Package 'SensoMineR'

October 2, 2009

Version 1.10

Date 2009-30-00

Title Sensory data analysis with R

Author Francois Husson, Sebastien Le

Maintainer Francois Husson <husson@agrocampus-ouest.fr>

Depends FactoMineR

Description an R package for analysing sensory data

License GPL (>= 2)

URL http://sensominer.free.fr, http://www.agrocampus-rennes.fr/math/SensoMineR

Encoding latin1

Repository CRAN

Date/Publication 2009-10-02 13:33:40

R topics documented:

`ardi` *Automatic Research of DIvergences between scores*

Description

Spot the most singular or particular data with respect to all descriptors and to two qualitative variables and all their possible categories combinations.
Computes the highest differences between all the categories of the variables *product*, *panelist* and all their possible combinations, with respect to a set of quantitative variables (the sensory descriptors).

Usage

```
ardi(donnee, col.p, col.j, firstvar, lastvar = ncol(donnee),
      nbval = 10, center = TRUE, scale = FALSE)
```

Arguments

donnee	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
col.p	the position of the *product* variable
col.j	the position of the *panelist* variable
firstvar	the position of the first sensory descriptor
lastvar	the position of the last sensory descriptor (by default the last column of donnee)
nbval	the number of highest divergences to be displayed
center	by default, data are mean centered by panelist
scale	by default, data are not scaled by panelist

Details

Step 1 For each quantitative variable, means by all the possible combinations (panelist,product) are computed.
Step 2 Then, data are mean centered and scaled to unit variance by descriptor and the divergence corresponds to the absolute value of the entries.
Step 3 Means on divergences are computed by products or by panelists and then sorted.

Value

A list containing the following elements:

tab	a data frame (descriptors are mean centered per panelist and scaled to unit variance)
panelist	a data frame, by default the 10 highest divergences between panelists according to the sensory descriptors
product	a data frame, by default the 10 highest divergences between products according to the sensory descriptors
combination	a data frame, by default the 10 highest divergences between panelists and products according to the sensory descriptors

Author(s)

François Husson, Sébastien Lê

See Also

decat

Examples

```
## Not run:
data(chocolates)
ardi(sensochoc, col.p = 4, col.j = 1, firstvar = 5)

## In the case where there's one particular variable of interest
```

```
ardi(sensochoc, col.p = 4, col.j = 1, firstvar = 7, lastvar = 7)
## End(Not run)
```

`averagetable` *Computes a (products,descriptors) matrix*

Description

Returns the (products,descriptors) matrix with entries the means over panelists and sessions.
Computes analyses of variance automatically for a given model and a set of quantitative variables.
Returns a data matrix where each row is associated with each category of a given categorical variable (in most cases, the categorical variable is the *product* variable), each column is associated with a quantitative variable, and each cell is the corresponding adjusted mean or mean.
Computes the average data table with respect to a categorical variable and a set of quantitative variables.

Usage

```
averagetable(donnee, formul, subset = NULL, method = "coeff",
    firstvar, lastvar = ncol(donnee), file = NULL)
```

Arguments

`donnee`	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
`formul`	the model with respect to which the factor levels of the categorical variable of interest are calculated
`subset`	an optional vector specifying a subset of observations to be used in the fitting process
`method`	two possibilities, "coeff" (by default) or "mean"
`firstvar`	the position of the first endogenous variable
`lastvar`	the position of the last endogenous variable (by default the last column of `donnee`)
`file`	the name of the output file (by default, NULL and results are not in a file)

Details

The `formul` parameter can be filled in for a given analysis of variance model. The `formul` parameter must begin with the categorical variable of interest (generally the *product* variable) followed by the different other factors (and eventually their interactions) of interest. Classicially, one can used `formul = "~Product+Panelist+Product:Panelist"`. In practise and in our type of applications, this function is very useful to obtain a data matrix in which rows represent products and columns represent sensory descriptors.
If "mean" is assigned to the `method` parameter, then the `formul` parameter can be restricted to the sole variable of interest (generally the *product* variable).
If data are balanced, the two options "mean" and "coeff" give the same results.

Value

Return a matrix of dimension (*p,q*), where *p* is the number of categories of the qualitative variable of interest (in most cases, *p* is the number of products) and *q* is the number of (sensory) descriptors. If "coeff" is assigned to the `method` parameter then the function *averagetable* returns the matrix of the adjusted means; if "mean" is assigned to the `method` parameter then the function averagetable returns the matrix of the means per category.

Author(s)

François Husson ⟨François.Husson@agrocampus-rennes.fr⟩

References

P. Lea, T. Naes, M. Rodbotten. *Analysis of variance for sensory data*.
H. Sahai, M. I. Ageel. *The analysis of variance*.

See Also

`aov`

Examples

```
data(chocolates)
resaverage<-averagetable(sensochoc, formul = "~Product+Panelist",
    firstvar = 5)
coltable(magicsort(resaverage), level.upper = 6,level.lower = 4,
    main.title = "Average by chocolate")

res.pca = PCA(resaverage, scale.unit = TRUE)
```

---

barrow *Barplot per row with respect to a set of quantitative variables*

Description

Returns as many barplots as there are rows in a matrix. The barplots are automatically generated for all the quantitative variables.

Usage

```
barrow(donnee, numr = 2, numc = 2, numchar = 8, color = "lightblue",
      title = NULL)
```

Arguments

donnee	a data frame of dimension (p,q), where p is the number of products and q is the number of sensory descriptors for instance
numr	the number of barplots to be displayed per row (by default 2)
numc	the number of barplots to be displayed per column (by default 2)
numchar	the number of character used to write the boxplot labels (by default 8)
color	the color of the barplots (by default "lightblue")
title	the title used in the graphs

Details

Missing values are ignored when forming barplots.

Author(s)

Sébastien Lê ⟨Sebastien.Le@agrocampus-rennes.fr⟩

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.
Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) *Graphical Methods for Data Analysis*. Wadsworth & Brooks/Cole.

See Also

plot

Examples

```
data(chocolates)
resdecat<-decat(sensochoc, formul = "~Product+Panelist", firstvar = 5,
    graph = FALSE)
## Not run:
barrow(resdecat$tabT)
barrow(resdecat$coeff, color = "orange")
## End(Not run)
```

---

boxprod *Boxplot per category with respect to a categorical variable and a set of quantitative variables*

Description

Returns as many boxplots as there are categories for a given categorical variable of interest (in most cases, the *product* variable). The boxplots are automatically generated for all the quantitative variables (in our type of applications, variables are often sensory descriptors).

Usage

```
boxprod(donnee, col.p, firstvar, lastvar = ncol(donnee),
    numr = 2, numc = 2)
```

Arguments

donnee	a data frame
col.p	the position of the categorical variable of interest
firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee)
numr	the number of boxplots per row (by default 2)
numc	the number of boxplots per column (by default 2)

Details

Missing values are ignored when forming boxplots.

Author(s)

François Husson ⟨François.Husson@agrocampus-rennes.fr⟩
Sébastien Lê ⟨Sebastien.Le@agrocampus-rennes.fr⟩

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.
Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) *Graphical Methods for Data Analysis*. Wadsworth & Brooks/Cole.

See Also

boxplot which does the computation, bxp for the plotting and more examples; and stripchart for an alternative (with small data sets).

Examples

```
data(chocolates)
boxprod(sensochoc, col.p = 4, firstvar = 5, numr = 2, numc = 2)
```

carto *Preference Mapping Techniques*

Description

Performs preference mapping techniques based on multidimensional exploratory data analysis.

Usage

```
carto(Mat, MatH,
      level = 0, regmod = 1, coord = c(1, 2), asp = 1,
      cex = 1.3, col = "steelblue4", font = 2, clabel = 0.8,
      label.j = FALSE, resolution = 200, nb.clusters = 0)
```

Arguments

Mat	a data frame corresponding to the axes of the map
MatH	a data frame in which each row represent a product and each column represent the hedonic scores of a given consumer for the products
level	the number of standard deviations used in the calculation of the preference response surface for all the consumers
regmod	the type of regression model used in the calculation of the preference response surface for all the consumers. regmod = 1: quadratic model, regmod = 2: vector model, regmod = 3: circular model, regmod = 4: elliptical model
coord	a vector of length 2, the rank of the axis used to display the results if "manual" is not assigned to the option parameter
asp	if 1 is assigned to that parameter, the graphic displays are output in an orthonormal coordinate system
cex	cf. function par in the **graphics** package
col	cf. function par in the **graphics** package
font	cf. function par in the **graphics** package
clabel	cf. the **ade4** package
label.j	boolean, if T then the labels of the panelists who gave the hedonic scores are displayed
resolution	resolution of the map
nb.clusters	number of clusters to use (by default, 0 and the optimal numer of clusters is calculated

Details

The preference mapping methods are commonly used in the fields of market research and research and development to explore and understand the structure and tendencies of consumer preferences, to link consumer preference information to other data and to predict the behavior of consumers in terms of acceptance of a given product.
This function refers to the method introduced by M. Danzart. A response surface is computed per consumer; then according to certain threshold preference zones are delimited and finally superimposed.

Author(s)

François Husson ⟨François.Husson@agrocampus-rennes.fr⟩
Sébastien Lê ⟨Sebastien.Le@agrocampus-rennes.fr⟩

References

Danzart M., Sieffermann J.M., Delarue J. (2004). New developments in preference mapping techniques: finding out a consumer optimal product, its sensory profile and the key sensory attributes. *7th Sensometrics Conference, July 27-30, 2004, Davis, CA*.

See Also

MFA, GPA

Examples

```
## Example 1: carto for the sensory descriptors
data(cocktail)
res.pca <- PCA(senso.cocktail)
carto(res.pca$ind$coord[,1:2], hedo.cocktail)

## Example 2
## Not run:
data(cocktail)
res.mfa <- MFA(cbind.data.frame(senso.cocktail,compo.cocktail),
    group=c(ncol(senso.cocktail),ncol(compo.cocktail)),
    name.group=c("senso","compo"))
carto(res.mfa$ind$coord[,1:2], hedo.cocktail)
## End(Not run)
```

---

chocolates *Chocolates data*

Description

The data used here refer to six varieties of chocolates sold in France.

- For the sensory description: each chocolate was evaluated twice by 29 panelists according to 14 sensory descriptors;
- For the hedonic data: each chocolate was evaluated on a structured scale from 0 to 10, by 222 consumers, according to their liking (0) or disliking (10);
- For the sensory panels description: each chocolate was evaluated by 7 panels according to 14 sensory descriptors.

Usage

```
data(chocolates)
```

Format

There are three data frames: - sensochoc: a data frame with 348 rows and 19 columns: 5 qualitative variables (Panelist, Session, Form, Rank, Product) and 14 sensory descriptors;
- hedochoc: a data frame with 6 rows and 222 columns: each row corresponds to a chocolate and each column to the hedonic scores given by one of the 222 consumers participating in the study;
- sensopanels: a data frame with 6 rows and 98 columns: each row corresponds to a chocolate and each column to the mean over the panelists of a given panel according to a sensory descriptor.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(chocolates)
decat(sensochoc, formul = "~Product+Panelist", firstvar = 5, graph = FALSE)
```

---

cocktail *Cocktail data*

Description

The data used here refer to 16 cocktails.

There are 3 files corresponding to the composition of the cocktails; the sensory description of the cocktails; the hedonic scores.

- For the composition of the cocktails: The mango, banana, orange and lemon concentration are known;
- For the sensory description: each cocktail was evaluated by 12 panelists according to 13 sensory descriptors (only the average of each cocktail are given). - For the hedonic data: each cocktail was evaluated on a structured scale from 0 to 10, by 100 consumers, according to their liking (0) or disliking (10).

Usage

```
data(cocktail)
```

Format

There are three data frames: - compo.cocktail: a data frame with 16 rows and 4 columns: the composition of each cocktail is given for the 4 ingredients;
- senso.cocktail: a data frame with 16 rows and 13 columns: each cocktail was evaluated by 12 panelists according to 13 sensory descriptors;
hedo.cocktail: a data frame with 16 rows and 100 columns: each cocktail was evaluated on a structured scale from 0 to 10, by 100 consumers, according to their liking (0) or disliking (10).

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(cocktail)
```

---

coltable *Color the cells of a data frame according to 4 threshold levels*

Description

Return a colored display of a data frame according to 4 threshold levels.

Usage

```
coltable(matrice, col.mat = matrice,
    nbrow = nrow(matrice), nbcol = ncol(matrice),
    level.lower = 0.05, col.lower = "mistyrose",
    level.upper = 1.96, col.upper = "lightblue",
    cex = 0,nbdec = 4, main.title = NULL, level.lower2 = -1e10,
    col.lower2 = "red", level.upper2 = 1e10,
    col.upper2 = "blue", novalue = FALSE)
```

Arguments

matrice	a data frame (or a matrix) with only quantitative variables
col.mat	a data frame (or a matrix) from which the cells of the matrice data frame are colored; by default, col.mat=matrice
nbrow	the number of rows to be displayed (by default, nrow(matrice))
nbcol	the number of columns to be displayed (by default, ncol(matrice))
level.lower	the threshold below which cells are colored in col.lower
col.lower	the color used for level.lower

level.upper	the threshold above which cells are colored in col.upper
col.upper	the color used for level.upper
cex	cf. function par in the **graphics** package
nbdec	the number of decimal places displayed
main.title	title of the graph(s)
level.lower2	the threshold below which cells are colored in col.lower2; this level should be less than level.lower
col.lower2	the color used for level.lower2
level.upper2	the threshold above which cells are colored in col.upper2; this level should be greater than level.upper
col.upper2	the color used for level.upper2
novalue	boolean, if TRUE the values are not written

Details

This function is very useful especially when there are a lot of values to check.

Author(s)

François Husson, Sébastien Lê

Examples

```
## Example 1
data(chocolates)
resdecat<-decat(sensochoc, formul = "~Product+Panelist", firstvar = 5,
    graph = FALSE)
resaverage<-averagetable(sensochoc, formul = "~Product+Panelist",
    firstvar = 5)
resaverage.sort = resaverage[rownames(magicsort(resdecat$tabT)),
    colnames(magicsort(resdecat$tabT))]
coltable(resaverage.sort, magicsort(resdecat$tabT),
    level.lower = -1.96, level.upper = 1.96,
    main.title = "Average by chocolate")

## Example 3
## Not run:
data(chocolates)
resperf<-paneliperf(sensochoc,
    formul = "~Product+Panelist+Product:Panelist",
    formul.j = "~Product", col.j = 1, firstvar = 5, lastvar = 12,
    synthesis = FALSE, graph = FALSE)
resperfprob<-magicsort(resperf$prob.ind, method = "median")
coltable(resperfprob, level.lower = 0.05, level.upper = 1,
    main.title = "P-value of the F-test (by panelist)")

resperfr2<-magicsort(resperf$r2.ind, method = "median",
    ascending = FALSE)
coltable(resperfr2, level.lower = 0.00, level.upper = 0.85,
```

```
    main.title = "Adjusted R-square (by panelist)")
## End(Not run)
```

`compo.cocktail` *Composition of the cocktails data*

Description

The data used here refer to the composition of 16 cocktails, i.e. the mango, banana, orange and lemon concentration.

Usage

```
data(cocktail)
```

Format

A data frame with 16 rows and 4 columns: the composition of each cocktail is given for the 4 ingredients.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(cocktail)
```

`construct.axes` *Coordinates of individuals and illustrative individuals for PCA or MFA*

Description

This function is especially designed to be used in a sensory data analysis context. Returns the coordinates of the products when performing either PCA or MFA and the coordinates of the "partial" products when performing MFA. Returns also the panelists' coordinates when projected as illustrative rows onto the products' space. Produces graphs of products and descriptors from the output of PCA or MFA.

Usage

```
construct.axes(matrice, coord = c(1,2), scale.unit = TRUE, group = NULL,
      name.group = NULL, centerbypanelist = FALSE, scalebypanelist = FALSE,
      method = "coeff")
```

Arguments

matrice	a data.frame made up of at least two qualitative variables (the *panelist* and the *product* variables), the others are sensory descriptors used to perform an MFA or a PCA if group = NULL
coord	a length 2 vector specifying the components to plot
scale.unit	boolean, if TRUE the descriptors are scaled to unit variance
group	the number of variables in each group of variables when multiple factor analysis is performed (by default this parameter equals NULL and a PCA is performed)
name.group	the names of the groups of variables when mfa is performed (if group differs from NULL)
centerbypanelist	center the data by panelist before the construction of the axes
scalebypanelist	scale the data by panelist before the construction of the axes
method	the method to replace the missing values: "average" or "coeff" (coefficients of the *product* variable in the anova model)

Details

The input data set is an object of class data.frame, for which the two first columns are qualitative variables (the first variable refers to the *panelist* variable and the second to the *product* variable) and the others are quantitative.

The ouptut of this function is a list with one element when performing PCA and two elements when performing MFA. The first element is the data frame of the coordinates of the products according to the whole panel (Panelist=0) and to the panelists. The second element is the data frame of the coordinates of the "partial products" according to the whole panel (Panelist=0) and to the panelists.

This function is necessary when calculating confidence ellipses for products.

Value

A list containing the following elements:

eig	a matrix with the component of the factor analysis (in row) and the eigenvalues, the inertia and the cumulative inertia for each component
moyen	the coordinates of the products with respect to the panel and to each panelists
partiel	the coordinates of the *partial* products with respect to the panel and to each panelists

Returns also a correlation circle as well as a graph of individuals

Author(s)

François Husson

References

Escofier, B. and Pagès, J. (1990) *Analyses factorielles simples et multiples: objectifs, méthodes et interprétation* Dunod, Paris. 1–267.

Escofier, B. and Pagès, J. (1994) Multiple factor analysis (AFMULT package). *Computational Statistics and Data Analysis*, **18**, 121–140.

See Also

MFA

Examples

```
## Example1: PCA
data(chocolates)
donnee <- cbind.data.frame(sensochoc[,c(1,4,5:18)])
axe <- construct.axes(donnee, scale.unit = TRUE)

## Example2: MFA (two groups of variables)
data(chocolates)
donnee <- cbind.data.frame(sensochoc[,c(1,4,5:18)])
axe <- construct.axes(donnee, group = c(6,8),
    name.group = c("A-F","T-S"),scale.unit = TRUE)
```

---

cpa *Consumers' Preferences Analysis*

Description

Performs preference mapping techniques based on multidimensional exploratory data analysis. This methodology is oriented towards consumers' preferences; here consumers are pictured according only to their preferences. In this manner, the distance between two consumers is very natural and easy to interpret, and a clustering of the consumers is also very easy to obtain.

Usage

```
cpa(senso, hedo, coord=c(1,2), center = TRUE, scale = TRUE,
    nb.clusters = 0, scale.unit = FALSE, name.panelist = TRUE,
    col = terrain.colors(45)[1:41])
```

Arguments

`senso`	a data frame of dimension (*p,k*), where *p* is the number of products and *k* the number of sensory descriptors
`hedo`	a data frame of dimension (*p,j*), where *p* is the number of products and *j* the number of consumers or panelists
`coord`	a length 2 vector specifying the components to plot

center	boolean, if TRUE then data are mean centered
scale	boolean, if TRUE then data are scaled to unit variance
nb.clusters	number of clusters to use (by default, 0 and the optimal numer of clusters is calculated
scale.unit	boolean, if TRUE then PCA is made on scaled data
name.panelist	boolean, if TRUE then the name of the panelist is written
col	color palette

Details

This methodology is oriented towards consumers' preferences; here, consumers are pictured according only to their preferences. In this manner, the distance between two consumers is very natural and easy to interpret, and a clustering of the consumers is also very easy to obtain using a classic hierarchical clustering procedure performed on Euclidian distances with the Ward's minimum variance criterion. The originality of the representation is that the characteristics of the products are also superimposed to the former picture.

Value

Return the following results:

clusters	the cluster number allocated to each consumer
result	the coordinates of the panelists, of the clusters, of the archetypes
prod.clusters	a list with as many elements as there are clusters; each element of the list gathers the specific products for its corresponding cluster
des.clusters	the correlation coefficients between the average hedonic scores per cluster and the sensory descriptors

A dendogram which highlight the clustering, a correlation circle that displays the hedonic scores, a graph of the consumers such as two consumers are all the more close that they do like the same products, as many graphs as there are variables: for a given variable, each consumer is colored according to the coefficient of correlation based on his hedonic scores and the variable.

Author(s)

François Husson ⟨François.Husson@agrocampus-rennes.fr⟩
Sébastien Lê ⟨Sebastien.Le@agrocampus-rennes.fr⟩

References

S. Lê, F. Husson, J. Pagès (2005). Another look at sensory data: how to "have your salmon and eat it, too!". *6th Pangborn sensory science symposium, August 7-11, 2005, Harrogate, UK.*

Examples

```
## Not run:
data(cocktail)
res.cpa = cpa(cbind(compo.cocktail, senso.cocktail), hedo.cocktail)
## If you prefer a graph in black and white and with 3 clusters
res.cpa = cpa(cbind(compo.cocktail, senso.cocktail), hedo.cocktail,
    name.panelist = TRUE, col = gray((50:1)/50), nb.clusters = 3)
## End(Not run)
```

---

decat *DEscription of CATegories*

Description

This function is designed to point out the variables that are the most characteristic according to the set of products in its whole, and to each of the products in particular.

This function is designed to test the main effect of a categorical variable (F-test) and the significance of its coefficients (T-test) for a set of endogenous variables and a given analysis of variance model. In most cases, the main effect is the product effect and the endogenous variables are the sensory descriptors.

Usage

```
decat(donnee, formul, firstvar, lastvar = length(colnames(donnee)),
    proba = 0.05, graph = TRUE, col.lower = "mistyrose",
    col.upper = "lightblue", nbrow = NULL, nbcol = NULL, random = TRUE)
```

Arguments

donnee	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
formul	the model that is to be tested
firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee)
proba	the significance threshold considered for the analyses of variance (by default 0.05)
graph	a boolean, if TRUE a barplot of the P-values associated with the F-test of the product effet is displayed
col.lower	the color used for 'level.lower'. Only useful if graph is TRUE
col.upper	the color used for 'upper.lower'. Only useful if graph is TRUE
nbrow	the number of rows to be displayed (by default, all the values are displayed). Only useful if graph is TRUE
nbcol	the number of columns to be displayed (by default, all the values are displayed). Only useful if graph is TRUE
random	boolean, effect should be possible as fixed or random (default as random)

Details

The `formul` parameter must be filled in by an analysis of variance model and must begin with the categorical variable of interest (e.g. the product effect) followed by the different other factors of interest (and their combinations). E.g.:`formul = "~Product+Panelist+Session"`.

Value

A list containing the following elements:

`tabF`	the V-test and the P-value of the F-test for each descriptor resulting from the analysis of variance model
`tabT`	a (products,descriptors) data frame, in which each cell is the Vtest for a given product and a given descriptor
`coeff`	a (products,descriptors) data frame, in which each cell is the coefficient resulting from the analysis of variance model for a given product and a given descriptor
`resF`	the V-test and the P-value for each descriptor resulting from the analysis of variance model, sorted in ascending order
`resT`	a list which elements are data frames, one data frame per product: the coefficient, the P-value and the Vtest for each significant descriptor resulting from the analysis of variance model, sorted in descending order

A barplot of the P-values associated with the F-test of the product effet.
A colored table with the adjusted means of the categorical variable: the values significantly different from the general mean are colored (singnificantly different with the `proba` level); the significantly less are colored in red (by default) and the significantly great are colored in blue.

Author(s)

François Husson

References

P. Lea, T. Naes, M. Rodbotten. *Analysis of variance for sensory data*.
H. Sahai, M. I. Ageel. *The analysis of variance*.

See Also

`aov`

Examples

```
### Example 1
data(chocolates)
## model (AOV): " descriptor = product + panelist "
resdecat<-decat(sensochoc, formul="~Product+Panelist", firstvar = 5)
barrow(resdecat$tabT)
barrow(t(resdecat$tabT), numr = 3, numc = 3)
barrow(resdecat$coeff, color = "orange")
```

```
### Example 2
data(chocolates)
## model (AOV): " descriptor = product + panelist "
res2 <-decat(sensochoc, formul="~Product+Panelist", firstvar = 5,
    proba=1, graph = FALSE)
```

`fast` *Factorial Approach for Sorting Task data*

Description

Perform Factorial Approach for Sorting Task data (FAST) on a table where the rows (i) are products and the columns (j) are consumers. A cell (i,j) corresponds either to the number of the group to which the product i belongs to for the consumer j, or, in the case of "qualified" categorization, to the sequence of words associted to the group of which the product belongs to (i) for the consumer j.

Usage

```
fast(don,alpha=0.05,mot_min=2,graph=TRUE,ncp=5,B=200)
```

Arguments

`don`	a data frame with n rows (products) and p columns (assesor : categorical variables)
`alpha`	the confidence level of the ellipses
`mot_min`	minimum sample size for the word selection in textual analysis
`graph`	boolean, if TRUE a graph is displayed
`ncp`	number of dimensions kept in the results (by default 5)
`B`	the number of simulations (corresponding to the number of virtual panels) used to compute the ellipses

Value

A list containing the following elements:

`eig`	a matrix containing all the eigenvalues, the percentage of variance and the cumulative percentage of variance
`var`	a list of matrices containing all the results for the categories (coordinates, square cosine, contributions, v.test)
`ind`	a list of matrices containing all the results for the products (coordinates, square cosine, contributions)
`group`	a list of matrices containing all the results for consumers (coordinates, square cosine, contributions)
`cooccur`	the reordered co-occurrence matrix among products
`reord`	the reordered matrix products*consumers
`cramer`	the Cramer's V matrix between all the consumers
`call`	a list with some statistics

Author(s)

Marine Cadoret, Sébastien L\textasciicircume ⟨sebastien.le@agrocampus-ouest.fr⟩

References

Cadoret, M., L\textasciicircume, S., Pagès, J. (2008) *A novel Factorial Approach for analysing Sorting Task data*. 9th Sensometrics meeting. St Catharines, Canada

Examples

```
## Not run:
data(perfume)
## Example of fast results
res.fast<-fast(perfume)
## End(Not run)
```

---

graphinter *Graphical display of the interaction between two qualitative variables*

Description

This function is designed to display the interaction between two qualitative variables, in most cases the *product* and the *session* variables.

Usage

```
graphinter(donnee, col.p, col.j, firstvar, lastvar=ncol(donnee),
    numr = 2,numc = 2)
```

Arguments

donnee	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
col.p	the position of one categorical variables of interest (the *product* variable)
col.j	the position of one categorical variables of interest (the *session* variable)
firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee)
numr	the number of graphs per row (by default 2)
numc	the number of graphs per column (by default 2)

Details

The data set must be balanced (or not unbalanced too much).

Value

If the variables of interest are the *product* and the *session* variables, a list containing the following components:

`prod`	a data frame of dimension (*p,q*), the means over the panelists and the sessions for the *p* products and the *q* sensory descriptors
`seance`	as many matrices of dimension (*p,q*) as there are sessions, the means over the panelists for the *p* products, the *q* sensory descriptors and for each session

The graphical display of the interaction for each sensory descriptor.

Author(s)

François Husson, Sébastien Lê

References

P. Lea, T. Naes, M. Rodbotten. *Analysis of variance for sensory data*.
H. Sahai, M. I. Ageel. *The analysis of variance*.

See Also

`aov`

Examples

```
## Not run:
data(chocolates)
graphinter(sensochoc, col.p = 4, col.j = 2, firstvar = 5, lastvar = 12,
    numr = 1, numc = 1)
## End(Not run)
```

---

hedo.cocktail *Cocktails hedonic scores*

Description

The data used here refer to 16 cocktails. Each cocktail was evaluated on a structured scale from 0 to 10, by 100 consumers, according to their liking (0) or disliking (10).

Usage

```
data(cocktail)
```

Format

A data frame with 16 rows and 100 columns: each row corresponds to a cocktail and each column to the hedonic scores given by one of the 100 consumers participating in the study.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(cocktail)
```

`hedochoc` *Chocolates hedonic scores*

Description

The data used here refer to six varieties of chocolates sold in France. Each chocolate was evaluated on a structured scale from 0 to 10, by 222 consumers, according to their liking (0) or disliking (10).

Usage

```
data(chocolates)
```

Format

A data frame with 6 rows and 222 columns: each row corresponds to a chocolate and each column to the hedonic scores given by one of the 222 consumers participating in the study.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(chocolates)
```

`histprod` *Histogram for each descriptor*

Description

Computes automatically histograms for a set of quantitative variables.

Usage

```
histprod(donnee, firstvar, lastvar = ncol(donnee), numr = 2,
    numc = 2, adjust = 1)
```

Arguments

donnee	a data frame
firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee)
numr	the number of histograms per row (by default 2)
numc	the number of histograms per column (by default 2)
adjust	the bandwidth used is actually 'adjust*bw'. This makes it easy to specify values like "half the default" bandwidth.

Details

Displays histograms with a common Y-axis as well as the local estimator of the density for each descriptor, hence the adjust parameter to fill in. Displays also the normal distribution with mean and variance the respective values estimated for each descriptor.

Author(s)

Sébastien Lê

See Also

density, hist

Examples

```
data(chocolates)
histprod(sensochoc, firstvar = 5, lastvar = 10)
```

---

indscal *Construct the Indscal model for Napping data type*

Description

This version of the Indscal model is specially adapted to Napping data type, i.e. products (stimuli) are positioned on a tableclothe by panelists, then their coordinates are used as input for the Indscal model.

Usage

```
indscal(matrice, matrice.illu = NULL, maxit = 200, coord = c(1,2),
    eps = 1/10^5)
```

Arguments

matrice	a data frame of dimension (*p,2j*), where *p* represents the number of products and *j* the number of panelists (two coordinates per panelist)
matrice.illu	a data frame with illustrative variables (with the same row.names in common as in matrice)
maxit	the maximum number of iterations until the algorithm stops
coord	a length 2 vector specifying the components to plot
eps	a threshold with respect to which the algorithm stops, i.e. when the difference between the criterion function at step *n* and *n+1* is less than eps

Value

Returns a list including:

W	a matrix with the subject coordinates
points	a matrix with the stimuli (individuals) coordinates
subvar	a vector with the strain between each configuration and the stimuli configuration
r2	the strain criterion

The functions returns the three following graphs:
A stimuli representation, ie. a representation of the products
A representation of the weights computed by the Indscal model.
A correlation circle of the variables enhanced by illustrative variables (supplementary columns)

Author(s)

Peter Ellis
François Husson

References

Carroll, J.D. & J.J. Chang (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35, 283-319.

See Also

nappeplot, pmfa

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
resindscal<- indscal(napping.don, napping.words)
x11()
prefpls(cbind(resindscal$points, napping.words))
x11()
pmfa(napping.don, napping.words, mean.conf = resindscal$points)
## End(Not run)
```

`interact` *Estimation of interaction coefficients*

Description

Computes automatically the interaction coefficients between two quantitative variables `col.p` and `col.j` for the following model: `"~col.p+col.j+col.p:col.j"`.

Usage

```
interact(donnee, col.p, col.j, firstvar, lastvar = ncol(donnee))
```

Arguments

`donnee`	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
`col.p`	the position of the *product* effect for instance
`col.j`	the position of the *panelist* effect for instance
`firstvar`	the position of the first endogenous variable
`lastvar`	the position of the last endogenous variable (by default the last column of `donnee`)

Details

In most cases `col.p` represents the *product* effect, `col.j` represents the *panelist* effect, and the variables of interest are the sensory descriptors. The model considered is the following one: `"~Product+Panelist+Product:Panelist"`.
Data must be complete (but not necessarily balanced).

Value

Returns an array of dimension (*p,j,k*), where *p* is the number of products, *j* the number of panelists and *k* the number of sensory descriptors. The entries of this array are the interaction coefficients between a panelist and a product for a given descriptor.
For each sensory descriptor, returns a graph where each (panelist,product) interaction coefficient is displayed, a graph where the contribution to the (panelist,product) interaction coefficient by product is displayed, a graph where the contribution to the (panelist,product) interaction coefficient by panelist is displayed.

Author(s)

François Husson

See Also

`aov`

Examples

```
## Not run:
data(chocolates)
resinteract=interact(sensochoc, col.p = 4, col.j = 1, firstvar = 5)
## End(Not run)
```

`magicsort` *Returns a sorted data matrix*

Description

Sort the rows and columns of a matrix in a "magic" order or by ascending (or descending) mean or median or geometrical mean.

Usage

```
magicsort(matrice, sort.mat = matrice, method = "magic",
    byrow = TRUE, bycol = TRUE, ascending = TRUE)
```

Arguments

`matrice`	a data matrix to sort
`sort.mat`	sort the rows and columns according to the result of the PCA made on this matrix (by default the `matrice`)
`method`	four types of calculations, magic (`"magic"`), (`"median"`), arithmetical (`"mean"`) or geometrical (`"geo"`) mean (by default magic)
`byrow`	boolean, if TRUE then data are sorted over the rows
`bycol`	boolean, if TRUE then data are sorted over the columns
`ascending`	boolean, if TRUE then data are sorted in ascending order

Details

Very useful function to compare results.

Author(s)

François Husson, Sébastien Lê

Examples

```
## Example 1
data(chocolates)
resdecat<-decat(sensochoc, formul = "~Product", firstvar = 5,
    graph = FALSE)
coltable(magicsort(resdecat$tabT), level.lower = -1.96,
    level.upper = 1.96, main.title = "Products' description")
```

```
## Example 2
data(chocolates)
resperf<-paneliperf(sensochoc,
    formul = "~Product+Panelist+Product:Panelist",
    formul.j = "~Product", col.j = 1, firstvar = 5, lastvar = 12,
    synthesis = FALSE, graph = FALSE)
res.sort=magicsort(resperf$prob.ind, method = "median")
coltable(res.sort, main.title = "P-values of the F-test by panelist")
```

`nappeplot` *Plot panelists' tableclothe*

Description

Plot panelists' tableclothe.

Usage

```
nappeplot(donnee, numr = 2, numc = 2, color = "blue", lim = c(60,40))
```

Arguments

`donnee`	a data frame of dimension (*p,2j*), where *p* represents the number of products and *j* the number of panelists
`numr`	the number of tableclothe per row (by default 2)
`numc`	the number of tableclothe per column (by default 2)
`color`	the color used to display the products
`lim`	the size of the tableclothe

Details

The data used here refer to a specific experiment, where panelists are asked to position products on a tableclothe of dimension `lim`, by default (60,40).

Value

Returns as many graphs as there are panelists, each graph represents products positioned by a given panelist on a tablecloth

Author(s)

François Husson

References

Pagès J. (2005). Collection and analysis of perceived product inter-distances using multiple factor analysis; application to the study of ten white wines from the Loire Valley. *Food Quality and Preference*. 16 (7) pp. 642-649.

See Also

napping, pmfa, indscal

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
## End(Not run)
```

---

napping *Napping data*

Description

The data used here refer to 10 different French wines evaluated by 11 panelists.
They were asked to position the wines on a tableclothe of dimension (60,40). They were asked to describe each wine using their own word list.

Usage

```
data(napping)
```

Format

There are two data frames: - napping.don: A data frame of dimension (10,22): each row represents a French wine, each couple (Xi,Yi) represents the coordinates of the wines positioned on a tableclothe for a given panelist;
- napping.words: A data frame of dimension (10,14): each row represents a French wine, each column an attribute, each cell the number of times a given attribute was quoted for a given wine.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
x11()
pmfa(napping.don, napping.words)
## End(Not run)
```

napping.don *An example of Napping data*

Description

The data used here refer to 10 different French wines evaluated by 11 panelists. They were asked to position the wines on a tableclothe of dimension (60,40).

Usage

```
data(napping)
```

Format

A data frame of dimension (10,22): each row represents a French wine, each couple (Xi,Yi) represents the coordinates of the wines positioned on a tableclothe for a given panelist.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
## End(Not run)
```

napping.words *An example of "illustrative" variables to enhance results from Napping data*

Description

The data used here refer to 10 different French wines evaluated by 11 panelists. They were asked to describe each wine using their own word list.

Usage

```
data(napping)
```

Format

A data frame of dimension (10,14): each row represents a French wine, each column an attribute, each cell the number of times a given attribute was quoted for a given wine

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
x11()
pmfa(napping.don, napping.words)
## End(Not run)
```

---

optimaldesign *Construction of an optimal design*

Description

Construction of an optimal design balanced for first order of carry-over effect.

Usage

```
optimaldesign(nbPanelist, nbProd, nbProdByPanelist, nbPanelistMin = nbPanelist,  or
```

Arguments

nbPanelist	Maximum number of panelists
nbProd	Number of products
nbProdByPanelist	Number of products that each panelist will evaluate
nbPanelistMin	Minimum number of panelists who will evaluate the products
ordre	Boolean, if TRUE the order of presentation of the product to the panelist is given
weight	Importance of the rank and of the carry-over effect. From 0 to 1, if 0 the design will only take into account the carry-over effect (and not the rank effect).
graine	initialization of the algorithm
nbDesignProd	Number of iteration of the algorithm to affect the products to the panelists
nbDesignOrdre	Number of iteration of the algorithm for the rank of presentation
matEssImp	Matrix of the imposed experiments

Value

List with

design	Design with the products evaluated by each panelist
rank	Design with the products evaluated by each panelist and with the rank

Author(s)

E. Périnel, O. Tran, J. Mazet

References

Périnel E. & Pagès J. (2003). Optimal nested cross-over designs in sensory analysis. *Food Quality and Preference*. 15 (5). pp. 439-446.

Examples

```
## Not run:
optimaldesign(nbPanelist=10,nbPanelistMin=8,nbProd=5,nbProdByPanelist=3)
## End(Not run)
```

---

paneliperf — *Panelists' performance according to their capabilities to dicriminate between products*

Description

Computes automatically P-values, Vtests, residuals, r-square for each category of a given qualitative variable (e.g. the *panelist* variable);
Computes he agreement between each panelist and the panel results;
Gives the panel results (optional).

Usage

```
paneliperf(donnee, formul, formul.j = "~Product", col.j, firstvar,
      lastvar = ncol(donnee), synthesis = FALSE, random = TRUE,
      graph = FALSE)
```

Arguments

donnee	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
formul	the aov model used for the panel
formul.j	the aov model used for each panelist (no *panelist* effect allowed)
col.j	the position of the *panelist* variable
firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee
synthesis	boolean, the possibility to have the anova results for the panel model
random	boolean, the status of the Panelist variable in the anova model for the panel
graph	boolean, draws the PCA and MFA graphs

Details

The `formul` parameter must be filled in by an analysis of variance model and must begin with the categorical variable of interest (e.g. the product effect) followed by the different other factors of interest (and their combinations). E.g.:`formul = "~Product+Session"`.

Value

A list containing the following components:

`prob.ind`	a matrix which rows are the panelist, which columns are the endogenous variables (in most cases the sensory descriptors) and which entries are the P-values associated to the AOV model
`vtest.ind`	a matrix which rows are the panelist, which columns are the endogenous variables (in most cases the sensory descriptors) and which entries are the Vtests associated to the AOV model
`res.ind`	a matrix which rows are the panelist, which columns are the endogenous variables (in most cases the sensory descriptors) and which entries are the residuals associated to the AOV model
`r2.ind`	a matrix which rows are the panelist, which columns are the endogenous variables (in most cases the sensory descriptors) and which entries are the R-square associated to the AOV model
`signif.ind`	a vector with the number of significant descriptors per panelist
`agree.ind`	a matrix with as many rows as there are panelists and as many columns as there are descriptors and the entries of this matrix are the correlation coefficients between the product coefficients for the panel and for the panelists
`complete`	a matrix with the v-test corresponding to the p.value (see `p.values` below), the median of the agreement (see `agree` upper), the standard deviation of the panel anova model (see `res` below)
`p.value`	a matrix of dimension (*k,m*) of *P-values* associated with the F-test for the *k* descriptors and the *m* factors and their combinations considered in the analysis of variance model of interest
`variability`	a matrix of dimension (*k,m*) where the entries correspond to the percentages of variability due to the effects introduced in the analysis of variance model of interest
`res`	a vector of dimension *k* of residual terms for the analysis of variance model of interest
`r2`	a vector of dimension *k* of r-squared for the analysis of variance model of interest

The usual graphs when MFA is performed on the data.frame resulting from vtest.ind and agree.ind. The PCA graphs for the complete output.

Author(s)

François Husson, Sébastien Lê

References

P. Lea, T. Naes, M. Rodbotten. *Analysis of variance for sensory data*. H. Sahai, M. I. Ageel. *The analysis of variance*.

See Also

panelperf, aov

Examples

```
## Not run:
data(chocolates)
res<-paneliperf(sensochoc, formul = "~Product+Panelist+Session+
  Product:Panelist+Product:Session+Panelist:Session",
  formul.j = "~Product", col.j = 1, firstvar = 5, synthesis = TRUE)
resprob<-magicsort(res$prob.ind, method = "median")
coltable(resprob, level.lower = 0.05, level.upper = 1,
    main.title = "P-value of the F-test (by panelist)")
hist(resprob,main="Histogram of the P-values",xlab="P-values")

resr2<-magicsort(res$r2.ind, method = "median", ascending = FALSE)
coltable(resr2, level.lower = 0.00, level.upper = 0.85,
    main.title = "Adjusted R-square (by panelist)")

resagree<-magicsort(res$agree, sort.mat = res$r2.ind, method = "median")
coltable(resagree, level.lower = 0.00, level.upper = 0.85,
    main.title = "Agreement between panelists")
hist(resagree,main="Histogram of the agreement between panelist and panel",
    xlab="Correlation coefficient between the product effect for
    panelist and panel")

coltable(magicsort(res$p.value, sort.mat = res$p.value[,1], bycol = FALSE,
    method = "median"),
    main.title = "Panel performance (sorted by product P-value)")
## End(Not run)
```

---

panellipse *Confidence ellipses around products based on panelists descriptions*

Description

Virtual panels are generated using Boostrap techniques in order to display confidence ellipses around products.

Usage

```
panellipse(donnee, col.p, col.j, firstvar, lastvar = ncol(donnee),
    alpha = 0.05, coord = c(1,2), scale.unit = TRUE, nbsimul = 500,
    nbchoix = NULL, group = NULL, name.group = NULL,
```

```
level.search.desc = 0.2, centerbypanelist = TRUE,
scalebypanelist = FALSE, name.panelist = FALSE,
variability.variable = TRUE, cex = 1, color = NULL)
```

Arguments

donnee	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
col.p	the position of the *product* variable
col.j	the position of the *panelist* variable
firstvar	the position of the first sensory descriptor
lastvar	the position of the last sensory descriptor (by default the last column of donnee)
alpha	the confidence level of the ellipses
coord	a length 2 vector specifying the components to plot
scale.unit	boolean, if T the descriptors are scaled to unit variance
nbsimul	the number of simulations (corresponding to the number of virtual panels) used to compute the ellipses
nbchoix	the number of panelists forming a virtual panel, by default the number of panelists in the original panel
group	the number of variables in each group of variables when multiple factor analysis is performed (by default this parameter equals NULL and a PCA is performed)
name.group	the names of the groups of variables when mfa is performed (if group differs from NULL)
level.search.desc	the threshold above which a descriptor is not considered as discriminant according to AOV model "descriptor=Product+Panelist"
centerbypanelist	boolean, if T center the data by panelist before the construction of the axes
scalebypanelist	boolean, if T scale the data by panelist before the construction of the axes (by default, FALSE is assigned to that parameter)
name.panelist	boolean, if T then the name of each panelist is displayed on the plotpanelist graph (by default, FALSE is assigned to that parameter)
variability.variable	boolean, if T a plot with the variability of the variable is drawn and a confidence intervals of the correlations between descriptors are calculated
cex	cf. function par in the **graphics** package
color	a vector with the colors used; by default there are 35 colors defined

Details

Panellipse, step by step:
Step 1 Performs a selection of discriminating descriptors with respect to a threshold set by users
Step 2 Virtual panels are generated using Boostrap techniques; the number of panels as well as their size are set by users with the *nbsimul* and *nbchoix* parameters
Step 3 Coordinates of the products with respect to each virtual panels are computed
Step 4 Each product is then circled by its confidence ellipse generated by virtual panels and comprising (1-alpha)*100 percent of the virtual products
Step 5 Variability of the variables is drawn and confidence interval of the correlation coefficient between descriptors are calculated by bootstrap

Value

A list containing the following elements:

`eig`	a matrix with the component of the factor analysis (in row) and the eigenvalues, the inertia and the cumulative inertia for each component
`coordinates`	a list with: the coordinates of the products with respect to the panel and to each panelists and the coordinates of the *partial* products with respect to the panel and to each panelists
`hotelling`	Returns a matrix with the P-values of the Hotelling's T2 tests for each pair of products: this matrix allows to find the product which are significatnly different for the 2-components sensory description; if an MFA is done, *hotelling* returns as many matrices as there are group, these matrices allows to find the product which are significantly different for the 2-components sensory description of the group, and it returns also a *global* matrix corresponding to the P-values for the tests corresponding to the mean product.

Returns a graph of the products as well as a correlation circle of the descriptors.

Returns a graph where each product is displayed with respect to a panel and to each panelist composing the panel; products described by the panel are displayed as square, they are displayed as circle when they are described by each panelist.

Returns a graph where each product is circled by its confidence ellipse generated by virtual panels. When a Multiple Factor Analysis is performed, returns a graph where each partial product is circled by its confidence ellipse generated by virtual panels.

Returns a graph where the variability of each variable is drawn on the correlation circle graph.

Author(s)

François Husson

References

Husson F., Le Dien S. & Pagès J. (2005). Confidence ellipse for the sensory profiles obtained by Principal Components Analysis. *Food Quality and Preference*. 16 (3), 245-250.
Pagès J. & Husson F. (2005). Multiple Factor Analysis with confidence ellipses: a methodology to study the relationships between sensory and instrumental data. To be published in *Journal of Chemometrics*.
Husson F., Lê S. & Pagès J. Variability of the representation of the variables resulting from PCA in the case of a conventional sensory profile. *Food Quality and Preference*. 16 (3), 245-250.

See Also

`panellipse.session`, `panelmatch`

Examples

```
## Not run:
## Example 1: PCA
data(chocolates)
res <- panellipse(sensochoc, col.p = 4, col.j = 1, firstvar = 5)
coltable(res$hotelling, main.title = "P-values for the Hotelling's T2 tests")

## If we consider only 12 panelists in a virtual panel,
## what would be the size of the ellipses
res2 <- panellipse(sensochoc, col.p = 4, col.j = 1, nbchoix = 12, firstvar = 5)
coltable(res2$hotelling, main.title = "P-values for the Hotelling's T2 tests")

## If we want the confidence ellipses around the individual descriptions
panellipse(sensochoc, col.p = 4, col.j = 1, nbchoix = 1, firstvar = 5)

## Example 2: MFA
data(chocolates)
res <- panellipse(sensochoc, col.p = 4, col.j = 1, firstvar = 5,
    group = c(6,8), name.group = c("G1","G2"))
for (i in 1:dim(res$hotelling$bygroup)[3]) coltable(res$hotelling$bygroup[,,i],
    main.title = paste("P-values for the Hotelling's T2 tests (",
    dimnames(res$hotelling$bygroup)[3][[1]][i],")",sep=""))
## End(Not run)
```

---

panellipse.session *Repetability of panelists descriptions studied by confidence ellipses around products per session*

Description

Virtual panels are generated using Boostrap techniques in order to display confidence ellipses around products.

Usage

```
panellipse.session(donnee, col.p, col.j, col.s, firstvar,
    lastvar = ncol(donnee), alpha = 0.05, coord = c(1,2),
    scale.unit = TRUE, nbsimul = 500, nbchoix = NULL,
    level.search.desc = 0.2,  centerbypanelist = TRUE,
    scalebypanelist = FALSE, name.panelist = FALSE,
    variability.variable = FALSE, cex = 1, color= NULL)
```

Arguments

donnee	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
col.p	the position of the *product* variable
col.j	the position of the *panelist* variable
col.s	the position of the *session* variable
firstvar	the position of the first sensory descriptor
lastvar	the position of the last sensory descriptor (by default the last column of donnee)
alpha	the confidence level of the ellipses
coord	a length 2 vector specifying the components to plot
scale.unit	boolean, if T the descriptors are scaled to unit variance
nbsimul	the number of simulations (corresponding to the number of virtual panels) used to compute the ellipses
nbchoix	the number of panelists forming a virtual panel, by default the number of panelists in the original panel
level.search.desc	the threshold above which a descriptor is not considered as discriminant according to AOV model "descriptor=Product+Panelist"
centerbypanelist	boolean, if T center the data by panelist before the construction of the axes
scalebypanelist	boolean, if T scale the data by panelist before the construction of the axes (by default, FALSE is assigned to that parameter)
name.panelist	boolean, if T then the name of each panelist is displayed on the plotpanelist graph (by default, FALSE is assigned to that parameter)
variability.variable	boolean, if T a plot with the variability of the variable is drawn and a confidence intervals of the correlations between descriptors are calculated
cex	cf. function par in the **graphics** package
color	a vector with the colors used; by default there are 35 colors defined

Details

panellipse.session, step by step:
Step 1 Construct a data frame by session
Step 2 Performs a selection of discriminating descriptors with respect to a threshold set by users
Step 3 MFA is computed with one group for one session
Step 4 Virtual panels are generated using Boostrap techniques; the number of panels as well as their size are set by users with the *nbsimul* and *nbchoix* parameters
Step 5 Coordinates of the products with respect to each virtual panels are computed
Step 6 Each product is then circled by its confidence ellipse generated by virtual panels and comprising (1-alpha)*100 percent of the virtual products

Value

A list containing the following elements:

`bysession`	the data by session
`eig`	a matrix with the component of the factor analysis (in row) and the eigenvalues, the inertia and the cumulative inertia for each component
`coordinates`	a list with: the coordinates of the products with respect to the panel and to each panelists and the coordinates of the *partial* products with respect to the panel and to each panelists
`hotelling`	returns a matrix with the P-values of the Hotelling's T2 tests for each pair of products: this matrix allows to find the product which are significatnly different for the 2-components sensory description
`variability`	returns an index of the sessions' reproductibility: the first eigenvalue of the separate PCA performed on homologous descriptors

Returns a graph of the products as well as a correlation circle of the descriptors.

Returns a graph where each product is displayed with respect to a panel and to each panelist composing the panel; products described by the panel are displayed as square, they are displayed as circle when they are described by each panelist.

Returns a graph where each product is circled by its confidence ellipse generated by virtual panels.

Returns a graph where each partial product is circled by its confidence ellipse generated by virtual panels.

Returns a graph where the variability of each variable is drawn on the correlation circle graph.

Author(s)

François Husson, Sébastien Lê

References

Husson F., Le Dien S. & Pagès J. (2005). Confidence ellipse for the sensory profiles obtained by Principal Components Analysis. *Food Quality and Preference*. 16 (3), 245-250.
Pagès J. & Husson F. (2005). Multiple Factor Analysis with confidence ellipses: a methodology to study the relationships between sensory and instrumental data. To be published in *Journal of Chemometrics*.
Husson F., Lê S. & Pagès J. Variability of the representation of the variables resulting from PCA in the case of a conventional sensory profile. *Food Quality and Preference*. 16 (3), 245-250.

See Also

panellipse

Examples

```
data(chocolates)
res <- panellipse.session(sensochoc, col.p = 4, col.j = 1, col.s = 2,
    firstvar = 5)
magicsort(res$variability)
for (i in 1:dim(res$hotelling$bysession)[3]) coltable(res$hotelling$bysession[,,i],
    main.title = paste("P-values for the Hotelling's T2 tests (",
    dimnames(res$hotelling$bysession)[3][[1]][i],")",sep=""))
```

---

panelmatch *Confidence ellipses around products based on panel descriptions*

Description

Comparison of panels.

Usage

```
panelmatch(donnee, col.p, col.j, firstvar,
    alpha = 0.05, coord = c(1,2), scale.unit = TRUE, nbsimul = 500,
    nbchoix = NULL, centerbypanelist = TRUE,
    scalebypanelist = FALSE, name.panelist = FALSE, cex = 1,
    color = NULL, hierar = NULL)
```

Arguments

donnee	a list of data frames, each one made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
col.p	the position of the *product* variable (in each data frame, the same position)
col.j	the position of the *panelist* variable (in each data frame, the same position)
firstvar	the position of the first sensory descriptor (in each data frame, the same position)
alpha	the confidence level of the ellipses

coord	a length 2 vector specifying the components to plot
scale.unit	boolean, if T the descriptors are scaled to unit variance
nbsimul	the number of simulations (corresponding to the number of virtual panels) used to compute the ellipses
nbchoix	the number of panelists forming a virtual panel, by default the number of panelists in the original panel
centerbypanelist	boolean, if T center the data by panelist before the construction of the axes
scalebypanelist	boolean, if T scale the data by panelist before the construction of the axes (by default, FALSE is assigned to that parameter)
name.panelist	boolean, if T then the name of each panelist is displayed on the plotpanelist graph (by default, FALSE is assigned to that parameter)
cex	cf. function par in the **graphics** package
color	a vector with the colors used; by default there are 35 colors defined
hierar	hierarchy in the variable (see hmfa)

Value

A list containing the following elements:

eig	a matrix with the component of the factor analysis (in row) and the eigenvalues, the inertia and the cumulative inertia for each component
coordinates	a list with: the coordinates of the products with respect to the panel and to each panelists and the coordinates of the *partial* products with respect to the panel and to each panelists
hotelling	Returns a matrix with the P-values of the Hotelling's T2 tests for each pair of products: this matrix allows to find the product which are significatnly different for the 2-components sensory description

Returns a graph of the products as well as a correlation circle of the descriptors.

Returns a graph where each product is displayed with respect to a panel and to each panelist composing the panel; products described by the panel are displayed as square, they are displayed as circle when they are described by each panelist.

Returns a graph where each product is circled by its confidence ellipse generated by virtual panels. When a Multiple Factor Analysis is performed, returns a graph where each partial product is circled by its confidence ellipse generated by virtual panels.

Author(s)

François Husson

References

Husson F., Le Dien S. & Pagès J. (2005). Confidence ellipse for the sensory profiles obtained by Principal Components Analysis. *Food Quality and Preference*. 16 (3), 245-250.
Pagès J. & Husson F. (2005). Multiple Factor Analysis with confidence ellipses: a methodology to study the relationships between sensory and instrumental data. To be published in *Journal of Chemometrics*.

See Also

panellipse, panellipse.session

Examples

```
## Not run:
data(chocolates)
Panel1= sensochoc[sensochoc[,1]<11,]
Panel2= sensochoc[(sensochoc[,1]<21)&(sensochoc[,1]>10),]
Panel3= sensochoc[sensochoc[,1]>20,]
res <- panelmatch(list(P1=Panel1,P2=Panel2,P3=Panel3), col.p = 4, col.j = 1, firstvar = 5)
## End(Not run)
```

---

panelperf *Panel's performance according to its capabilities to dicriminate between products*

Description

Computes automatically P-values associated with the F-test as well as the residual term for a given analysis of variance model.

Usage

```
panelperf(donnee, formul, subset = NULL, firstvar,
    lastvar = ncol(donnee), random = TRUE)
```

Arguments

donnee	a data frame
formul	the model that is to be tested
subset	cf. function lm in the **stats** package
firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee
random	boolean, effect should be possible as fixed or random (default as random)

Details

The `formul` parameter must be filled in by an analysis of variance model and must begin with the categorical variable of interest (e.g. the product effect) followed by the different other factors of interest (and their combinations). E.g.:`formul = "~Product+Session"`.

Value

A list containing the following components:

`p.value`	a matrix of dimension (*k,m*) of *P-values* associated with the F-test for the *k* descriptors and the *m* factors and their combinations considered in the analysis of variance model of interest
`variability`	a matrix of dimension (*k,m*) where the entries correspond to the percentages of variability due to the effects introduced in the analysis of variance model of interest
`res`	a vector of dimension *k* of residual terms for the analysis of variance model of interest
`r2`	a vector of dimension *k* of r-squared for the analysis of variance model of interest

Author(s)

François Husson, Sébastien Lê

References

P. Lea, T. Naes, M. Rodbotten. *Analysis of variance for sensory data*.
H. Sahai, M. I. Ageel. *The analysis of variance*.

See Also

`paneliperf`, `aov`

Examples

```
data(chocolates)
res=panelperf(sensochoc, firstvar = 5, formul = "~Product+Panelist+
    Session+Product:Panelist+Session:Product+Panelist:Session")
## Sort results by product p.values.
coltable(magicsort(res$p.value, sort.mat = res$p.value[,1], bycol = FALSE,
    method = "median"), main.title = "Panel performance (sorted by product P-value)")
```

perfume *Perfume*

Description

The data used here refer to 12 luxury perfumes categorized by 30 consumers.

Usage

```
data(perfume)
```

Format

A data frame with 12 rows (the number of perfumes) and 30 columns (the number of consumers): a cell corresponds either to the number of the group to which the product belongs to for the consumer, or, in the case of "qualified" categorization, to the sequence of words associted to the group of which the product belongs to for the consumer.

Source

Applied mathematics department, AGROCAMPUS

Examples

```
## Not run:
data(perfume)
res.fast = fast(perfume)
## End(Not run)
```

plot.fast *Make Factorial Approach for Sorting Task data (FAST) graphs*

Description

Plot the graphs for Factorial Approach for Sorting Task data (FAST).

Usage

```
plot.fast(x,choix="ind", axes = c(1, 2), xlim = NULL, ylim = NULL, invisible = NULL
    col.ind = "blue", col.var = "red", col.quali.sup = "darkred",
    col.ind.sup ="darkblue", col.quanti.sup = "black",label = "all",
    cex = 1,lab.grpe = TRUE, title = NULL, habillage = "none", palette = NULL,
    new.plot = TRUE, ...)
```

Arguments

x	an object of class fast
axes	a length 2 vector specifying the components to plot
choix	the graph to plot ("ind" for the products, "var" for the vcategories, "group" for the consumers)
xlim	range for the plotted 'x' values, defaulting to the range of the finite values of 'x'
ylim	range for the plotted 'y' values, defaulting to the range of the finite values of 'y'
habillage	give no color for the individuals ("none"), or color the products among a consumer (give the number of the consumer)
col.ind	a color for the products
col.var	a color for the categories
col.quali.sup	a color for the supplementary categories
col.ind.sup	a color for the supplementary individuals
col.quanti.sup	a color for the quantitative supplementary variables
label	a list of character for the elements which are labelled (by default, all the elements are labelled ("ind", "var"))
invisible	string indicating if some points should not be drawn ("ind" or "var")
cex	cf. function par in the **graphics** package
lab.grpe	boolean, if TRUE, the consumers are labelled
title	string corresponding to the title of the graph you draw (by default NULL and a title is chosen)
palette	the color palette used to draw the points. By default colors are chosen. If you want to define the colors : palette=palette(c("black","red","blue")); or you can use: palette=palette(rainbow(30)), or in black and white for example: palette=palette(gray(seq(0,.9,len=25
new.plot	boolean, if TRUE, a new graphical device is created
...	further arguments passed to or from other methods

Value

Returns the products factor map, the categories factor map and the consumers factor map.

Author(s)

Marine Cadoret, Sébastien L\textasciicircume ⟨sebastien.le@agrocampus-ouest.fr⟩

See Also

fast

Examples

```
## Not run:
data(perfume)
res.fast <- fast(perfume,graph=FALSE)
plot.fast(res.fast,choix="ind",invisible="var",habillage=5)
plot.fast(res.fast,choix="group")
## End(Not run)
```

`plotpanelist` *Plotpanelist*

Description

Displays panelists' sensory profiles onto the products' space

Usage

```
plotpanelist(mat, coord = c(1,2), name = FALSE, eig, cex = 1, color = NULL)
```

Arguments

`mat`	a data frame structured as the first element of the list resulting from the function construct.axes, i.e. the coordinates of the products with respect to the panel and to each panelists
`coord`	a length 2 vector specifying the components to plot
`name`	boolean, if T then the name of each panelist is displayed on the graph (by default, FALSE is assigned to that parameter)
`eig`	a matrix with the component of the factor analysis (in row) and the eigenvalues, the inertia and the cumulative inertia for each component. Typically, the `eig` output of the `construct.axes` function
`cex`	cf. function `par` in the **graphics** package
`color`	a vector with the colors used; by default there are 35 colors defined

Value

Returns a graph where each product is displayed with respect to a panel and to each panelist composing the panel. Products described by the panel are displayed as square, they are displayed as circle when they are described by each panelist.

Author(s)

François Husson

Examples

```
data(chocolates)
donnee <- cbind.data.frame(sensochoc[,c(1,4,5:18)])
axe <- construct.axes(donnee, scale.unit = TRUE)
plotpanelist(axe$moyen, eig = signif(axe$eig,4))
```

`pmfa` *Procrustean Multiple Factor Analysis (PMFA)*

Description

Performs Multiple Factor Analysis combined with Procrustean Analysis.

Usage

```
pmfa(matrice, matrice.illu = NULL, mean.conf = NULL, dilat = TRUE,
      graph.ind = TRUE, graph.mfa = TRUE, lim = c(60,40), coord = c(1,2), cex = 0.8
```

Arguments

`matrice`	a data frame of dimension (*p,2j*), where *p* represents the number of products and *j* the number of panelists
`matrice.illu`	a data frame with illustrative variables (with the same row.names in common as in `matrice`)
`mean.conf`	coordinates of the average configuration (by default NULL, the average configuration is generated by MFA)
`dilat`	boolean, if TRUE (which is the default value) the Morand's dilatation is used
`graph.ind`	boolean, if TRUE (which is the default value) superimposes each panelist's configuration on the average configuration
`graph.mfa`	boolean, if TRUE (which is the default value) and if `mean.conf` = NULL the graphs of the MFA are drawn
`lim`	size of the tablecothe
`coord`	a length 2 vector specifying the components to plot
`cex`	cf. function `par` in the **graphics** package

Details

Performs first Multiple Factor Analysis on the tableclothes, then GPA in order to superimpose as well as possible panelist's configuration on the average configuration obtained by MFA (in the case where `mean.conf` is NULL). If `mean.conf` is not NULL the configuration used is the one input by the user.

Value

Returns the RV coefficient between each individual configuration and the consensus.
If `mean.conf` is NULL (and `graph.mfa` is TRUE), returns the usual graphs resulting from the MFA function: the graph of the individuals and their partial representations, the graph of the variables (i.e. the coordinates of the products given by each panelist).
If `mean.conf` is not NULL returns the configuration input by the user.
When `matrice.illu` is not NULL, returns a graph of illustrative variables.
Returns as many superimposed representations of individual configurations as there are panelists.

Author(s)

François Husson, Sébastien Lê

References

Morand, E., Pagès, J. Procrustes multiple factor analysis to analyze the overall perception of food products. *Food Quality and Preference* 14, 182-188.

See Also

MFA, nappeplot, indscal

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
x11()
pmfa(napping.don, napping.words)
## End(Not run)
```

---

print.fast *Print Factorial Approach for Sorting Task data (FAST) results*

Description

Print Factorial Approach for Sorting Task data (FAST) results.

Usage

```
print.fast(x, file = NULL, sep = ";", ...)
```

Arguments

x	an object of class fast
file	A connection, or a character string naming the file to print to. If NULL (the default), the results are not printed in a file
sep	character string to insert between the objects to print (if the argument file is not NULL
...	further arguments passed to or from other methods

Author(s)

Marine Cadoret, Sébastien L\textasciicircume ⟨sebastien.le@agrocampus-ouest.fr⟩

See Also

fast

Examples

```
## Not run:
data(perfume)
res.fast <- fast(perfume,graph=FALSE)
print.fast(res.fast, file="c:/essai.csv", sep = ";")
## End(Not run)
```

---

scalebypanelist *Scale by panelist*

Description

Returns a data frame with entries the means of the products over the sessions for the whole panel and for each panelist.

Usage

```
scalebypanelist(matrice, center = TRUE, scale = FALSE, col.p,
    col.j, firstvar, lastvar = ncol(matrice), method = "coeff")
```

Arguments

matrice	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
center	boolean, if T scores given by panelists are centered
scale	boolean, if T scores given by panelists are scaled to unit varaince
col.p	the position of one categorical variables of interest (the *product* variable)
col.j	the position of one categorical variables of interest (the *panelist* variable)

firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee)
method	the method to replace the missing values: "average" or "coeff" (coefficients of the *product* variable in the anova model)

Value

Returns a data frame of dimension (*p*(1+j),k+2*), where *p* is the number of products, *j* the number of panelists, and *k* the number of sensory descriptors (the first two variables correspond to the *panelist* and the *product* variables). This data frame contains the means of the products over the sessions for the whole panel and for each panelist (data may be scaled to unit variance or not, this parameter is set by users).

Author(s)

François Husson

Examples

```
data(chocolates)
res=scalebypanelist(sensochoc, col.p = 4, col.j = 1, firstvar = 5)
res
```

---

search.desc *Search for discriminating descriptors*

Description

This function is designed to select the significant descriptors in a data frame

Usage

```
search.desc(matrice, col.j, col.p, firstvar,
      lastvar = ncol(matrice), level = 0.5)
```

Arguments

matrice	a data frame made up of at least two qualitative variables (*product*, *panelist*) and a set of quantitative variables (sensory descriptors)
col.j	the position of the categorical variable which make the variability, panelist for sensory studies. The value of col.j can also be NULL if no categorical variables make the variability.
col.p	the position of the categorical variable of interest, product for sensory studies
firstvar	the position of the first endogenous variable
lastvar	the position of the last endogenous variable (by default the last column of donnee
level	the threshold (P-value) below which variables are considered as discriminating for the following analysis of variance model: descriptor=col.p+col.j

Value

Returns a data frame with all the qualitative variables and only discriminating variables

Author(s)

François Husson

Examples

```
data(chocolates)
## In this example, all the descriptos are discriminated
interesting.desc <- search.desc(sensochoc, col.j = 1, col.p = 4,
    firstvar = 5, level = 0.5)
```

---

senso.cocktail *Sensory data for 16 cocktails*

Description

The data used here refer to the sensory description of 16 cocktails. Each cocktail was evaluated by 12 panelists according to 13 sensory descriptors (only the average of each cocktail are given).

Usage

```
data(cocktail)
```

Format

A data frame with 16 rows and 13 columns: each cocktail was evaluated by 12 panelists according to 13 sensory descriptors.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(cocktail)
```

sensochoc *Sensory data for 6 chocolates*

Description

The data used here refer to the sensory description of six varieties of chocolates sold in France: each chocolate was evaluated twice by 29 panelists according to 14 sensory descriptors.

Usage

```
data(chocolates)
```

Format

A data frame with 348 rows and 19 columns: 5 qualitative variables (Panelist, Session, Form, Rank, Product) and 14 sensory descriptors.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(chocolates)
decat(sensochoc, formul = "~Product+Panelist", firstvar = 5, graph = FALSE)
```

sensopanels *Sensory profiles given by 7 panels*

Description

The data used here refer to six varieties of chocolates sold in France. Each chocolate was evaluated by 7 panels according to 14 sensory descriptors.

Usage

```
data(chocolates)
```

Format

A data frame with 6 rows and 98 columns: each row corresponds to a chocolate and each column to the mean over the panelists of a given panel according to a sensory descriptor.

Source

Département de mathématiques appliquées, Agrocampus Rennes

Examples

```
data(chocolates)
```

`triangle.design` *Construct a design for triangle tests*

Description

Construct a design to make triangle tests.

Usage

```
triangle.design (nbprod , nbpanelist, bypanelist = nbprod*(nbprod-1)/2,
   labprod=1:nbprod, labpanelist=1:nbpanelist)
```

Arguments

`nbprod`	number of products to compare
`nbpanelist`	number of panelists who make the triangle test
`bypanelist`	number of expermient that each panelist can done (by default each panelist make all the comparisons between the products
`labprod`	name of the products (by default, the product are coded from 1 to the number of products
`labpanelist`	name of the panelists (by default, the panelists are coded from 1 to the number of panelists

Details

Triangle test: panelists receive three coded samples. They are told that two of the sample are the same and one is different. Panelists are asked to identify the odd sample.

Value

Returns an data.frame of dimension (*t,3*), where *t* is the number of experiments. In column 1, 2 and 3 the product to test are given. The product in column 1 is by coded "X", in column 2 is coded by "Y" and in column 3 is coded by "Z". Panelist should start by product "X", then "Y" and then by "Z".

Author(s)

François Husson

See Also

`triangle.test`, `triangle.pair.test`

Examples

```
##Example 1
design1 = triangle.design (nbprod = 4, nbpanelist = 8)

##Example 2
design2 = triangle.design(nbprod = 4, nbpanelist = 6, bypanelist = 3,
  labprod=c("prod1","prod2","prod3","prod4"),
  labpanelist=c("John","Audrey","Peter","Martina","James","Lisa"))
```

---

triangle.pair.test *Make a Triangle test for two products*

Description

Make a Triangle test for two products.

Usage

```
triangle.pair.test (nb.good, nb.answer)
```

Arguments

nb.good	number of panelists who identify the odd sample
nb.answer	number of panelists who make the triangle test

Details

Triangle test: panelists receive three coded samples. They are told that two of the sample are the same and one is different. Panelists are asked to identify the odd sample.

Value

Returns

p.value	the p-value of the Triangle test;
Estimation	estimation by Maximum Likelihood of the number of panelists who really perceive the difference between the products;
ML	Maximum Likelihood of the estimation of the number of panelists who really perceive the difference between the products;
minimum	minimum of panelists who should detect the odd product to can say that panelists perceive the difference between the products.

Author(s)

François Husson

See Also

triangle.test, triangle.design

Examples

```
triangle.pair.test (11, 20)
```

---

triangle.test *Make a Triangle test for a set of products*

Description

Make a Triangle test for a set of products.

Usage

```
triangle.test (design, answer, preference = NULL)
```

Arguments

design	a data.frame corresponding to the design use to make the Triangle test (typically the ouput of the function triangle.design
answer	a vector of the answers of all the panelists; all the answer should be "X", "Y" or "Z"
preference	a vector of the preference of the panelists; all the answer should be "X", "Y" or "Z" (by default, there preference are not taken into account)

Details

Triangle test: panelists receive three coded samples. They are told that two of the sample are the same and one is different. Panelists are asked to identify the odd sample.

Value

Returns a list of matrices. Each matrix give the reult for all the pair of products:

nb.comp	a matrix with the number of comparisons done for each pair of products;
nb.ident	a matrix with the number of panelists who indicate the odd product for each pair of products;
p.value	a matrix with the p-value of the Triangle tests for each pair of products;
nb.recognition	estimation of the panelists who really perceived the difference between two product, for each pair of product;
maxML	Maximum Likelihood of the estimation of the number of panelists who really perceive the difference between the products;

confusion	estimation of the percentage of panelists who do not perceived the difference between two product, for each pair of product;
minimum	minimum of panelists who should detect the odd product to can say that panelists perceive the difference between the products, for each pair of products;
preference	number of times that product of row i is prefered that product in column j for the panelists who find the odd product.

Author(s)

François Husson

See Also

triangle.pair.test, triangle.design

Examples

```
design = triangle.design(nbprod = 4, nbpanelist = 6, bypanelist = 3)
answer = c("X","Y","Y","X","Z","X","Y","X","Z",
    "X","X","Z","X","Y","X","Z","X","Y")
triangle.test (design, answer)
```

Index