arguments**

- **Xtable**  
  The three-way data array. It must be an `R` object array. When non-symmetrical analysis for one response variable is performed, the response mode is the row variable.

- **nboots**  
  The number of bootstrap samples to generate when `boots = T`. By default `nboots = 0`.

- **resamptype**  
  Set value of `resamptype` according to two methods: `resamptype=1` corresponds to multinomial distribution and `resamptype=2` to Poisson distribution.
**simulabootstrat**

**Author(s)**

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

**References**


---

**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 bootstrap samples have the same marginal proportions and total number of the original table. The adopted sampling scheme is stratified.

**Usage**

```r
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**

**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**

```r
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**


---

**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**

```r
step.g3(param)
```
step.g3ordered

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


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

stei3

Component matrices from the Tucker3 decomposition

Description
The function computes the component matrices from the Tuckals3 algorithm.

Usage
stei3(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, stei3, 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

stei3ordered

Component matrices from the Trivariate Moment Decomposition de-

composition

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

Usage
stei3ordered(param)
tau3

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


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


Examples

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

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

```r
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 components of inertia, the percentage of explained inertia, the degrees of freedom and p-value of each term of the partition.
**threewayboot**

**Author(s)**

Rosaria Lombardo, Eric J Beh.

**References**


**Examples**

```r
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 bootstrap samples have the same marginal proportions and the total number of the original table. Do nboots bootstrap on the indicator matrix X (observations x (rows+cols+tubs) categories). From a three-way contingency table, it makes the indicator using `makeindicator`. The output is a list of three-way tables.

**Usage**

```r
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**

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 threshold 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

**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**

```r
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 threshold 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


tunelocal

Dimension selection for three-dimensional correspondence biplot.

Description

This function allows to select the optimal dimension number for correspondence biplot, given the set of possible dimension combination of the original data. For exploring, it is possible to check the optimal dimension of bootstrap samples which have the same marginal proportions and the total number of the original table. When the input parameter boots = T, it does bootstrap sampling. There are three kinds of possible bootstrap sampling. When boottype = "bootnp" it performs a non parametric bootstrap sampling. When boottype = "bootsimple" it performs a parametric (for resamptype=1 is multinomial or for resamptype=2 is poisson) simple bootstrap sampling. When boottype = "bootpstrat", it performs a parametric stratified bootstrap sampling.

Usage

tunelocal(Xdata, ca3type = "CA3", norder = 3, digits = 3, boots = FALSE, nboots = 0, boottype = "bootsimple", resamptype = 1)

Arguments

Xdata 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.
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.
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.
tunelocal

nboots
The number of bootstrap samples to generate when boots = TRUE. By default
nboots = 0.

boottype
The specification of the kind of bootstrap sampling to be performed. If boottype
= "bootsimple", then a parametric bootstrap using a simple sampling scheme
will be performed (default sampling). If boottype = "bootsstrat", then a
parametric bootstrap using a stratified sampling scheme will be performed. If
boottype = "bootsnp", 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 distribu-
tion.

Value

XG
The list of tables on which is performed the three-way CA variant. It consists of
the original array and (when boots=T) bootstrapped arrays.

output1
Chi-square criterion and df of models on the convex hull when using the original
array.

output2
Chi-square criterion and df of models on the convex hull when using boot-
strapped arrays.

output3
Badness of fit criterion and df of models on the convex hull when using the
original array.

output4
Badness of fit criterion and df of models on the convex hull when using boot-
strapped arrays.

output5
Goodness of fit criterion and df of models on the convex hull when using the
original array.

output6
Goodness of fit criterion and df of models on the convex hull when using boot-
strapped arrays.

Author(s)
Rosaria Lombardo, Michel van de Velden, Eric J Beh.

References
John Wiley & Sons.
Ceulemans E, and Kiers H A L (2006). Selecting among three-mode principal component models of different types and
complexities: A numerical convex hull based method. British Journal of Mathematical & Statistical
Psychology, 59, 133-150.

Examples

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

*Topic **Three-way correspondence analysis**
c3austic, 3
*Topic **Trivariate Moment Decomposition**
init3ordered, 14
init3ordered1, 15
*Topic **Tuckals3**
step3.40
*Topic **Tucker3 components**
newcomp3, 22
newcomp3ordered, 23
newcomp3ordered1, 24
newcomp3ordered2, 24
*Topic **Tucker3**
init3, 13
init3ordered, 14
init3ordered1, 15
init3ordered2, 16
*Topic **Tucklas3**
init3, 13
init3ordered2, 16
step3ordered, 40
tucker, 44
tuckerORDERED, 45
*Topic **array**
margI, 20
margJ, 21
margK, 22
*Topic **biplot**
c3plot, 4
*Topic **bootstrap**
simulaboottsimple, 36
simulaboottstrat, 37
threeawayboot, 43
tunelocal, 46
*Topic **component**
criteria, 10
*Topic **core**
step3, 38
step3ordered, 39
*Topic **criterion**
criter, 9
loss1.3, 17
loss1.3ordered, 18
*Topic **datasets**
happy, 12
olive, 27
*Topic **dimension**
simulaboottsimple, 36
simulaboottstrat, 37
threeawayboot, 43
tunelocal, 46
*Topic **graphical display**
plot.CA3variants, 30
*Topic **index**
tau3, 41
*Topic **indicator**
makeindicator, 20
*Topic **interactive biplot**
plot.CA3variants, 30
*Topic **interactive plot**
plot.CA3variants, 30
*Topic **loss**
loss, 19
*Topic **marginal**
margI, 20
margJ, 21
margK, 22
*Topic **matrix**
makeindicator, 20
*Topic **multivariate**
flatten, 11
Kron, 17
loss1.3, 17
loss1.3ordered, 18
newcomp3, 22
newcomp3ordered, 23
newcomp3ordered1, 24
newcomp3ordered2, 24
nsca3basic, 25
print.CA3variants, 31
reconst3, 35
rstand3, 35
standtab, 38
step.g3, 38
step.g3ordered, 39
step13, 40
step13ordered, 40
tucker, 44
tuckerORDERED, 45
*Topic **nonparametric**
  emerson.poly, 10
*Topic **ordered**
  oca3basic, 26
  onsca3basic, 28
*Topic **partition**
  print.CA3variants, 31
tau3, 41
*Topic **plot**
  caplot3d, 6
*Topic **product**
  Kron, 17
  p.ext, 29
  prod3, 34
*Topic **reconstruction**
  reconst3, 35
*Topic **three way index**
  chi3ordered, 8
tau3ordered, 42
*Topic **three-way index**
  chi3, 7
*Topic **three-way non-symmetric correspondence analysis**
  nsca3basic, 25
*Topic **three-way non-symmetrical correspondence analysis.**
  CA3variants, 5
*Topic **three-way symmetrical correspondence analysis.**
  CA3variants, 5
*Topic **tuckals3**
  p.ext, 29
*Topic **tucker**
  criteria, 10
*Topic **weighted array**
  prod3, 34
*Topic **weight**
  rstand3, 35
ca3basic, 3
caplot3d, 6
caps, 29
caps1.3, 30
caps1.3ordered, 31
caps1.3ordered1, 32
caps1.3ordered2, 33
caps2, 19
caps3, 22
emerson.poly, 10
flatten, 11
flatten, 11
happy, 12
init3, 13
init3ordered, 14
init3ordered1, 15
init3ordered2, 16
Kron, 17
loss1.3, 17
loss1.3ordered, 18
loss2, 19
makeindicator, 20
margI, 20
margI, 21
margK, 22

newcomp3, 22
newcomp3ordered, 23
newcomp3ordered1, 24
newcomp3ordered2, 24
nsca3basic, 25
onsca3basic, 28

oca3basic, 26
olive, 27
onsca3basic, 28
p.ext, 29
plot.CA3variants, 30
print.CA3variants, 31
prod3, 34
reconst3, 35
INDEX

rstand3, 35
simulabootsimple, 36
simulabootstrat, 37
standtab, 38
step.g3, 38
step.g3ordered, 39
stepi3, 40
stepi3ordered, 40

tau3, 41
tau3ordered, 42
threewayboot, 43
tucker, 44
tuckerORDERED, 45
tunelocal, 46