

Package ‘CAinterprTools’

May 7, 2026

Title Graphical Aid in Correspondence Analysis Interpretation and Significance Testings

Version 1.1.0

Description Allows to plot a number of information related to the interpretation of Correspondence Analysis' results. It provides the facility to plot the contribution of rows and columns categories to the principal dimensions, the quality of points display on selected dimensions, the correlation of row and column categories to selected dimensions, etc. It also allows to assess which dimension(s) is important for the data structure interpretation by means of different statistics and tests. The package also offers the facility to plot the permuted distribution of the table total inertia as well as of the inertia accounted for by pairs of selected dimensions. Different facilities are also provided that aim to produce interpretation-oriented scatterplots. Reference: Alberti 2015 <[doi:10.1016/j.softx.2015.07.001](https://doi.org/10.1016/j.softx.2015.07.001)>.

Depends R (>= 3.1.0)

Imports ca (>= 0.70), classInt (>= 0.2-3), cluster (>= 2.0.7), FactoMineR (>= 1.40), ggplot2 (>= 3.0.0), ggrepel (>= 0.8.0), graphics (>= 3.4.3), Hmisc (>= 4.1-1), RcmdrMisc (>= 1.0), reshape2 (>= 1.4.3), stats (>= 3.4.3), utils (>= 3.4.3)

License GPL

LazyData true

Encoding UTF-8

RoxygenNote 7.1.0

NeedsCompilation no

Author Gianmarco Alberti [aut, cre]

Maintainer Gianmarco Alberti <gianmarcoalberti@gmail.com>

Repository CRAN

Date/Publication 2020-06-05 22:00:09 UTC

Contents

CAinterprTools-package	2
aver.rule	4

brand_coffee	4
breakfast	5
caCluster	5
caCorr	8
caPercept	9
caPlot	10
caPlus	12
caScatter	14
cols.cnt	15
cols.cnt.scatter	16
cols.corr	17
cols.corr.scatter	19
cols.qtl	20
diseases	20
fire_loss	21
greenacre_data	21
groupByCoord	22
malinvaud	23
rescale	23
rows.cnt	24
rows.cnt.scatter	26
rows.corr	27
rows.corr.scatter	28
rows.qtl	29
sig.dim.perm	30
sig.dim.perm.scree	31
sig.tot.inertia.perm	32
table.collapse	32
Index	34

CAinterprTools-package

*Package for Graphical Aid in Correspondence Analysis Interpretation
and Significance Testings*

Description

CAinterTools is a package that allows to plot a number of information related to the interpretation of Correspondence Analysis' results. It provides the facility to plot the contribution of rows and columns categories to the principal dimensions, the quality of points display on selected dimensions, the correlation of row and column categories to selected dimensions, etc. It also allows to assess which dimension(s) is important for the data structure interpretation by means of different statistics and tests. The package also offers the facility to plot the permuted distribution of the table total inertia as well as of the inertia accounted for by pairs of selected dimensions. Different facilities are also provided that aim to produce interpretation-oriented scatteplots.

Details

Package: CAinterprTools
Type: Package
Version: 1.0.0
Date: 2018-05
License: GPL

Author(s)

Gianmarco ALBERTI

Maintainer: Gianmarco ALBERTI <gianmarco.alberti@um.edu.mt>

References

Alberti G. 2013, An R script to facilitate Correspondence Analysis. A guide to the use and the interpretation of results from an archaeological perspective, *Archeologia e Calcolatori* 24, 25-54.

Alberti G. 2015, CAinterprTools: An R package to help interpreting Correspondence Analysis' results, *SoftwareX* 1-2, 26-31.

Benzecri J.P. 1992, *Correspondence Analysis Handbook*, New York, Marcel Dekker.

Blasius J., Greenacre M. 1998, *Visualization of Categorical Data*, San Diego-London, Academic Press.

Greenacre M. 2007, *Correspondence Analysis in Practice*, Boca Raton-London-New York, Chapman&Hall/CRC.

Le S., Josse J., Husson F. 2008, FactoMineR: An R package for multivariate analysis, *Journal of Statistical Software*, 25, 1-18.

Nenadic O., Greenacre M. 2007, Correspondence Analysis in R, with two- and three-dimensional graphics: The ca package, *Journal of Statistical Software*, 20, 1-13.

Saporta G. 2006, *Probabilites, analyse des donnees et statistique* (2e ed.), Paris, Editions Technip.

See Also

ca,FactoMineR

`aver.rule`*Average Rule chart*

Description

This function helps locating the number of dimensions that are important for CA interpretation, according to the so-called 'average rule'. The reference line showing up in the returned histogram indicates the threshold for an optimal dimensionality of the solution according to the average rule.

Usage

```
aver.rule(data)
```

Arguments

`data` Name of the dataset (must be in dataframe format).

Examples

```
data(greenacre_data)
aver.rule(greenacre_data)
```

`brand_coffee`*Dataset: Cross-tabulation of coffee brands vs. consumers' opinion*

Description

Cross-tabulation (23x6) of the coffee brands against consumers' opinion.

After: Kennedy R et al, Practical Applications of Correspondence Analysis to Categorical Data in Market Research, in Journal of Targeting Measurement and Analysis for Marketing, 1996

Usage

```
data(brand_coffee)
```

Format

```
dataframe
```

`breakfast`*Dataset: Cross-tabulation of breakfast food vs consumers' opinion*

Description

Cross-tabulation (14x8) of the breakfast food type against consumers' opinion.

After: Bendixen M, A Practical Guide to the Use of Correspondence Analysis in Marketing Research, in Research online 1, 1996, 16-38

Usage

```
data(breakfast)
```

Format

```
dataframe
```

`caCluster`*Clustering row/column categories on the basis of Correspondence Analysis coordinates from a space of user-defined dimensionality.*

Description

This function plots the result of cluster analysis performed on the results of Correspondence Analysis, providing the facility to produce a dendrogram, a silhouette plot depicting the "quality" of the clustering solution, and a scatterplot with points coded according to the cluster membership.

Usage

```
caCluster(  
  data,  
  which = "both",  
  dim = NULL,  
  dist.meth = "euclidean",  
  aggl.meth = "ward.D2",  
  opt.part = FALSE,  
  opt.part.meth = "mean",  
  part = NULL,  
  cex.dndr.lab = 0.85,  
  cex.sil.lab = 0.75,  
  cex.sctpl.lab = 3.5  
)
```

Arguments

<code>data</code>	Contingency table (dataframe format).
<code>which</code>	Takes "both" to cluster both row and column categories; "rows" or "columns" to cluster only row or column categories respectively
<code>dim</code>	Sets the dimensionality of the space whose coordinates are used to cluster the CA categories; it can be an integer or a vector (e.g., <code>c(2,3)</code>) specifying the first and second selected dimension. NULL is the default; it will make the clustering to be based on the maximum dimensionality of the dataset.
<code>dist.meth</code>	Sets the distance method used for the calculation of the distance between categories; "euclidean" is the default (see the help of the <code>dist()</code> function for more info and other methods available).
<code>aggl.meth</code>	Sets the agglomerative method to be used in the dendrogram construction; "ward.D2" is the default (see the help of the <code>hclust()</code> function for more info and for other methods available).
<code>opt.part</code>	Takes TRUE or FALSE (default) if the user wants or doesn't want an optimal partition to be suggested; the latter is based upon an iterative process that seek for the maximization of the average silhouette width.
<code>opt.part.meth</code>	Sets whether the optimal partition method will try to maximize the average ("mean") or median ("median") silhouette width. The former is the default.
<code>part</code>	Integer which sets the number of desired clusters (NULL is default); this will override the optimal cluster solution.
<code>cex.dndr.lab</code>	Sets the size of the dendrogram's labels. 0.85 is the default.
<code>cex.sil.lab</code>	Sets the size of the silhouette plot's labels. 0.75 is the default.
<code>cex.sctpl.lab</code>	Sets the size of the Correspondence Analysis scatterplot's labels. 3.5 is the default.

Details

The function provides the facility to perform hierarchical cluster analysis of row and/or column categories on the basis of Correspondence Analysis result. The clustering is based on the row and/or column categories' coordinates from:

- (1) a high-dimensional space corresponding to the whole dimensionality of the input contingency table;
- (2) a high-dimensional space of dimensionality smaller than the full dimensionality of the input dataset;
- (3) a bi-dimensional space defined by a pair of user-defined dimensions.

To obtain (1), the 'dim' parameter must be left in its default value (NULL);

To obtain (2), the 'dim' parameter must be given an integer (needless to say, smaller than the full dimensionality of the input data);

To obtain (3), the 'dim' parameter must be given a vector (e.g., `c(1,3)`) specifying the dimensions the user is interested in.

The method by which the distance is calculated is specified using the 'dist.meth' parameter, while the agglomerative method is specified using the 'aggl.meth' parameter. By default, they are set to "euclidean" and "ward.D2" respectively.

The user may want to specify beforehand the desired number of clusters (i.e., the cluster solution). This is accomplished feeding an integer into the 'part' parameter. A dendrogram (with rectangles indicating the clustering solution), a silhouette plot (indicating the "quality" of the cluster solution), and a CA scatterplot (with points given colours on the basis of their cluster membership) are returned. Please note that, when a high-dimensional space is selected, the scatterplot will use the first 2 CA dimensions; the user must keep in mind that the clustering based on a higher-dimensional space may not be well reflected on the subspace defined by the first two dimensions only.

Also note:

- if both row and column categories are subject to the clustering, the column categories will be flagged by an asterisk (*) in the dendrogram (and in the silhouette plot) just to make it easier to identify rows and columns;

- the silhouette plot displays the average silhouette width as a dashed vertical line; the dimensionality of the CA space used is reported in the plot's title; if a pair of dimensions has been used, the individual dimensions are reported in the plot's title;

- the silhouette plot's labels end with a number indicating the cluster to which each category is closer.

An optimal clustering solution can be obtained setting the 'opt.part' parameter to TRUE. The optimal partition is selected by means of an iterative routine which locates at which cluster solution the highest average silhouette width is achieved. If the 'opt.part' parameter is set to TRUE, an additional plot is returned along with the silhouette plot. It displays a scatterplot in which the cluster solution (x-axis) is plotted against the average silhouette width (y-axis). A vertical reference line indicate the cluster solution which maximize the silhouette width, corresponding to the suggested optimal partition.

The function returns a list storing information about the cluster membership (i.e., which categories belong to which cluster).

Further info and Disclaimer:

The silhouette plot is obtained from the silhouette() function out from the 'cluster' package (<https://cran.r-project.org/web/packages/cluster/index.html>). For a detailed description of the silhouette plot, its rationale, and its interpretation, see:

- Rousseeuw P J. 1987. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", Journal of Computational and Applied Mathematics 20, 53-65 (<http://www.sciencedirect.com/science/article/pii/03>

For the idea of clustering categories on the basis of the CA coordinates from a full high-dimensional space (or from a subset thereof), see:

- Ciampi et al. 2005. "Correspondence analysis and two-way clustering", SORT 29 (1), 27-4

- Beh et al. 2011. "A European perception of food using two methods of correspondence analysis", Food Quality and Preference 22(2), 226-231

Please note that the interpretation of the clustering when both row AND column categories are used must proceed with caution due to the issue of inter-class points' distance interpretation. For a full description of the issue (also with further references), see:

- Greenacre M. 2007. "Correspondence Analysis in Practice", Boca Raton-London-New York, Chapman&Hall/CRC, 267-268.

See Also

[groupBycoord](#)

Examples

```
data(brand_coffee)
```

```

#displays a dendrogram of row AND column categories
res <- caCluster(brand_coffee, opt.part=FALSE)

#displays a dendrogram for row AND column categories; the clustering is based on the CA
#coordinates from a full high-dimensional space. Rectangles indicating the clusters defined by
#the optimal partition method (see Details). A silhouette plot, a scatterplot, and a CA
#scatterplot with indication of cluster membership are also produced (see Details).
#The cluster membership is stored in the object 'res'.

res <- caCluster(brand_coffee, opt.part=TRUE)

#displays a dendrogram for row categories, with rectangles indicating the clusters defined by the
#optimal partition method (see Details). The clustering is based on a space of dimensionality 4.
#A silhouette plot, a scatterplot, and a CA scatterplot with indication of cluster membership are
#also produced (see Details). The cluster membership is stored in the object 'res'.

res <- caCluster(brand_coffee, which="rows", dim=4, opt.part=TRUE)

#like the above example, but the clustering is based on the coordinates on the sub-space defined
#by a pair of dimensions (i.e., 1 and 4).

res <- caCluster(brand_coffee, which="rows", dim=c(1,4), opt.part=TRUE)

```

caCorr

Chart of correlation between rows and columns categories

Description

This function calculates the strength of the correlation between rows and columns of the contingency table. A reference line indicates the threshold above which the correlation can be considered important.

Usage

```
caCorr(data)
```

Arguments

data Name of the dataset (in dataframe format).

Examples

```

data(greenacre_data)
caCorr(greenacre_data)

```

Description

This command allows to plot a variant of the traditional Correspondence Analysis scatterplots that allows facilitating the interpretation of the results. It aims at producing what in marketing research is called perceptual map, a visual representation of the CA results that seeks to avoid the problem of interpreting inter-spatial distance. It represents only one type of points (say, column points), and "gives names to the axes" corresponding to the major row category contributors to the two selected dimensions.

Usage

```
caPercept(  
  data,  
  x = 1,  
  y = 2,  
  focus = "row",  
  dim.corr = x,  
  guide = FALSE,  
  size.labls = 3  
)
```

Arguments

data	Contingency table, in dataframe format.
x	First dimensions to be plotted.
y	Second dimensions to be plotted.
focus	Takes "row" (default) if the interest is in assessing the contribution of the rows to the definition of the dimensions, "col" if the interest is on the columns.
dim.corr	Dimension for which the points' correlation (column points if focus is set to "row", row points if focus is set to "col") will be computed and used as input value for the size of the points. The default value is the smaller of the two input dimensions (i.e., x).
guide	TRUE or FALSE (default) if the user does or doesn't want the points being given a color code indicating with which of the two selected dimension they have a higher relative correlation.
size.labls	Adjust the size of the characters used in the labels that give names to the axes.

See Also

[caPlot](#)

Examples

```
data(brand_coffee)

caPercept(brand_coffee,1,2,focus="col",dim.corr=1, guide=FALSE)

#In the returned plot, axes are given names according to the major contributing column categories
# (i.e., coffee brands in this dataset), while the points correspond to the row categories
#(i.e., attributes). Points' size is proportional to the correlation of points with the 1st
#dimension. If 'guide' is set to TRUE, the returned plot is similar to the preceding one,
# but the points are given colour according to whether they are more correlated
# (in relative terms) to the first or to the second of the selected dimensions.
# In this example, points flagged with "->Dim 1" are more correlated to the 1st dimension,
# while those flagged with "->Dim 2" have a higher correlation with the 2nd dimension.
```

caPlot

Interpretation-oriented Correspondence Analysis scatterplots, with informative and flexible (non-overlapping) labels.

Description

This function allows to plot different types of CA scatterplots, adding information that are relevant to the CA interpretation. Thanks to the 'ggrepel' package, the labels tends to not overlap so producing a nicely readable chart.

Usage

```
caPlot(
  data,
  x = 1,
  y = 2,
  adv.labls = TRUE,
  cntr = "columns",
  percept = FALSE,
  qlt.thres = NULL,
  dot.size = 2.5,
  cex.labls = 3,
  cex.percept = 3
)
```

Arguments

data	Contingency table, in dataframe format.
x	First of the two desired dimensions to be plotted. 1 is the default.
y	Second of the two desired dimensions to be plotted. 2 is the default.
adv.labls	Logical value, which takes TRUE (default) or FALSE if the user wants or does not want advanced labels to be displayed.

<code>cntr</code>	If <code>adv.labls</code> is TRUE, the <code>'cntr'</code> parameter takes "rows" or "columns" if the user wants the rows' or columns' contribution to the selected dimensions to be shown in the scatterplot.
<code>percept</code>	Takes TRUE or FALSE (default) if the user does or doesn't want the scatterplot to be turned into a perceptual map.
<code>qlt.thres</code>	Sets the quality of the display's threshold under which points will not be given labels. NULL is the default.
<code>dot.size</code>	Sets the size of the scatterplot's dots. 2.5 is the default.
<code>cex.labls</code>	Sets the size of the scatterplot dots' labels. 3 is the default.
<code>cex.percept</code>	Sets the size of the characters displayed in the axes' labels featuring the perceptual map. 3 is the default.

Details

`caPlot()` provides the facility to produce:

(1) a 'regular' (symmetric) scatterplot, in which points' labels only report the categories' names.

(2) a scatterplot with advanced labels. If the user's interest lies (for instance) in interpreting the rows in the space defined by the column categories, by setting the parameter `'cntr'` to "columns" the columns' labels will be coupled with two asterisks within round brackets; each asterisk (if present) will indicate if the category is a major contributor to the definition of the first selected dimension (if the first asterisk to the left is present) and/or if the same category is also a major contributor to the definition of the second selected dimension (if the asterisk to the right is present). The rows' labels will report the correlation (i.e., $\sqrt{\text{COS}^2}$) with the selected dimensions; the correlation values are reported between square brackets; the left-hand side value refers to the correlation with the first selected dimensions, while the right-hand side value refers to the correlation with the second selected dimension. If the parameter `'cntr'` is set to "rows", the row categories' labels will indicate the contribution, and the column categories' labels will report the correlation values.

(3) a perceptual map, in which axes' poles are given names according to the categories (either rows or columns, as specified by the user) having a major contribution to the definition of the selected dimensions; rows' (or columns') labels will report the correlation with the selected dimensions.

The function returns a dataframe containing data about row and column points:

- (a) coordinates on the first selected dimension
- (b) coordinates on the second selected dimension
- (c) contribution to the first selected dimension
- (d) contribution to the second selected dimension
- (e) quality on the first selected dimension
- (f) quality on the second selected dimension
- (g) correlation with the first selected dimension
- (h) correlation with the second selected dimension
- (j) (k) asterisks indicating whether the corresponding category is a major contribution to the first and/or second selected dimension.

See Also

[caPercept](#) , [caPlus](#)

Examples

```

data(brand_coffee)

#displays a 'regular' (symmetric) CA scatterplot, with row and column categories displayed in the
#same space, and with points' labels just reporting the categories' names.
#Relevant information (see description above) are stored in the variable 'res'.

res <- caPlot(brand_coffee,1,2,adv.labls=FALSE)

#displays the CA scatterplot, with the columns' labels indicating which category
# has a major contribution to the definition of the selected dimensions.
# Rows' labels report the correlation (i.e., sqrt(COS2)) with the selected dimensions.

res <- caPlot(brand_coffee,1,2,cntr="columns")

#displays the CA scatterplot, with the rows' labels indicating
#which category has a major contribution to the definition of the selected dimensions.
#Columns' labels report the correlation (i.e., sqrt(COS2)) with the selected dimensions.

res <- caPlot(brand_coffee,1,2,cntr="rows")

#displays the CA scatterplot as a perceptual map;
#the poles of the selected dimensions will be given names according
#to the column categories that have a major contribution to the definition
#of the selected dimensions. Rows' labels report the correlation (i.e., sqrt(COS2))
#with the selected dimensions.

res <- caPlot(brand_coffee,1,2,cntr="columns", percept=TRUE)

```

caPlus

Facility for interpretation-oriented CA scatterplot

Description

This function allows to plot Correspondence Analysis scatterplots modified to help interpreting the analysis' results. In particular, the function aims at making easier to understand in the same visual context (a) which (say, column) categories are actually contributing to the definition of given pairs of dimensions, and (b) to eyeball which (say, row) categories are more correlated to which dimension.

Usage

```

caPlus(
  data,
  x = 1,
  y = 2,

```

```

    focus,
    row.suppl = FALSE,
    col.suppl = FALSE,
    oneplot = FALSE,
    inches = 0.35,
    cex = 0.5
  )

```

Arguments

data	Object returned by the FactoMineR's CA() function (see example provided below); if supplementary data (i.e., rows and/or columns) are present, when using CA(), the analyst has to use the proper settings required by that function.
x	First dimensions to be plotted (x=1 by default).
y	Second dimensions to be plotted (y=2 by default).
focus	Takes "R" if the interest is in assessing the contribution of rows to the definition of the dimensions, "C" if the interest is on the columns.
row.suppl	Takes TRUE or FALSE if supplementary row data are present or absent (FALSE is the default value).
col.suppl	Takes TRUE or FALSE if supplementary column data are present or absent (FALSE is the default value).
oneplot	Takes TRUE or FALSE if the analyst wants the four returned charts on the same page (recommended) or on four separate windows (FALSE is the default value).
inches	Numerical value used to resize the size of the points' bubbles (see below); the default value is 0.35.
cex	Numerical value used to set the size of labels' font; the default value is 0.50.

See Also

[caPlot](#), [caPercept](#), [CA](#)

Examples

```

data(greenacre_data)

#performs CA by means of FactoMineR's CA command, and store the result in the object named resCA.
library(FactoMineR)
resCA <- CA(greenacre_data, graph=FALSE)

#If supplementary data are present, the user has to specify which rows and/or columns
#are supplementary into this function (see FactoMineR's documentation).
caPlus(resCA, 1, 2, focus="C", row.suppl=FALSE, col.suppl=FALSE, oneplot=TRUE)

```

`caScatter`*Scatterplot visualization facility*

Description

This function allows to get different types of CA scatterplots. It is just a wrapper for functions from the 'ca' and 'FactoMineR' packages.

Usage

```
caScatter(data, x = 1, y = 2, type)
```

Arguments

<code>data</code>	Name of the contingency table (must be in dataframe format).
<code>x</code>	First dimension to be plotted (x=1 by default).
<code>y</code>	Second dimensions to be plotted (y=2 by default).
<code>type</code>	Type of scatterplot to be returned (see examples).

See Also

[caPlot](#) , [caPercept](#) , [caPlus](#) , [ca](#) , [plot.CA](#) , [HCPC](#)

Examples

```
data(greenacre_data)

# symmetric scatterplot for rows and columns
caScatter(greenacre_data, 1, 2, type=1)

# Standard Biplot; 2 plots are returned:
#one with row-categories vectors displayed, one for columns categories vectors.
caScatter(greenacre_data, 1, 2, type=2)

# scatterplot of row categories with groupings
#shown by different colors; scatterplot for column categories is also returned
caScatter(greenacre_data, 1, 2, type=3)

# 3D scatterplot with cluster tree for row categories;
#scatterplot for column categories is also returned.
caScatter(greenacre_data, 1, 2, type=4)
```

cols.cnt

*Columns contribution chart***Description**

This function allows to calculate the contribution of the column categories to the selected dimension.

Usage

```
cols.cnt(
  data,
  x = 1,
  categ.sort = TRUE,
  corr.thrs = 0,
  leg = TRUE,
  cex.labls = 0.75,
  dotprightm = 5,
  cex.leg = 0.6,
  leg.x.spc = 1,
  leg.y.spc = 1
)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	Dimension for which the column categories contribution is returned (1st dimension by default).
categ.sort	Logical value (TRUE/FALSE) which allows to sort the categories in descending order of contribution to the inertia of the selected dimension. TRUE is set by default.
corr.thrs	Threshold above which the row categories correlation will be displayed in the plot's legend.
leg	Enable (TRUE; default) or disable (FALSE) the legend at the right-hand side of the dot plot.
cex.labls	Adjust the size of the dot plot's labels.
dotprightm	Increases the empty space between the right margin of the dot plot and the left margin of the legend box.
cex.leg	Adjust the size of the legend's characters.
leg.x.spc	Adjust the horizontal space of the chart's legend. See more info from the 'legend' function's help (?legend).
leg.y.spc	Adjust the y interspace of the chart's legend. See more info from the 'legend' function's help (?legend).

Details

The function displays the contribution of the categories as a dot plot. A reference line indicates the threshold above which a contribution can be considered important for the determination of the selected dimension. The parameter `categ.sort=TRUE` sorts the categories in descending order of contribution to the inertia of the selected dimension. At the left-hand side of the plot, the categories' labels are given a symbol (+ or -) according to whether each category is actually contributing to the definition of the positive or negative side of the dimension, respectively. The categories are grouped into two groups: 'major' and 'minor' contributors to the inertia of the selected dimension. At the right-hand side, a legend (which is enabled/disabled using the 'leg' parameter) reports the correlation ($\sqrt{\text{COS}^2}$) of the row categories with the selected dimension. A symbol (+ or -) indicates with which side of the selected dimension each row category is correlated.

See Also

[cols.cntr.scatter](#) , [rows.cntr](#) , [rows.cntr.scatter](#)

Examples

```
data(greenacre_data)

# Plots the contribution of the column
#categories to the 2nd CA dimension, and also displays the contribution to the total inertia.
#The categories are sorted in descending order of contribution
#to the inertia of the selected dimension.

cols.cntr(greenacre_data, 2, categ.sort=TRUE)
```

`cols.cntr.scatter` *Scatterplot for column categories contribution to dimensions*

Description

This function allows to plot a scatterplot of the contribution of column categories to two selected dimensions. Two reference lines (in RED) indicate the threshold above which the contribution can be considered important for the determination of the dimensions. A diagonal line is a visual aid to eyeball whether a category is actually contributing more (in relative terms) to either of the two dimensions. The column categories' labels are coupled with + or - symbols within round brackets indicating which to side of the two selected dimensions the contribution values that can be read off from the chart are actually referring. The first symbol (i.e., the one to the left), either + or -, refers to the first of the selected dimensions (i.e., the one reported on the x-axis). The second symbol (i.e., the one to the right) refers to the second of the selected dimensions (i.e., the one reported on the y-axis).

Usage

```
cols.cntr.scatter(data, x = 1, y = 2, filter = FALSE, cex.labls = 3)
```

Arguments

<code>data</code>	Name of the dataset (must be in dataframe format).
<code>x</code>	First dimension for which the contributions are reported (<code>x=1</code> by default).
<code>y</code>	Second dimension for which the contributions are reported (<code>y=2</code> by default).
<code>filter</code>	Filter the categories in order to only display those who have a major contribution to the definition of the selected dimensions.
<code>cex.labls</code>	Adjust the size of the categories' labels

See Also

[cols.cnttr](#) , [rows.cnttr](#) , [rows.cnttr.scatter](#)

Examples

```
data(greenacre_data)

#Plots the scatterplot of the column categories contribution to dimensions 1&2.

cols.cnttr.scatter(greenacre_data,1,2)
```

`cols.corr`

Chart of columns correlation with a selected dimension

Description

This function allows to calculate the correlation ($\sqrt{\text{COS}^2}$) of the column categories with the selected dimension.

Usage

```
cols.corr(
  data,
  x = 1,
  categ.sort = TRUE,
  filter = FALSE,
  leg = TRUE,
  dotprightm = 5,
  cex.leg = 0.6,
  cex.labls = 0.75,
  leg.x.spc = 1,
  leg.y.spc = 1
)
```

Arguments

<code>data</code>	Name of the dataset (must be in dataframe format).
<code>x</code>	Dimension for which the column categories correlation is returned (1st dimension by default).
<code>categ.sort</code>	Logical value (TRUE/FALSE) which allows to sort the categories in descending order of correlation with the selected dimension. TRUE is set by default.
<code>filter</code>	Filter the row categories listed in the top-right legend, only showing those who have a major contribution to the definition of the selected dimension.
<code>leg</code>	Enable (TRUE; default) or disable (FALSE) the legend at the right-hand side of the dot plot.
<code>dotprightm</code>	Increases the empty space between the right margin of the dot plot and the left margin of the legend box.
<code>cex.leg</code>	Adjust the size of the legend's characters.
<code>cex.labls</code>	Adjust the size of the dot plot's labels.
<code>leg.x.spc</code>	Adjust the horizontal space of the chart's legend. See more info from the 'legend' function's help (?legend).
<code>leg.y.spc</code>	Adjust the y interspace of the chart's legend. See more info from the 'legend' function's help (?legend).

Details

The function displays the correlation of the column categories with the selected dimension; the parameter `categ.sort=TRUE` arrange the categories in decreasing order of correlation. At the left-hand side, the categories' labels show a symbol (+ or -) according to which side of the selected dimension they are correlated, either positive or negative. The categories are grouped into two groups: categories correlated with the positive ('pole +') or negative ('pole -') pole of the selected dimension. At the right-hand side, a legend (which is enabled/disabled using the 'leg' parameter) indicates the row categories' contribution (in permills) to the selected dimension (value enclosed within round brackets), and a symbol (+ or -) indicating whether they are actually contributing to the definition of the positive or negative side of the dimension, respectively. Further, an asterisk (*) flags the categories which can be considered major contributors to the definition of the dimension.

See Also

[cols.corr.scatter](#), [rows.corr](#), [rows.corr.scatter](#)

Examples

```
data(greenacre_data)

#Plots the correlation of the column categories with the 1st CA dimension.
cols.corr(greenacre_data, 1, categ.sort=TRUE)
```

cols.corr.scatter	<i>Scatterplot for column categories correlation with dimensions</i>
-------------------	--

Description

This function allows to plot a scatterplot of the correlation ($\sqrt{\text{COS}^2}$) of column categories with two selected dimensions. A diagonal line is a visual aid to eyeball whether a category is actually more correlated (in relative terms) to either of the two dimensions. The column categories' labels are coupled with two + or - symbols within round brackets indicating to which side of the two selected dimensions the correlation values that can be read off from the chart are actually referring. The first symbol (i.e., the one to the left), either + or -, refers to the first of the selected dimensions (i.e., the one reported on the x-axis). The second symbol (i.e., the one to the right) refers to the second of the selected dimensions (i.e., the one reported on the y-axis).

Usage

```
cols.corr.scatter(data, x = 1, y = 2, cex.labels = 3)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	First dimension for which the correlations are reported (x=1 by default).
y	Second dimension for which the correlations are reported (y=2 by default).
cex.labels	Adjust the size of the categories' labels

See Also

[cols.corr](#) , [rows.corr](#) , [rows.corr.scatter](#)

Examples

```
data(greenacre_data) #load the sample dataset

#Plots the scatterplot of the column categories correlation with dimensions 1&2.
cols.corr.scatter(greenacre_data,1,2)
```

`cols.qlt` *Chart of columns quality of the display*

Description

This function allows you to calculate the quality of the display of the column categories on pairs of selected dimensions.

Usage

```
cols.qlt(data, x = 1, y = 2, categ.sort = TRUE, cex.labls = 0.75)
```

Arguments

<code>data</code>	Name of the dataset (must be in dataframe format).
<code>x</code>	First dimension for which the quality is calculated (x=1 by default).
<code>y</code>	Second dimension for which the quality is calculated (y=2 by default).
<code>categ.sort</code>	Logical value (TRUE/FALSE) which allows to sort the categories in descending order of quality of the representation on the subspace defined by the selected dimensions. TRUE is set by default.
<code>cex.labls</code>	Adjust the size of the dot plot's labels.

See Also

[rows.qlt](#)

Examples

```
data(greenacre_data)

#Plots the quality of the display of the column categories on the 1&2 dimensions.
cols.qlt(greenacre_data, 1,2, categ.sort=TRUE)
```

`diseases` *Dataset: Cross-tabulation of quantity of tobacco smoked daily vs. cause of death*

Description

Cross-tabulation (15x4) of the amount of tobacco smoked on a daily basis (in gramms) against cause of death.

After: Velleman P F, Hoaglin D C, Applications, Basics, and Computing of Exploratory Data Analysis, Wadsworth Pub Co 1984 (Exhibit 8-1)

Usage

```
data(diseases)
```

Format

```
dataframe
```

```
fire_loss
```

Dataset: Cross-tabulation of cause of fire vs. amount of money loss

Description

Cross-tabulation (9x4) of the amount of money loss against cause of fire.

After: Li et al, Influences of Time, Location, and Cause Factors on the Probability of Fire Loss in China: A Correspondence Analysis, in Fire Technology 50(5), 2014, 1181-1200 (table 5)

Usage

```
data(fire_loss)
```

Format

```
dataframe
```

```
greenacre_data
```

Dataset: Cross-tabulation of funding category vs. University faculty

Description

Cross-tabulation (10x5) of funding category against University faculty.

After: Greenacre M, Correspondence Analysis in Practice, Boca Raton-London-New York, Chapman&Hall/CRC 2007 (exhibit 12.1)

Usage

```
data(greenacre_data)
```

Format

```
dataframe
```

groupBycoord	<i>Define groups of categories on the basis of a selected partition into k groups employing the Jenks' natural break method on the selected dimension's coordinates</i>
--------------	---

Description

The function allows to group the row/column categories into k user-defined partitions.

Usage

```
groupBycoord(data, x = 1, k = 3, which = "rows", cex.labels = 0.75)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	Dimension whose coordinates are used to build the partitions.
k	Number of groups.
which	Specify if rows ("rows"; default) or columns ("cols") must be grouped.
cex.labels	Set the size of the labels of the dot chart (0.75 by default).

Details

K groups are created employing the Jenks' natural break method applied on the selected dimension's coordinates. A dot chart is returned representing the categories grouped into the selected partitions. At the bottom of the chart, the Goodness of Fit statistic is also reported. The function also returns a dataframe storing the categories' coordinates on the selected dimension and the group each category belongs to.

See Also

[caCluster](#)

Examples

```
data(greenacre_data)

#divide the row categories into 3 groups on the basis of the coordinates
#of the 1st dimension, and store the result into a 'res' object
res <- groupBycoord(greenacre_data, x=1, k=3, which="rows")
```

`malinvaud`*Malinvaud's test for significance of the CA dimensions*

Description

This function allows you to perform the Malinvaud's test, which assesses the significance of the CA dimensions.

Usage

```
malinvaud(data)
```

Arguments

`data` Name of the dataset (must be in dataframe format).

Details

The function returns both a table in the R console and a plot. The former lists relevant information, among which the significance of each CA dimension. The dot chart graphically represents the p-value of each dimension; dimensions are grouped by level of significance; a red reference lines indicates the 0.05 threshold.

See Also

[sig.dim.perm.scree](#)

Examples

```
data(greenacre_data)

#perform the Malinvaud test using the 'greenacre_data' dataset
#and store the output table in a object named 'res'
res <- malinvaud(greenacre_data)
```

`rescale`*Rescaling row/column categories coordinates between a minimum and maximum value*

Description

This function allows to rescale the coordinates of a selected dimension to be constrained between a minimum and a maximum user-defined value.

Usage

```
rescale(data, x = 1, which = "rows", min.v = 0, max.v = 100)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	Dimension for which the row categories contribution is returned (1st dimension by default).
which	Specify if rows ("rows", default) or columns ("cols") must be grouped.
min.v	Minimum value of the new scale (0 by default).
max.v	Maximum value of the new scale (100 by default).

Details

The rationale of the function is that users may wish to use the coordinates on a given dimension to devise a scale, along the lines of what is accomplished in:

Greenacre M 2002, "The Use of Correspondence Analysis in the Exploration of Health Survey Data", Documentos de Trabajo 5, Fundacion BBVA, pp. 7-39

The function returns a chart representing the row/column categories against the rescaled coordinates from the selected dimension. A dataframe is also returned containing the original values (i.e., the coordinates) and the corresponding rescaled values.

Examples

```
data(greenacre_data)

#rescale the row coordinates between 0 and 10
res <- rescale(greenacre_data, which="rows", min.v=0, max.v=10)
```

rows.cntnr

Rows contribution chart

Description

This function allows to calculate the contribution of the row categories to the selected dimension.

Usage

```
rows.cntnr(
  data,
  x = 1,
  categ.sort = TRUE,
  corr.thrs = 0,
  leg = TRUE,
  cex.labls = 0.75,
```

```

    dotprightm = 5,
    cex.leg = 0.6,
    leg.x.spc = 1,
    leg.y.spc = 1
  )

```

Arguments

<code>data</code>	Name of the dataset (must be in dataframe format).
<code>x</code>	Dimension for which the row categories contribution is returned (1st dimension by default).
<code>categ.sort</code>	Logical value (TRUE/FALSE) which allows to sort the categories in descending order of contribution to the inertia of the selected dimension. TRUE is set by default.
<code>corr.thrs</code>	Threshold above which the column categories correlation will be displayed in the plot's legend.
<code>leg</code>	Enable (TRUE; default) or disable (FALSE) the legend at the right-hand side of the dot plot.
<code>cex.labls</code>	Adjust the size of the dot plot's labels.
<code>dotprightm</code>	Increases the empty space between the right margin of the dot plot and the left margin of the legend box.
<code>cex.leg</code>	Adjust the size of the legend's characters.
<code>leg.x.spc</code>	Adjust the horizontal space of the chart's legend. See more info from the 'legend' function's help (?legend).
<code>leg.y.spc</code>	Adjust the y interspace of the chart's legend. See more info from the 'legend' function's help (?legend).

Details

The function displays the contribution of the categories as a dot plot. A reference line indicates the threshold above which a contribution can be considered important for the determination of the selected dimension. The parameter `categ.sort=TRUE` sorts the categories in descending order of contribution to the inertia of the selected dimension. At the left-hand side of the plot, the categories' labels are given a symbol (+ or -) according to whether each category is actually contributing to the definition of the positive or negative side of the dimension, respectively. The categories are grouped into two groups: 'major' and 'minor' contributors to the inertia of the selected dimension. At the right-hand side, a legend (which is enabled/disabled using the 'leg' parameter) reports the correlation ($\sqrt{\text{COS}^2}$) of the column categories with the selected dimension. A symbol (+ or -) indicates with which side of the selected dimension each column category is correlated.

See Also

[rows.cntf.scatter](#), [cols.cntf](#), [cols.cntf.scatter](#)

Examples

```

data(greenacre_data)

#Plots the contribution of the row categories to the 2nd CA dimension,
#and also displays the contribution to the total inertia.
#The categories are sorted in descending order of contribution to the inertia
#of the selected dimension.
rows.cnt.scatter(greenacre_data, 2, categ.sort=TRUE)

```

rows.cnt.scatter *Scatterplot for row categories contribution to dimensions*

Description

This function allows to plot a scatterplot of the contribution of row categories to two selected dimensions. Two reference lines (in RED) indicate the threshold above which the contribution can be considered important for the determination of the dimensions. A diagonal line is a visual aid to eyeball whether a category is actually contributing more (in relative terms) to either of the two dimensions. The row categories' labels are coupled with + or - symbols within round brackets indicating to which side of the two selected dimensions the contribution values that can be read off from the chart are actually referring. The first symbol (i.e., the one to the left), either + or -, refers to the first of the selected dimensions (i.e., the one reported on the x-axis). The second symbol (i.e., the one to the right) refers to the second of the selected dimensions (i.e., the one reported on the y-axis).

Usage

```
rows.cnt.scatter(data, x = 1, y = 2, filter = FALSE, cex.labls = 3)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	First dimension for which the contributions are reported (x=1 by default).
y	Second dimension for which the contributions are reported (y=2 by default).
filter	Filter the categories in order to only display those who have a major contribution to the definition of the selected dimensions.
cex.labls	Adjust the size of the categories' labels

See Also

[rows.cnt](#) , [cols.cnt](#) , [cols.cnt.scatter](#)

Examples

```
data(greenacre_data)

#Plot the scatterplot of the row categories contribution to dimensions 1&2.
rows.cntr.scatter(greenacre_data,1,2)
```

rows.corr

*Chart of rows correlation with a selected dimension***Description**

This function allows to calculate the correlation ($\sqrt{\text{COS}^2}$) of the row categories with the selected dimension.

Usage

```
rows.corr(
  data,
  x = 1,
  categ.sort = TRUE,
  filter = FALSE,
  leg = TRUE,
  dotprightm = 5,
  cex.leg = 0.6,
  cex.labls = 0.75,
  leg.x.spc = 1,
  leg.y.spc = 1
)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	Dimension for which the row categories correlation is returned (1st dimension by default).
categ.sort	Logical value (TRUE/FALSE) which allows to sort the categories in descending order of correlation with the selected dimension. TRUE is set by default.
filter	Filter the column categories listed in the top-right legend, only showing those who have a major contribution to the definition of the selected dimension.
leg	Enable (TRUE; default) or disable (FALSE) the legend at the right-hand side of the dot plot.
dotprightm	Increases the empty space between the right margin of the dot plot and the left margin of the legend box.
cex.leg	Adjust the size of the legend's characters.
cex.labls	Adjust the size of the dot plot's labels.

leg.x.spc	Adjust the horizontal space of the chart's legend. See more info from the 'legend' function's help (?legend).
leg.y.spc	Adjust the y interspace of the chart's legend. See more info from the 'legend' function's help (?legend).

Details

The function displays the correlation of the row categories with the selected dimension; the parameter `categ.sort=TRUE` arrange the categories in decreasing order of correlation. At the left-hand side, the categories' labels show a symbol (+ or -) according to which side of the selected dimension they are correlated, either positive or negative. The categories are grouped into two groups: categories correlated with the positive ('pole +') or negative ('pole -') pole of the selected dimension. At the right-hand side, a legend (which is enabled/disabled using the 'leg' parameter) indicates the column categories' contribution (in permills) to the selected dimension (value enclosed within round brackets), and a symbol (+ or -) indicating whether they are actually contributing to the definition of the positive or negative side of the dimension, respectively. Further, an asterisk (*) flags the categories which can be considered major contributors to the definition of the dimension.

See Also

[rows.corr.scatter](#) , [cols.corr](#) , [cols.corr.scatter](#)

Examples

```
data(greenacre_data)

#Plots the correlation of the row categories with the 1st CA dimension.
rows.corr(greenacre_data, 1, categ.sort=TRUE)
```

rows.corr.scatter *Scatterplot for row categories correlation with dimensions*

Description

This function allows to plot a scatterplot of the correlation ($\sqrt{\text{COS}^2}$) of row categories with two selected dimensions. A diagonal line is a visual aid to eyeball whether a category is actually more correlated (in relative terms) to either of the two dimensions. The row categories' labels are coupled with two + or - symbols within round brackets indicating to which side of the two selected dimensions the correlation values that can be read off from the chart are actually referring. The first symbol (i.e., the one to the left), either + or -, refers to the first of the selected dimensions (i.e., the one reported on the x-axis). The second symbol (i.e., the one to the right) refers to the second of the selected dimensions (i.e., the one reported on the y-axis).

Usage

```
rows.corr.scatter(data, x = 1, y = 2, cex.labels = 3)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	First dimension for which the correlations are reported (x=1 by default).
y	Second dimension for which the correlations are reported (y=2 by default).
cex.labls	Adjust the size of the categories' labels

See Also

[rows.corr](#) , [cols.corr](#) , [cols.corr.scatter](#)

Examples

```
data(greenacre_data)

#Plots the scatterplot of the row categories correlation with dimensions 1&2.
rows.corr.scatter(greenacre_data,1,2)
```

rows.qlt

Chart of rows quality of the display

Description

This function allows you to calculate the quality of the display of the row categories on pairs of selected dimensions.

Usage

```
rows.qlt(data, x = 1, y = 2, categ.sort = TRUE, cex.labls = 0.75)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	First dimension for which the quality is calculated (x=1 by default).
y	Second dimension for which the quality is calculated (y=2 by default).
categ.sort	Logical value (TRUE/FALSE) which allows to sort the categories in descending order of quality of the representation on the subspace defined by the selected dimensions. TRUE is set by default.
cex.labls	Adjust the size of the dot plot's labels.

See Also

[cols.qlt](#)

Examples

```
data(greenacre_data)

#Plots the quality of the display of the row categories on the 1&2 dimensions.
rows.qlt(greenacre_data,1,2,categ.sort=TRUE)
```

sig.dim.perm

*Permuted significance of CA dimensions***Description**

This function calculates the permuted significance of a pair of selected CA dimensions. Number of permutation set at 999 by default, but can be increased by the user. A scatterplot of the permuted inertia of a pair of selected dimensions is produced. Permuted p.values are reported in the axes' labels and are also returned in a dataframe.

Usage

```
sig.dim.perm(data, x = 1, y = 2, B = 999)
```

Arguments

data	Name of the dataset (must be in dataframe format).
x	First dimension whose significance is calculated (x=1 by default).
y	Second dimension whose significance is calculated (y=2 by default).
B	Number of permutations (999 by default).

Value

The function returns a dataframe storing the permuted p-values of each CA dimension.

See Also

[sig.dim.perm.scree](#)

Examples

```
data(greenacre_data)

#Produces a scatterplot of the permuted inertia of the 1 CA dimension
#against the permuted inertia of the 2 CA dimension.
#The observed inertia of the selected dimensions is displayed as a large red dot;
#pvalues are reported in the axes labels (and are stored in a 'pvalues' object).

pvalues <- sig.dim.perm(greenacre_data, 1,2, B=99)
```

sig.dim.perm.scee	<i>Scree plot to test the significance of CA dimensions by means of a randomized procedure</i>
-------------------	--

Description

This function tests the significance of the CA dimensions by means of permutation of the input contingency table. Number of permutation set at 999 by default, but can be increased by the user. The function return a scree-plot displaying for each dimension the observed eigenvalue and the 95th percentile of the permuted distribution of the corresponding eigenvalue. Observed eigenvalues that are larger than the corresponding 95th percentile are significant at least at alpha 0.05. Permuted p-values are displayed into the chart and also returned as dataframe.

Usage

```
sig.dim.perm.scee(data, B = 999, cex = 0.7, pos = 4, offset = 0.5)
```

Arguments

data	Name of the contingency table (must be in dataframe format).
B	Number of permutations to be used (999 by default).
cex	Controls the size of the labels reporting the p values; see the help documentation of the text() function by typing ?text.
pos	Controls the position of the labels reporting the p values; see the help documentation of the text() function by typing ?text.
offset	Controls the offset of the labels reporting the p values; see the help documentation of the text() function by typing ?text.

Value

The function returns a dataframe storing the permuted p-values of each CA dimension.

See Also

[sig.dim.perm](#)

Examples

```
data(greenacre_data)
pvalues <- sig.dim.perm.scee(greenacre_data, 99)
```

sig.tot.inertia.perm *Permuted significance of the CA total inertia*

Description

This function calculates the permuted significance of CA total inertia. Number of permutation is customizable (set at 999 by default). A frequency distribution histogram of permuted CA total inertia is produced and p.value of the observed total inertia is reported.

Usage

```
sig.tot.inertia.perm(data, B = 999)
```

Arguments

data Name of the dataset (must be in dataframe format).
 B Number of permutations (999 by default).

See Also

[sig.dim.perm.scree](#) , [sig.dim.perm](#)

Examples

```
data(greenacre_data)

#Returns the frequency distribution histogram of the permuted total inertia
#(using 99 permutations). The observed total inertia and the 95th percentile
#of the permuted inertia are also displayed for testing the significance
#of the observed total inertia.

sig.tot.inertia.perm(greenacre_data, 99)
```

table.collapse *Collapse rows and columns of a table on the basis of hierarchical clustering*

Description

This function allows to collapse the rows and columns of the input contingency table on the basis of the results of a hierarchical clustering.

Usage

```
table.collapse(data, graph = FALSE)
```

Arguments

data	Name of the dataset (must be in dataframe format)
graph	Logical (TRUE/FALSE); it takes TRUE if the user wants the row and column profiles dendrograms to be produced.

Details

The function returns a list containing the input table, the rows-collapsed table, the columns-collapsed table, and a table with both rows and columns collapsed. It optionally returns two dendrograms (one for the row profiles, one for the column profiles) representing the clusters.

The hierarchical clustering is obtained using the FactoMineR's 'HCPC()' function.

Rationale: clustering rows and/or columns of a table could interest the users who want to know where a "significant association is concentrated" by "collecting together similar rows (or columns) in discrete groups" (Greenacre M, Correspondence Analysis in Practice, Boca Raton-London-New York, Chapman&Hall/CRC 2007, pp. 116, 120). Rows and/or columns are progressively aggregated in a way in which every successive merging produces the smallest change in the table's inertia. The underlying logic lies in the fact that rows (or columns) whose merging produces a small change in table's inertia have similar profiles. This procedure can be thought of as maximizing the between-group inertia and minimizing the within-group inertia.

A method essentially similar is that provided by the 'FactoMineR' package (Husson F, Le S, Pages J, Exploratory Multivariate Analysis by Example Using R, Boca Raton-London-New York, CRC Press, pp. 177-185). The cluster solution is based on the following rationale: a division into Q (i.e., a given number of) clusters is suggested when the increase in between-group inertia attained when passing from a Q-1 to a Q partition is greater than that from a Q to a Q+1 clusters partition. In other words, during the process of rows (or columns) merging, if the following aggregation raises highly the within-group inertia, it means that at the further step very different profiles are being aggregated.

See Also

[HCPC](#), [plot.CA](#)

Examples

```
data(greenacre_data)

#collapse the table, store the results into an object called 'res', and return 2 dendrograms
res <- table.collapse(greenacre_data, graph=TRUE)
```

Index

- * **aver.rule**
 - aver.rule, 4
 - * **caCluster**
 - caCluster, 5
 - * **caCorr**
 - caCorr, 8
 - * **caPercept**
 - caPercept, 9
 - * **caPlot**
 - caPlot, 10
 - * **caPlus**
 - caPlus, 12
 - * **caScatter**
 - caScatter, 14
 - * **cols.cntr.scatter**
 - cols.cntr.scatter, 16
 - * **cols.cntr**
 - cols.cntr, 15
 - * **cols.corr.scatter**
 - cols.corr.scatter, 19
 - * **cols.corr**
 - cols.corr, 17
 - * **cols.qlt**
 - cols.qlt, 20
 - * **datasets**
 - brand_coffee, 4
 - breakfast, 5
 - diseases, 20
 - fire_loss, 21
 - greenacre_data, 21
 - * **groupBycoord**
 - groupBycoord, 22
 - * **malinvaud**
 - malinvaud, 23
 - * **package**
 - CAinterprTools-package, 2
 - * **rescale**
 - rescale, 23
 - * **rows.cntr.scatter**
 - rows.cntr.scatter, 26
 - * **rows.cntr**
 - rows.cntr, 24
 - * **rows.corr.scatter**
 - rows.corr.scatter, 28
 - * **rows.corr**
 - rows.corr, 27
 - * **rows.qlt**
 - rows.qlt, 29
 - * **sig.dim.perm.scree**
 - sig.dim.perm.scree, 31
 - * **sig.dim.perm**
 - sig.dim.perm, 30
 - * **sig.tot.inertia.perm**
 - sig.tot.inertia.perm, 32
 - * **table.collapse**
 - table.collapse, 32
- aver.rule, 4
- brand_coffee, 4
- breakfast, 5
- CA, 13
- ca, 14
- caCluster, 5, 22
- caCorr, 8
- CAinterprTools
(CAinterprTools-package), 2
- CAinterprTools-package, 2
- caPercept, 9, 11, 13, 14
- caPlot, 9, 10, 13, 14
- caPlus, 11, 12, 14
- caScatter, 14
- cols.cntr, 15, 17, 25, 26
- cols.cntr.scatter, 16, 16, 25, 26
- cols.corr, 17, 19, 28, 29
- cols.corr.scatter, 18, 19, 28, 29
- cols.qlt, 20, 29
- diseases, 20

fire_loss, 21

greenacre_data, 21
groupBycoord, 7, 22

HCPC, 14, 33

malinvaud, 23

plot.CA, 14, 33

rescale, 23

rows.cntr, 16, 17, 24, 26

rows.cntr.scatter, 16, 17, 25, 26

rows.corr, 18, 19, 27, 29

rows.corr.scatter, 18, 19, 28, 28

rows.qlt, 20, 29

sig.dim.perm, 30, 31, 32

sig.dim.perm.scree, 23, 30, 31, 32

sig.tot.inertia.perm, 32

table.collapse, 32