

# Package ‘ca’

January 2, 2012

**Version** 0.33

**Date** 2010-01-15

**Title** Simple, Multiple and Joint Correspondence Analysis

**Author** Michael Greenacre <michael.greenacre@upf.edu>, Oleg Nenadic  
<onenadi@uni-goettingen.de>

**Maintainer** Oleg Nenadic <onenadi@uni-goettingen.de>

**Depends** R (>= 2.0.0), rgl (>= 0.64-10)

**Description** A package for computation and visualization of simple,multiple and joint correspon-  
dence analysis.

**License** GPL

**URL** <http://www.carme-n.org/>

**Repository** CRAN

**Date/Publication** 2010-01-17 10:52:35

## R topics documented:

author	2
ca	2
iterate.mjca	4
mjca	5
pchlist	7
plot.ca	7
plot.mjca	10
plot3d.ca	12
print.ca	14
print.mjca	14
print.summary.ca	15
print.summary.mjca	16
smoke	16

summary.ca . . . . .	17
summary.mjca . . . . .	18
wg93 . . . . .	19

<b>Index</b>	<b>20</b>
--------------	-----------

---

author	<i>Author dataset</i>
--------	-----------------------

---

### Description

This data matrix contains the counts of the 26 letters of the alphabet (columns of matrix) for 12 different novels (rows of matrix). Each row contains letter counts in a sample of text from each work, excluding proper nouns.

### Usage

```
data(author)
```

### Format

Data frame containing the 12 x 26 matrix.

### Source

W.A. Larsen & R. McGill, unpublished data collected in 1973.

---

ca	<i>Simple correspondence analysis</i>
----	---------------------------------------

---

### Description

Computation of simple correspondence analysis.

### Usage

```
ca(obj, nd = NA, suprow = NA, supcol = NA,
    subsetrow = NA, subsetcol = NA)
```

### Arguments

obj	A two-way table of non-negative data, usually frequencies.
nd	Number of dimensions to be included in the output; if NA the maximum possible dimensions are included.
suprow	Indices of supplementary rows.
supcol	Indices of supplementary columns.
subsetrow	Row indices of subset.
subsetcol	Column indices of subset.

## Details

The function `ca` computes a simple correspondence analysis based on the singular value decomposition.

The options `suprow` and `supcol` allow supplementary (passive) rows and columns to be specified. Using the options `subsetrow` and/or `subsetcol` result in a subset CA being performed.

## Value

<code>sv</code>	Singular values
<code>nd</code>	Dimension of the solution
<code>rownames</code>	Row names
<code>rowmass</code>	Row masses
<code>rowdist</code>	Row chi-square distances to centroid
<code>rowinertia</code>	Row inertias
<code>rowcoord</code>	Row standard coordinates
<code>rowsup</code>	Indices of row supplementary points
<code>colnames</code>	Column names
<code>colmass</code>	Column masses
<code>coldist</code>	Column chi-square distances to centroid
<code>colinertia</code>	Column inertias
<code>colcoord</code>	Column standard coordinates
<code>colsup</code>	Indices of column supplementary points

## References

Nenadic, O. and Greenacre, M. (2007). Correspondence analysis in R, with two- and three-dimensional graphics: The `ca` package. *Journal of Statistical Software*, 20 (3), available at <http://www.jstatsoft.org/v20/i03/>

Greenacre, M. (2007). *Correspondence Analysis in Practice*. Second Edition. Chapman & Hall / CRC, London.

Blasius, J. and Greenacre, M. J. (1994), Computation of correspondence analysis, in “Correspondence Analysis in the Social Sciences”, pp. 53-75, Academic Press, London.

Greenacre, M.J. and Pardo, R. (2006), Subset correspondence analysis: visualizing relationships among a selected set of response categories from a questionnaire survey. *Sociological Methods and Research*, 35, pp. 193-218.

## See Also

[svd](#), [plot.ca](#), [plot3d.ca](#), [summary.ca](#), [print.ca](#)

## Examples

```
data(author)
ca(author)
plot(ca(author))
```

---

`iterate.mjca`*Updating a Burt matrix in Joint Correspondence Analysis*

---

### Description

Updating a Burt matrix in Joint Correspondence Analysis based on iteratively weighted least squares.

### Usage

```
iterate.mjca(B, lev.n, nd = 2, maxit = 50, epsilon = 0.0001)
```

### Arguments

<code>B</code>	A Burt matrix.
<code>lev.n</code>	The number of levels for each factor from the original response pattern matrix.
<code>nd</code>	The required dimensionality of the solution.
<code>maxit</code>	The maximum number of iterations.
<code>epsilon</code>	A convergence criterion for the maximum absolute difference of updated values compared to the previous values. The iteration is completed when all differences are smaller than <code>epsilon</code> .

### Details

The function `iterate.mjca` computes the updated Burt matrix. This function is called from the function `mjca` when the option `lambda="JCA"`, i.e. when a Joint Correspondence Analysis is performed.

### Value

<code>B.star</code>	The updated Burt matrix
<code>crit</code>	Vector of length 2 containing the number of iterations and <code>epsilon</code>

### See Also

[mjca](#)

---

mjca *Multiple and joint correspondence analysis*

---

**Description**

Computation of multiple and joint correspondence analysis.

**Usage**

```

mjca(obj, nd = 2, lambda = "adjusted", supcol = NA, subsetcol = NA,
      ps = "", maxit = 50, epsilon = 0.0001)

```

**Arguments**

obj	A response pattern matrix (data frame containing factors).
nd	Number of dimensions to be included in the output; if NA the maximum possible dimensions are included.
lambda	Gives the scaling method. Possible values include "indicator", "Burt", "adjusted" and "JCA". Using lambda = "JCA" results in a joint correspondence analysis using iterative adjustment of the Burt matrix in the solution space.
supcol	Indices of supplementary columns.
subsetcol	Indices of subset categories.
ps	Separator used for combining variable and category names.
maxit	The maximum number of iterations (Joint Correspondence Analysis).
epsilon	A convergence criterion (Joint Correspondence Analysis).

**Details**

The function mjca computes a multiple or joint correspondence analysis based on the eigenvalue decomposition of the Burt matrix.

**Value**

sv	Eigenvalues (lambda = "indicator") or singular values (lambda = "Burt", "adjusted" or "JCA")
lambda	Scaling method
inertia.e	Percentages of explained inertia
inertia.t	Total inertia
inertia.et	Total percentage of explained inertia with the nd-dimensional solution
levelnames	Names of the factor/level combinations
levels.n	Number of levels in each factor
nd	User-specified dimensionality of the solution
nd.max	Maximum possible dimensionality of the solution

rownames	Row names
rowmass	Row masses
rowdist	Row chi-square distances to centroid
rowinertia	Row inertias
rowcoord	Row standard coordinates
colnames	Column names
colmass	Column masses
coldist	Column chi-square distances to centroid
colinertia	Column inertias
colcoord	Column standard coordinates
colsup	Indices of column supplementary points (of the Burt and Indicator matrix)
subsetcol	Indices of subset columns
Burt	Burt matrix
Burt.upd	The updated Burt matrix (JCA only)
subinertia	Inertias of sub-matrices
JCA.iter	Vector of length two containing the number of iterations and the epsilon (JCA only)
call	Return of match.call

## References

- Nenadic, O. and Greenacre, M. (2007), Correspondence analysis in R, with two- and three-dimensional graphics: The ca package. *Journal of Statistical Software*, 20 (3), available at <http://www.jstatsoft.org/v20/i03/>
- Nenadic, O. and Greenacre, M. (2007), Computation of Multiple Correspondence Analysis, with Code in R, in *Multiple Correspondence Analysis and Related Methods* (eds. M. Greenacre and J. Blasius), Chapman & Hall / CRC, Boca Raton, London, New York, pp. 523-551.
- Greenacre, M.J. and Pardo, R. (2006), Subset correspondence analysis: visualizing relationships among a selected set of response categories from a questionnaire survey. *Sociological Methods and Research*, 35, pp. 193-218.

## See Also

[eigen](#), [plot.mjca](#), [summary.mjca](#), [print.mjca](#)

## Examples

```
library(MASS)
data(farms)
mjca(farms)

# Joint correspondence analysis:
mjca(farms, lambda = "JCA")
```

---

pchlist	<i>Listing the set of available symbols.</i>
---------	--

---

**Description**

A plot of the available symbols for use with the option pch.

**Usage**

```
pchlist()
```

**Details**

This function generates a numbered list of the plotting symbols available for use in the functions [plot.ca](#) and [plot3d.ca](#).

**See Also**

[plot.ca](#), [plot3d.ca](#)

**Examples**

```
pchlist()
```

---

plot.ca	<i>Plotting 2D maps in correspondence analysis</i>
---------	--

---

**Description**

Graphical display of correspondence analysis results in two dimensions

**Usage**

```
## S3 method for class 'ca'  
plot(x, dim = c(1,2), map = "symmetric", what = c("all", "all"),  
      mass = c(FALSE, FALSE), contrib = c("none", "none"),  
      col = c("#0000FF", "#FF0000"), pch = c(16, 1, 17, 24),  
      labels = c(2, 2), arrows = c(FALSE, FALSE), ...)
```

**Arguments**

x	Simple correspondence analysis object returned by <a href="#">ca</a>
dim	Numerical vector of length 2 indicating the dimensions to plot on horizontal and vertical axes respectively; default is first dimension horizontal and second dimension vertical.
map	Character string specifying the map type. Allowed options include "symmetric" (default) "rowprincipal" "colprincipal" "symbiplot" "rowgab" "colgab" "rowgreen" "colgreen"
what	Vector of two character strings specifying the contents of the plot. First entry sets the rows and the second entry the columns. Allowed values are "all" (all available points, default) "active" (only active points are displayed) "passive" (only supplementary points are displayed) "none" (no points are displayed) The status (active or supplementary) of rows and columns is set in <a href="#">ca</a> using the options <code>suprow</code> and <code>supcol</code> .
mass	Vector of two logicals specifying if the mass should be represented by the area of the point symbols (first entry for rows, second one for columns)
contrib	Vector of two character strings specifying if contributions (relative or absolute) should be represented by different colour intensities. Available options are "none" (contributions are not indicated in the plot). "absolute" (absolute contributions are indicated by colour intensities). "relative" (relative contributions are indicated by colour intensities). If set to "absolute" or "relative", points with zero contribution are displayed in white. The higher the contribution of a point, the closer the corresponding colour to the one specified by the <code>col</code> option.
col	Vector of length 2 specifying the colours of row and column point symbols, by default blue for rows and red for columns. Colours can be entered in hexadecimal (e.g. "#FF0000"), rgb (e.g. <code>rgb(1, 0, 0)</code> ) values or by R-name (e.g. "red").
pch	Vector of length 4 giving the type of points to be used for row active and supplementary, column active and supplementary points. See <a href="#">pchlist</a> for a list of symbols.
labels	Vector of length two specifying if the plot should contain symbols only (0), labels only (1) or both symbols and labels (2). Setting <code>labels</code> to 2 results in the symbols being plotted at the coordinates and the labels with an offset.
arrows	Vector of two logicals specifying if the plot should contain points (FALSE, default) or arrows (TRUE). First value sets the rows and the second value sets the columns.
...	Further arguments passed to <a href="#">plot</a> and <a href="#">points</a> .

## Details

The function `plot.ca` makes a two-dimensional map of the object created by `ca` with respect to two selected dimensions. By default the scaling option of the map is "symmetric", that is the so-called *symmetric map*. In this map both the row and column points are scaled to have inertias (weighted variances) equal to the principal inertia (eigenvalue or squared singular value) along the principal axes, that is both rows and columns are in principal coordinates. Other options are as follows:

- `"rowprincipal"` or `"colprincipal"` - these are the so-called *asymmetric maps*, with either rows in principal coordinates and columns in standard coordinates, or vice versa (also known as row-metric-preserving or column-metric-preserving respectively). These maps are biplots;
- `"symbiplot"` - this scales both rows and columns to have variances equal to the singular values (square roots of eigenvalues), which gives a symmetric biplot but does not preserve row or column metrics;
- `"rowgab"` or `"colgab"` - these are asymmetric maps (see above) with rows (respectively, columns) in principal coordinates and columns (respectively, rows) in standard coordinates multiplied by the mass of the corresponding point. These are also biplots and were proposed by Gabriel & Odoroff (1990);
- `"rowgreen"` or `"colgreen"` - these are similar to `"rowgab"` and `"colgab"` except that the points in standard coordinates are multiplied by the square root of the corresponding masses, giving reconstructions of the standardized residuals.

This function has options for sizing and shading the points. If the option `mass` is TRUE for a set of points, the size of the point symbol is proportional to the relative frequency (mass) of each point. If the option `contrib` is "absolute" or "relative" for a set of points, the colour intensity of the point symbol is proportional to the absolute contribution of the points to the planar display or, respectively, the quality of representation of the points in the display.

## References

- Gabriel, K.R. and Odoroff, C. (1990). Biplots in biomedical research. *Statistics in Medicine*, 9, pp. 469-485.
- Greenacre, M.J. (1993) *Correspondence Analysis in Practice*. Academic Press, London.
- Greenacre, M.J. (1993) Biplots in correspondence Analysis, *Journal of Applied Statistics*, 20, pp. 251 - 269.

## See Also

[ca](#), [summary.ca](#), [print.ca](#), [plot3d.ca](#), [pchlist](#)

## Examples

```
data(smoke)

# A two-dimensional map with standard settings
plot(ca(smoke))

# Mass for rows and columns represented by the size of the point symbols
```

```

plot(ca(smoke), mass = c(TRUE, TRUE))

# Displaying the column profiles only with masses represented by size of point
# symbols and relative contributions by colour intensity.
# Since the arguments are recycled it is sufficient to give only one argument
# for mass and contrib.
data(author)
plot(ca(author), what = c("none", "all"), mass = TRUE, contrib = "relative")

```

---

plot.mjca

*Plotting 2D maps in multiple and joint correspondence analysis*


---

## Description

Graphical display of multiple and joint correspondence analysis results in two dimensions

## Usage

```

## S3 method for class 'mjca'
plot(x, dim = c(1,2), map = "symmetric", centroids = FALSE,
     what = c("all", "all"), mass = c(FALSE, FALSE),
     contrib = c("none", "none"), col = c("#000000", "#FF0000"),
     pch = c(16, 1, 17, 24), labels = c(2, 2),
     arrows = c(FALSE, FALSE), ...)

```

## Arguments

x	Multiple or joint correspondence analysis object returned by <code>mjca</code>
dim	Numerical vector of length 2 indicating the dimensions to plot on horizontal and vertical axes respectively; default is first dimension horizontal and second dimension vertical.
map	Character string specifying the map type. Allowed options include "symmetric" (default), "rowprincipal", "colprincipal", "symbiplot", "rowgab", "colgab", "rowgreen", "colgreen"
centroids	Logical indicating if column centroids should be added to the plot
what	Vector of two character strings specifying the contents of the plot. First entry sets the rows and the second entry the columns. Allowed values are "all" (all available points, default), "active" (only active points are displayed), "passive" (only supplementary points are displayed)

	"none" (no points are displayed) The status (active or supplementary) of columns is set in <code>mjca</code> using the option <code>supcol</code> .
<code>mass</code>	Vector of two logicals specifying if the mass should be represented by the area of the point symbols (first entry for rows, second one for columns)
<code>contrib</code>	Vector of two character strings specifying if contributions (relative or absolute) should be represented by different colour intensities. Available options are "none" (contributions are not indicated in the plot). "absolute" (absolute contributions are indicated by colour intensities). "relative" (relative contributions are indicated by colour intensities). If set to "absolute" or "relative", points with zero contribution are displayed in white. The higher the contribution of a point, the closer the corresponding colour to the one specified by the <code>col</code> option.
<code>col</code>	Vector of length 2 specifying the colours of row and column point symbols, by default black for rows and red for columns. Colours can be entered in hexadecimal (e.g. "#FF0000"), rgb (e.g. <code>rgb(1, 0, 0)</code> ) values or by R-name (e.g. "red").
<code>pch</code>	Vector of length 4 giving the type of points to be used for row active and supplementary, column active and supplementary points. See <code>pchlist</code> for a list of symbols.
<code>labels</code>	Vector of length two specifying if the plot should contain symbols only (0), labels only (1) or both symbols and labels (2). Setting <code>labels</code> to 2 results in the symbols being plotted at the coordinates and the labels with an offset.
<code>arrows</code>	Vector of two logicals specifying if the plot should contain points (FALSE, default) or arrows (TRUE). First value sets the rows and the second value sets the columns.
<code>...</code>	Further arguments passed to <code>plot</code> and <code>points</code> .

## Details

The function `plot.mjca` makes a two-dimensional map of the object created by `mjca` with respect to two selected dimensions. By default the scaling option of the map is "symmetric", that is the so-called *symmetric map*. In this map both the row and column points are scaled to have inertias (weighted variances) equal to the principal inertia (eigenvalue) along the principal axes, that is both rows and columns are in principal coordinates. Other options are as follows:

- `"rowprincipal"` or `"colprincipal"` - these are the so-called *asymmetric maps*, with either rows in principal coordinates and columns in standard coordinates, or vice versa (also known as row-metric-preserving or column-metric-preserving respectively). These maps are biplots;
- `"symbiplot"` - this scales both rows and columns to have variances equal to the singular values (square roots of eigenvalues), which gives a symmetric biplot but does not preserve row or column metrics;
- `"rowgab"` or `"colgab"` - these are asymmetric maps (see above) with rows (respectively, columns) in principal coordinates and columns (respectively, rows) in standard coordinates multiplied by the mass of the corresponding point. These are also biplots and were proposed by Gabriel & Odoroff (1990);

- `-"rowgreen"` or `-"colgreen"` - these are similar to `-"rowgab"` and `-"colgab"` except that the points in standard coordinates are multiplied by the square root of the corresponding masses, giving reconstructions of the standardized residuals.

This function has options for sizing and shading the points. If the option `mass` is `TRUE` for a set of points, the size of the point symbol is proportional to the relative frequency (mass) of each point. If the option `contrib` is `-"absolute"` or `-"relative"` for a set of points, the colour intensity of the point symbol is proportional to the absolute contribution of the points to the planar display or, respectively, the quality of representation of the points in the display.

## References

- Gabriel, K.R. and Odoroff, C. (1990). Biplots in biomedical research. *Statistics in Medicine*, 9, pp. 469-485.
- Greenacre, M.J. (1993) *Correspondence Analysis in Practice*. Academic Press, London.
- Greenacre, M.J. (1993) Biplots in correspondence Analysis, *Journal of Applied Statistics*, 20, pp. 251 - 269.

## See Also

[mjca](#), [summary.mjca](#), [print.mjca](#), [pchlist](#)

## Examples

```
library(MASS)
data(farms)

# A two-dimensional map with standard settings
plot(mjca(farms))

# Mass for columns represented by the size of the point symbols
plot(mjca(farms), mass = c(FALSE, TRUE))
```

---

plot3d.ca

*Plotting 3D maps in correspondence analysis*

---

## Description

Graphical display of correspondence analysis in three dimensions

## Usage

```
plot3d.ca(x, dim = c(1, 2, 3), map = "symmetric", what = c("all", "all"),
  contrib = c("none", "none"), col = c("#6666FF", "#FF6666"),
  labcol = c("#0000FF", "#FF0000"), pch = c(16, 1, 18, 9),
  labels = c(2, 2), sf = 0.00002, arrows = c(FALSE, FALSE), ...)
```

**Arguments**

x	Simple correspondence analysis object returned by <code>ca</code>
dim	Numerical vector of length 2 indicating the dimensions to plot
map	Character string specifying the map type. Allowed options include "symmetric" (default) "rowprincipal" "colprincipal" "symbiplot" "rowgab" "colgab" "rowgreen" "colgreen"
what	Vector of two character strings specifying the contents of the plot. First entry sets the rows and the second entry the columns. Allowed values are "none" (no points are displayed) "active" (only active points are displayed, default) "supplementary" (only supplementary points are displayed) "all" (all available points) The status (active or supplementary) is set in <code>ca</code> .
contrib	Vector of two character strings specifying if contributions (relative or absolute) should be indicated by different colour intensities. Available options are "none" (contributions are not indicated in the plot). "absolute" (absolute contributions are indicated by colour intensities). "relative" (relative contributions are indicated by colour intensities). If set to "absolute" or "relative", points with zero contribution are displayed in white. The higher the contribution of a point, the closer the corresponding colour to the one specified by the <code>col</code> option.
col	Vector of length 2 specifying the colours of row and column profiles. Colours can be entered in hexadecimal (e.g. "#FF0000"), rgb (e.g. <code>rgb(1,0,0)</code> ) values or by R-name (e.g. "red").
labcol	Vector of length 2 specifying the colours of row and column labels.
pch	Vector of length 2 giving the type of points to be used for rows and columns.
labels	Vector of length two specifying if the plot should contain symbols only (0), labels only (1) or both symbols and labels (2). Setting <code>labels</code> to 2 results in the symbols being plotted at the coordinates and the labels with an offset.
sf	A scaling factor for the volume of the 3d primitives.
arrows	Vector of two logicals specifying if the plot should contain points (FALSE, default) or arrows (TRUE). First value sets the rows and the second value sets the columns.
...	Further arguments passed to the <code>rgl</code> functions.

**See Also**

[ca](#)

---

print.ca                      *Printing ca objects*

---

### Description

Printing method for correspondence analysis objects

### Usage

```
## S3 method for class 'ca'  
print(x, ...)
```

### Arguments

x	Simple correspondence analysis object returned by <a href="#">ca</a>
...	Further arguments are ignored

### Details

The function `print.ca` gives the basic statistics of the `ca` object. First the eigenvalues (that is, principal inertias) and their percentages with respect to total inertia are printed. Then for the rows and columns respectively, the following are printed: the masses, chi-square distances of the points to the centroid (i.e., centroid of the active points), point inertias (for active points only) and principal coordinates on the first `nd` dimensions requested (default = 2 dimensions). The function [summary.ca](#) gives more detailed results about the inertia contributions of each point on each principal axis. For supplementary points, masses and inertias are not applicable.

### See Also

[ca](#)

### Examples

```
data(smoke)  
print(ca(smoke))
```

---

print.mjca                      *Printing mjca objects*

---

### Description

Printing method for multiple and joint correspondence analysis objects

### Usage

```
## S3 method for class 'mjca'  
print(x, ...)
```

**Arguments**

x                    Multiple or joint correspondence analysis object returned by [mjca](#)  
...                   Further arguments are ignored

**Details**

The function `print.mjca` gives the basic statistics of the `mjca` object. First the eigenvalues (that is, principal inertias) and their percentages with respect to total inertia are printed. Then for the rows and columns respectively, the following are printed: the masses, chi-square distances of the points to the centroid (i.e., centroid of the active points), point inertias (for active points only) and principal coordinates on the first `nd` dimensions requested (default = 2 dimensions). The function [summary.mjca](#) gives more detailed results about the inertia contributions of each point on each principal axis.

For supplementary points, masses and inertias are not applicable.

**See Also**

[mjca](#)

**Examples**

```
library(MASS)
data(farms)
print(mjca(farms))
```

---

print.summary.ca            *Printing summaries of ca objects*

---

**Description**

Printing method for summaries of correspondence analysis objects

**Usage**

```
## S3 method for class 'summary.ca'
print(x, ...)
```

**Arguments**

x                    Summary of a simple correspondence analysis object returned by [summary.ca](#)  
...                   Further arguments are ignored

**See Also**

[ca](#), [summary.ca](#)

---

`print.summary.mjca`      *Printing summaries of mjca objects*

---

### Description

Printing method for summaries of multiple and joint correspondence analysis objects

### Usage

```
## S3 method for class 'summary.mjca'  
print(x, ...)
```

### Arguments

`x`                      summary of a multiple or joint correspondence analysis object returned by `summary.mjca`  
`...`                    Further arguments are ignored

### See Also

[mjca](#), [summary.mjca](#)

---

`smoke`                      *Smoke dataset*

---

### Description

Artificial dataset in Greenacre (1984)

### Usage

```
data(smoke)
```

### Format

Table containing 5 rows (staff group) and 4 columns (smoking categories), giving the frequencies of smoking categories in each staff group in a fictional organization.

### References

Greenacre(1984)

---

`summary.ca`*Summarizing simple correspondence analysis*

---

## Description

Textual output summarizing the results of `ca`, including a scree-plot of the principal inertias and row and column contributions.

## Usage

```
## S3 method for class 'ca'  
summary(object, scree = TRUE, ...)
```

## Arguments

<code>object</code>	Simple correspondence analysis object returned by <code>ca</code> .
<code>scree</code>	Logical flag specifying if a scree-plot should be included in the output.
<code>...</code>	Further arguments (ignored)

## Details

The function `summary.ca` gives the detailed numerical results of the `ca` function. All the eigenvalues (principal inertias) are listed, their percentages with respect to total inertia, and a bar chart (also known as a scree plot). Then for the set of rows and columns a table of results is given in a standard format, where quantities are either multiplied by 1000 or expressed in permills (thousandths): the mass of each point (x1000), the quality of display in the solution subspace of `nd` dimensions, the inertia of the point (in permills of the total inertia), and then for each dimension of the solution the principal coordinate (x1000), the (relative) contribution COR of the principal axis to the point inertia (x1000) and the (absolute) contribution CTR of the point to the inertia of the axis (in permills of the principal inertia).

For supplementary points, masses, inertias and absolute contributions (CTR) are not applicable, but the relative contributions (COR) are valid as well as their sum over the set of chosen `nd` dimensions (QLT).

## Examples

```
data(smoke)  
summary(ca(smoke))
```

---

`summary.mjca`*Summarizing multiple and joint correspondence analysis*

---

### Description

Textual output summarizing the results of `mjca`, including a scree-plot of the principal inertias and row and column contributions.

### Usage

```
## S3 method for class 'mjca'  
summary(object, scree = TRUE, rows = FALSE, ...)
```

### Arguments

<code>object</code>	Multiple or joint correspondence analysis object returned by <code>mjca</code> .
<code>scree</code>	Logical flag specifying if a scree-plot should be included in the output.
<code>rows</code>	Logical specifying whether the results for the rows should be included in the output (default = FALSE).
<code>...</code>	Further arguments (ignored)

### Details

The function `summary.mjca` gives the detailed numerical results of the `mjca` function. All the eigenvalues (principal inertias) are listed, their percentages with respect to total inertia, and a bar chart (also known as a scree plot). Then for the set of rows and columns a table of results is given in a standard format, where quantities are either multiplied by 1000 or expressed in permills (thousandths): the mass of each point (x1000), the quality of display in the solution subspace of `nd` dimensions, the inertia of the point (in permills of the total inertia), and then for each dimension of the solution the principal coordinate (x1000), the (relative) contribution COR of the principal axis to the point inertia (x1000) and the (absolute) contribution CTR of the point to the inertia of the axis (in permills of the principal inertia).

For supplementary points, masses, inertias and absolute contributions (CTR) are not applicable, but the relative contributions (COR) are valid as well as their sum over the set of chosen `nd` dimensions (QLT).

### Examples

```
library(MASS)  
data(farms)  
summary(mjca(farms))
```

---

wg93

*International Social Survey Program on Environment 1993 - western  
German sample*

---

**Description**

This data frame contains records of four questions on attitude towards science with responses on a five-point scale (1=agree strongly to 5=disagree strongly) and three demographic variables (sex, age and education).

**Usage**

```
data(wg93)
```

**Format**

Data frame (871x7).

**Source**

ISSP (1993). International Social Survey Program: Environment. <http://www.issp.org>

# Index

## \*Topic **datasets**

author, 2  
smoke, 16  
wg93, 19

## \*Topic **multivariate**

ca, 2  
iterate.mjca, 4  
mjca, 5

author, 2

ca, 2, 8, 9, 13–15, 17

eigen, 6

iterate.mjca, 4

mjca, 4, 5, 10–12, 15, 16, 18

pchlist, 7, 8, 9, 11, 12

plot, 8, 11

plot.ca, 3, 7, 7

plot.mjca, 6, 10

plot3d.ca, 3, 7, 9, 12

points, 8, 11

print.ca, 3, 9, 14

print.mjca, 6, 12, 14

print.summary.ca, 15

print.summary.mjca, 16

smoke, 16

summary.ca, 3, 9, 14, 15, 17

summary.mjca, 6, 12, 15, 16, 18

svd, 3

wg93, 19