Package: SensoMineR (via r-universe)

September 12, 2024

Version 1.27

Title Sensory Data Analysis

Author Francois Husson, Sebastien Le, Marine Cadoret

Maintainer Francois Husson <francois.husson@institut-agro.fr>

Depends R (>= 4.0), FactoMineR (>= 2.7)

Imports cluster,KernSmooth,ggplot2,reshape2,AlgDesign,gtools,ggrepel

Description Statistical Methods to Analyse Sensory Data. SensoMineR: A package for sensory data analysis. S. Le and F. Husson (2008).

License GPL (>= 2)

URL http://sensominer.free.fr

BugReports https://github.com/husson/SensoMineR/issues

Encoding UTF-8

Repository https://husson.r-universe.dev

RemoteUrl https://github.com/husson/sensominer

RemoteRef HEAD

RemoteSha 0acab38385a684611ae29153f98d203b05996611

Contents

`analyse_holos` *Analyse Holos data*

Description

Provide numeric tools and graphical tools to analyse Holos data.

Usage

```
analyse_holos(data, method, axes = c(1, 2), graph = TRUE, export.res = FALSE)
```

Arguments

`data`	A list of object, as returned by the `format_holos` function.
`method`	A string corresponding to the holistic task realized by the subjects during the experiment: "N" for Napping task, "S" for Sorting task, or "SN" for Sorted Napping task.
`axes`	A length 2 vector specifying the components of the factorial analysis to plot. By default, the first two components are plotted.
`graph`	A boolean specifying if the graphical outputs of the factorial analysis should be plotted or not. By default, graph = TRUE.
`export.res`	A boolean specifying if all the graphical outputs should be exported in the working directory or not. By default, export.res = FALSE. NB: If method = "N", setting this argument to TRUE is the only way to access the individual cognitive processes.

Value

IDsubjects	A dataframe containing the concordance between the names of the subjects as given in the Holos experiment and their ID.
summary.task	The summary of the task realized by the subjects with pieces of information such as the number of steps performed by each subject, the duration of the task, etc.
res.FA	The results of the factorial analysis as returned by the MFA function of the FactoMineR package for Napping data, by the MCA function for Sorting data, and by the HMFA function for Sorted Napping data.
datasets	All the individuals data sets (digit-tracking data of each subject) and panel data sets (merged final configurations and verbalization).

References

Le, M.T., Brard, M. & Le, S. (2016). Holos: A collaborative environment for similarity-based holistic approaches. Behavior Research Methods.
Le, M.T., Husson, F. & Le, S. (2014). Digit-tracking: Interpreting the evolution over time of sensory dimensions of an individual product space issued from Napping and sorted Napping. Food Quality and Preference.

See Also

format_holos

Examples

```
## Not run:
data(videos)

# Example with Napping data
res.N <- analyse_holos(videos, method = "N", export.res = TRUE)
res.N$summary.task$nbstep.time # number of steps and duration of the task for each subject
res.N$summary.task$freq[[1]] # number of times the first subject moved each stimulus during the task
res.N$res.FA # MFA results that can be customized with the plot.MFA function of FactoMineR
res.N$datasets$digitdata[[1]] # digit-tracking data of the first subject
res.N$datasets$finaldata # Napping data (panel level)

# Example with Sorting data
res.S <- analyse_holos(videos, method = "S")
res.S$res.FA # MCA results that can be customized with the plot.MCA function of FactoMineR
res.S$datasets # Sorting data (panel level)
sorting.data <- apply(res.S$datasets, 2, as.factor) ?
res.fast <- fast(sorting.data)
ConsensualWords(res.fast)

## End(Not run)
```

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)

F Husson, S Le

See Also

decat

Examples

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

## 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)

Francois Husson <francois.husson@institut-agro.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)

S Le <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)
```

---

boot *Simulate virtual panels for several functions*

Description

Simulate virtual panels for the sorting task, the napping, the sorting napping, the free choice profiling, the hierarchical sorting task

Usage

```
boot(X, method = "sorting", axes = 1:2, scale = TRUE, ncp = NULL, group = NULL,
    nbsim = 200,level.conf = 0.95,nbchoix = NULL,color = NULL,cex = 0.8,
    title = NULL, new.plot = TRUE)
```

Arguments

X	data.frame
method	String with the method to use. The argument can be "sorting" (the default for sorting task data), "napping" (for napping data), "sortnapping" (for sorted napping), "freechoice" (for free choice profiling), "hsort" (for hierarchical sorting task data).
axes	a length 2 vector specifying the components to plot
scale	boolean, used when method="freechoice"; if TRUE, the variables are scaled
ncp	number of components used to procrustes the virtual subspaces on the true subspace; by default NULL and the number of components is estimated
group	a list indicating the number of variables in each group; used when method="freechoice" or method="hsort"
nbsim	the number of simulations (corresponding to the number of virtual panels) used to compute the ellipses
level.conf	confidence level used to construct the ellipses. By default, 0.95
nbchoix	the number of panelists forming a virtual panel, by default the number of panelists in the original panel
color	a vector with the colors used; by default there are 35 colors defined
cex	cf. function par in the **graphics** package
title	string corresponding to the title of the graph you draw (by default NULL and a title is chosen)
new.plot	boolean, if TRUE, a new graphical device is created

Details

Calculate virtual panels by bootstrap of the panelists. For each virtual panel, calculate the mean configuration and procrustes this configuration on the true configuration obtained from the true panel.

Value

Returns a list with estim.ncp which corresponds to the output of the estim_ncp function (function which estimates the number of components) and the simul object which can be used with the plotellipse function.

Author(s)

Marine Cadoret and Francois Husson

Examples

```
## Not run:
######## Napping example
data(napping)
res <- boot(napping.don,method="napping")

######## Sorting task example
data(perfume)
res <- boot(perfume,method="sorting")

######## Sorted task napping example
data(smoothies)
res <- boot(smoothies,method="sortnapping")

######## Hierarchical sorting task example
data(cards)
group.cards<-c(2,3,3,2,2,4,2,3,2,1,3,2,3,3,3,2,3,3,2,3,3,3,3,3,3,3,3,3,3)
res <- boot(cards,method="hsort", group=group.cards)

######## Free choice profiling example
data(perfume_fcp)
res <- boot(perfume_fcp, method="freechoice", group = c(12,7,7,7,6,8))

## 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)

F Husson <francois.husson@institut-agro.fr>
S Le <Sebastien.Le@agrocampus-ouest.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)
```

cards *Cards*

Description

The data used here refer to 16 cards (images) on which 30 children performed a hierarchical sorting task.

Usage

```
data(cards)
```

Format

A data frame with 16 rows (the number of cards) and 81 columns (the total number of levels provided by all children). For each child, we have several qualitative variables corresponding to nested partitions: a partition corresponds to a level provided by the child. The columns are grouped by child.

Source

Applied mathematics department, Institut Agro

Examples

```
## Not run:
data(cards)
## Example of FAHST
group.cards<-c(2,3,3,2,2,4,2,3,2,1,3,2,3,3,3,2,3,3,2,3,3,3,3,3,3,3,3,3,3,3)
res.fahst<-fahst(cards,group=group.cards)

## End(Not run)
```

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,
     graph.tree=TRUE,graph.corr=TRUE,graph.carto=TRUE,
     main=NULL,col.min=7.5,col.max=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
graph.tree	boolean, if TRUE plots the tree in 2 dimensions
graph.corr	boolean, if TRUE plots the variables factor map
graph.carto	boolean, if TRUE plots the preference map
main	an overall title for the plot
col.min	define the color which match to the low levels of preference
col.max	define the color which match to the high levels of preference

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)

Francois Husson <francois.husson@institut-agro.fr>
Sebastien Le <Sebastien.Le@agrocampus-ouest.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

```
## Not run:
## Example 1: carto for the sensory descriptors
data(cocktail)
res.pca <- PCA(senso.cocktail)
res.carto <- carto(res.pca$ind$coord[,1:2], hedo.cocktail)

## Example 2
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"))
res.carto <- carto(res.mfa$ind$coord[,1:2], hedo.cocktail)

## End(Not run)
```

`cartoconsumer` *Preference Mapping Techniques and segmentation of consumers*

Description

Performs preference mapping techniques based on multidimensional exploratory data analysis and segmentation of consumers.

Usage

```
cartoconsumer(res, data.pref, nb.clust=0, seuil=0.8, consol=TRUE, ncp=5,
   scale.conso=TRUE,graph.carto=TRUE,graph.hcpc=FALSE, graph.group=FALSE,
  col.min=7.5, col.max=0, contrast=0.2, level=0, asp=0,lwd=2)
```

Arguments

`res`	the result of a factor analysis
`data.pref`	a data frame in which each row represent a product and each column represent the hedonic scores of a given consumer for the products

nb.clust	an integer. If 0, the tree is cut at the level the user clicks on. If -1, the tree is automatically cut at the suggested level (see details). If a (positive) integer, the tree is cut with nb.cluters clusters
seuil	the size of the area kept for each group of consumers
consol	a boolean. If TRUE, a k-means consolidation is performed
ncp	number of dimensions kept in the results (by default 5)
scale.conso	scale data by consumer
graph.carto	if TRUE, the preference map is displayed. If FALSE, no graph is displayed
graph.hcpc	if TRUE, graphics of segmentation (trees and indivuals map) are displayed. If FALSE, no graph are displayed
graph.group	if TRUE, preference maps for each group are displayed. If FALSE, no map are displayed
col.min	define the color which match to the low levels of preference
col.max	define the color which match to the high levels of preference
contrast	define the color contrast between groups' areas and the rest of the map
level	the number of standard deviations used in the calculation of the preference response surface for all the consumers
asp	if 1 is assigned to that parameter, the graphic displays are output in an orthonormal coordinate system
lwd	control the line width for the outlines defining groups' areas

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 segmentation of consumers is performed, and a preference map is displayed for each group of consumers. The original preference map is built, the areas of each group are underlined thanks to a contrast, and the number of consumers is shown.

Author(s)

Francois Husson <husson@agrocampus-rennes.fr>
Sophie Birot and Celia Pontet

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, carto

Examples

```
## Not run:
## Example 1: carto on the sensory descriptors
data(cocktail)
res.pca <- PCA(senso.cocktail)
results1 <- cartoconsumer(res.pca, hedo.cocktail)
results2 <- cartoconsumer(res.pca, hedo.cocktail,
      graph.hcpc=TRUE,graph.group=TRUE)

## End(Not run)

## 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"))
results3 <- cartoconsumer(res.mfa, hedo.cocktail)

## End(Not run)
```

CA_JAR *Make a correspondence analysis on the JAR data*

Description

Plot the CA graph.

Usage

```
CA_JAR(x, col.p, col.j, col.pref, jarlevel="jar")
```

Arguments

x	data.frame
col.p	the position of the *product* variable
col.j	the position of the *panelist* variable
col.pref	the position of the *preference* variable
jarlevel	a string corresponding to the jar level (the level must be the same for all the jar variables)

Value

Draw a CA graph with the preference data as supplementary qualitative variables, the products as rows, and the categories of the jar variables as columns

Author(s)

Francois Husson

Examples

```
## Not run:
data(JAR)
res <- CA_JAR(x = JAR, col.p = 13, col.j = 1, col.pref = 2, jarlevel = "jar")
plot.CA(res$res.CA, invisible = "row", cex = 0.8)

## 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

Applied mathematics department, Institut Agro

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 disliking (0) or liking (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 disliking (0) or liking (10).

Source

Applied Mathematics Department, 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)

F Husson, S Le

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

Applied mathematics department, Institut Agro

Examples

```
data(cocktail)
```

`ConsensualWords` *Consensual words for Sorting Task data*

Description

This function is designed to point out the words that are used in a consensual way by consumers from a sorting task.

Usage

```
ConsensualWords(res.fast, nbtimes = 3, nbsimul = 500, proba = 0.05,
    graph = TRUE, axes = c(1,2))
```

Arguments

`res.fast`	an object of class fast
`nbtimes`	minimum sample size for the word selection
`nbsimul`	the number of simulations used to compute Bootstrap
`proba`	the significance threshold considered to consider a word as consensual (by default 0.05)
`graph`	boolean, if TRUE a graph is displayed
`axes`	a length 2 vector specifying the components to plot

Value

A list containing the following elements:

`Centroids`	coordinates of the consensual words on the dimensions of the fast result
`Within.inertia`	frequency of use of each word and within inertia associated
`Results.Bootstrap`	frequency of use of each word, within inertia associated and p-value calculated according to the Bootstrap technique
`Consensual.words`	a list of significant consensual words sorted from the most consensual to the less consensual

Author(s)

Francois Husson

Examples

```
## Not run:
data(perfume)
## Example of FAST results
res.fast<-fast(perfume,sep.words=";")
res.consensual<-ConsensualWords(res.fast)

## End(Not run)
```

`ConsistencyIdeal` *Sensory and Hedonic consistency of the ideal data*

Description

Evaluate the sensory and hedonic consistency of the ideal data, both at the consumer and panel level.

Usage

```
ConsistencyIdeal(dataset, col.p, col.j, col.lik, id.recogn, type="both", scale.unit=TRUE,
                 ncp=NULL, axes=c(1,2), nbsim=0, replace.na=FALSE, graph=TRUE)
```

Arguments

`dataset`	A matrix with at least two qualitative variables (*consumer* and *products*) and a set of quantitative variables containing at least 2*A variables (for both *perceived* and *ideal* intensities)
`col.p`	The position of the *product* variable
`col.j`	The position of the *consumer* variable
`col.lik`	The position of the *liking* variable
`id.recogn`	The sequence in the variable names which distinguish the ideal variables from the sensory variables. This sequence should be fixed and unique. Each ideal variable should be preceeded by the corresponding perceived intensity variable.
`type`	Define whether you want the sensory consistency only ("sensory"), the hedonic consistency only ("hedonic"), or both ("both")
`scale.unit`	Boolean, if TRUE the descriptors are scaled to unit variance
`ncp`	Number of dimensions kept in the results
`axes`	A length 2 vector specifying the components to plot
`nbsim`	The number of simulations performed. By default (=0), no simulations are performed and only the results for the real data are given
`replace.na`	Boolean, define whether the missing values (in the correlation matrix calculated for the consistency at the consumer level) should be ignored or replaced by 0
`graph`	Boolean, define whether the distribution of the correlation coefficient should be plot

Details

SENSORY CONSISTENCY
A the panel level:
A PCA is performed on the table crossing the J consumers in rows and the A ideal variables in columns (the averaged or corrected averaged is then considered).
On this space, the sensory description of the P products (P rows) on the A attributes is projected as supplementary entities while the hedonic table crossing the J consumers (in rows) and the P products (in columns) is projected as supplementary variables.
The sensory consistency is measured by the correspondence between the same products seen through the sensory and through the hedonic descriptions.
At the consumer level:
For each consumer, the correlation between the (corrected) ideal ratings and the correlation between the hedonic scores and the perceived intensity of each attribute is calculated.
A test on this correlation coefficient is performed for each consumer.
The distribution of these correlations coefficients are also given graphically.

HEDONIC CONSISTENCY
For each consumer, a PCR-model expressing the liking scores in function of the perceived intensity is created.
Once the model is created, the model is applied to the ideal ratings provided by the consumer considered and the hedonic score of the ideal product is estimated.
This hedonic score is then compared to the hedonic scores provided to the products tested.
If simulations are asked, the same procedure is estimated after re-sampling the vector of hedonic scores.
In that case, the distribution of the estimated ideal hedonic score can be estimated under H0 and the significance of the estimated ideal hedonic score can be done.

Value

A list containing the results for the sensory and hedonic consistency:

`Senso` contains the results of the sensory consistency

`Senso$panel` results for the consistency at the panel level including:

`Senso$panel$dataset`
the datasets used for the different PCA

`Senso$panel$PCA.ideal`
the results of the PCA for the creation of the ideal space

`Senso$panel$PCA.ideal_hedo`
the results of the PCA with projection of the hedonic scores

`Senso$panel$PCA.ideal_senso`
the results of the PCA with the proojection of the sensory descriptions

`Senso$panel$correlation`
the correlation between the *product* projected from the *sensory* and *hedonic* points of view

`Senso$conso` results of the consistency at the consumer level including:

`Senso$conso$driver.lik`
: the linear drivers of liking (correlation between perceived intensity and liking score for each attribute)

`Senso$conso$correlations`
: the correlations between drivers of liking and the difference (ideal-perceived) intensity

`Hedo$R2`
: the R2 coefficients of the indivvidual models

`Hedo$hedo`
: a list containing the hedonic scores for the *product*, *ideal products*, *average ideal product* and the *standardized ideal product* for each consumer

`Hedo$simulation`
: a list including the estimated hedonic score for each simulation, the p-value and the matrix of simulations used

Author(s)

Thierry Worch (thierry@qistatistics.co.uk)

References

Worch, T., Le, S., Punter, P., & Pages, J. (2012). Assessment of the consistency of ideal profiles according to non-ideal data for IPM. *Food Quality and Preference*, 24, 99-110., Worch, T., Le, S., Punter, P., & Pages, J. (2012). Extension of the consistency of the data obtained with the Ideal Profile Method: Would the ideal products be more liked than the tested products? *Food Quality and Preference*, 26, 74-80.

See Also

`panelperf`, `paneliperf`

Examples

```
## Not run:
data(perfume_ideal)
res <- ConsistencyIdeal(perfume_ideal, col.p=2, col.j=1,
   col.lik=ncol(perfume_ideal), id.recogn="id_",
   type="both", nbsim=100)

## End(Not run)
```

`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", graph.type = c("ggplot","classic"))
```

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)
`graph.type`	a character that gives the type of graph used: "ggplot" or "classic"

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)

Francois Husson

References

Escofier, B. and Pages, J. (1990) *Analyses factorielles simples et multiples: objectifs, methodes et interpretation* Dunod, Paris. 1–267.

Escofier, B. and Pages, 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,
    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
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
desc.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)

F Husson <francois.husson@institut-agro.fr>
S Le

References

S. Le, F. Husson, J. Pages (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,
    col = gray((50:1)/50), nb.clusters = 3)

## End(Not run)
```

---

cream_id *Cream Ideal Data*

Description

The data used here refer to the sensory description of 9 dessert chocolate creams.

Each cream was evaluated once by 86 French consumers and described on 13 attributes according to the Ideal Profile Method.
Both perceived and ideal intensities were asked. In addition, the overall liking is asked.

Usage

```
data(cream_id)
```

Format

A data frame made of 86*9=774 rows and 2 qualitative variables (panelist and product), 13*2 attributes (perceived and ideal intensities) and overall liking.

Source

Institut Agro, Melodie Sanchez, Sarah Sanchez

Examples

```
## Not run:
data(cream_id)
decat(cream_id, formul = "~product+user", firstvar = 3, graph = FALSE)

###IdMapConsumer function
data(cream_signa)
res.idmap <- IdMapConsumer(craem_id, cream_signa, col.p=2, col.j=1, col.lik=29,
num.col.var.signa=c(1:12), conf.level=0.90, id.recogn="id_", color = FALSE, simusigni = 500)


## End(Not run)
```

---

cream_signa *Data description of the consumers who made the Ideal for the cream*

Description

The data used here refer to the sensory description of 9 dessert chocolate creams.

Each cream was evaluated once by 86 French consumers and described on 13 attributes according to the Ideal Profile Method.
Both perceived and ideal intensities were asked. In addition, the overall liking is asked.

Usage

```
data(cream_signa)
```

Format

A data frame made of 86*9=774 rows and 2 qualitative variables (panelist and product), 13*2 attributes (perceived and ideal intensities) and overall liking.

Source

Institut Agro, Melodie Sanchez, Sarah Sanchez

Examples

```
## Not run:
data(cream_signa)
data(cream_signa)
res.idmap <- IdMapConsumer(cream_id, cream_signa, col.p=2, col.j=1, col.lik=29,
num.col.var.signa=c(1:12), conf.level=0.90, id.recogn="id_", color = FALSE, simusigni = 500)


## 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
`adjmean`	a (products,descriptors) data frame, in which each cell is the adjusted mean resulting from the analysis of variance model for a given product and a given descriptor

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 (significantly different with the `proba` level); the significantly less are colored in red (by default) and the significantly great are colored in blue.

Author(s)

Francois 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)
```

`fahst` *Factorial Approach for Hierarchical Sorting Task data*

Description

Perform Factorial Approach for Hierarchical Sorting Task data (FAHST) on a table where the rows (i) are products and the columns (j) are for each consumer the partitionning variables associated with nested sorting. The columns are grouped by consumer. For the partitionning variables, the label associated with a group can be an arbirary label (for example G1 for group 1, *etc.*) or the words associated with the group in the case of qualified hierarchical sorting.

Usage

```
fahst(don,group,alpha=0.05,graph=TRUE,axes=c(1,2),name.group=NULL,ncp=5,B=200,ncp.boot=2)
```

Arguments

`don`	a data frame with n rows (products) and p columns (nested partitions for all consumers)
`group`	a list indicating the number of levels (nested partitions) for each consumer
`alpha`	the confidence level of the ellipses
`graph`	boolean, if TRUE a graph is displayed
`axes`	a length 2 vector specifying the components to plot
`name.group`	a vector containing the name of the consumers (by default, NULL and the consumers are named J1, J2 and so on)
`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
`ncp.boot`	number of dimensions used for the Procrustean rotations to build confidence ellipses (by default 2)

Value

A list containing the following elements:

eig	a matrix containing all the eigenvalues, the percentage of variance and the cumulative percentage of variance
ind	a list of matrices containing all the results for the products (coordinates, square cosine, contributions)
var	a list of matrices containing all the results for the categories of the different nested partitions (coordinates, square cosine, contributions, v.test)
group	a list of matrices containing all the results for consumers (coordinates, square cosine, contributions)
call	a list with some statistics

Author(s)

Marine Cadoret, Sebastien Le <sebastien.le@institut-agro.fr>

References

Cadoret, M., Le, S., Pages, J. (2010) *A new approach for analyzing hierarchical sorting task data*. Sensometrics conference. Rotterdam, the Netherlands

Examples

```
## Not run:
data(cards)
## Example of FAHST results
group.cards<-c(2,3,3,2,2,4,2,3,2,1,3,2,3,3,3,2,3,3,2,3,3,3,3,3,3,3,3,3,3,3,3)
res.fahst<-fahst(cards,group=group.cards)

## End(Not run)
```

---

fasnt *Factorial Approach for Sorting Napping Task data*

Description

Perform Factorial Approach for Sorting Napping Task data (FASNT) on a table where the rows (i) are products and the columns (j) are for each consumer the coordinates of the products on the tablecloth associated with napping on the one hand and the partitionning variable associated with categorization on the other hand. The columns are grouped by consumer. For the partitionning variable, the label associated with a group can be an arbirary label (for example G1 for group 1, *etc.*) or the words associated with the group in the case of qualified sorted napping.

Usage

```
fasnt(don,first="nappe",B=100,axes=c(1,2),alpha=0.05,ncp=5,
     graph=TRUE,name.group=NULL,sep.word=" ",word.min=5,ncp.boot=2)
```

Arguments

don	a data frame with n rows (products) and p columns (assesor : categorical variables)
first	2 possibilities: "nappe" if the napping variables first appear for each consumer or "catego" if it is the categorization variable
B	the number of simulations (corresponding to the number of virtual panels) used to compute the ellipses
axes	a length 2 vector specifying the components to plot
alpha	the confidence level of the ellipses
ncp	number of dimensions kept in the results (by default 5)
graph	boolean, if TRUE a graph is displayed
name.group	a vector containing the name of the consumers (by default, NULL and the group are named J1, J2 and so on)
sep.word	the word separator character in the case of qualified sorted napping
word.min	minimum sample size for the word selection in textual analysis
ncp.boot	number of dimensions used for the Procrustean rotations to build confidence ellipses (by default 2)

Value

A list containing the following elements:

eig	a matrix containing all the eigenvalues, the percentage of variance and the cumulative percentage of variance
ind	a list of matrices containing all the results for the products (coordinates, square cosine, contributions)
quali.var	a list of matrices containing all the results for the categories of categorization (coordinates, square cosine, contributions, v.test)
quanti.var	a list of matrices containing all the results for the napping (coordinates, square cosine, contributions, v.test)
group	a list of matrices containing all the results for consumers (coordinates, square cosine, contributions)
indicator	a list of matrices containing different indicators for napping and categorization
textual	the results of the textual analysis for the products
call	a list with some statistics

Author(s)

Marine Cadoret, Sebastien Le <sebastien.le@institut-agro.fr>

References

Pag\‘es, J., Le, S., Cadoret, M. (2010) *The Sorted Napping: a new holistic approach in sensory evaluation*. Journal of Sensory Studies
Cadoret, M., Le, S., Pages, J. (2009) *Combining the best of two worlds, the "sorted napping"*. SPISE. Ho Chi Minh City, Vietnam

Examples

```
## Not run:
data(smoothies)
## Example of FASNT results
res.fasnt<-fasnt(smoothies,first="nappe",sep.word=";")

## End(Not run)
```

`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 for the consumer j, or, in the case of "qualified" categorization, to the sequence of words associted with the group to which the product i belongs for the consumer j.

Usage

```
fast(don,alpha=0.05,sep.words=" ",word.min=5,graph=TRUE,axes=c(1,2),
     ncp=5,B=200,label.miss=NULL,ncp.boot=NULL)
```

Arguments

`don`	a data frame with n rows (products) and p columns (assesor : categorical variables)
`alpha`	the confidence level of the ellipses
`sep.words`	the word separator character in the case of qualified categorization
`word.min`	minimum sample size for the word selection in textual analysis
`graph`	boolean, if TRUE a graph is displayed
`axes`	a length 2 vector specifying the components to plot
`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
`label.miss`	label associated with missing groups in the case of incomplete data set
`ncp.boot`	number of dimensions used for the Procrustean rotations to build confidence ellipses (by default NULL and the number of components is estimated)

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)
acm	all the results of the MCA
cooccur	the reordered co-occurrence matrix among products
reord	the reordered matrix products*consumers
cramer	the Cramer's V matrix between all the consumers
textual	the results of the textual analysis for the products
call	a list with some statistics

Author(s)

Marine Cadoret, Sebastien Le <sebastien.le@institut-agro.fr>

References

Cadoret, M., Le, S., Pages, J. (2008) *A novel Factorial Approach for analysing Sorting Task data*. 9th Sensometrics meeting. St Catharines, Canada
Cadoret, M., Le, S., Pages, J. (2009) *A Factorial Approach for Sorting Task data (FAST)*. Food Quality and Preference. 20. pp. 410-417
Cadoret, M., Le, S., Pages, J. (2009) *Missing values in categorization*. Applied Stochastic Models and Data Analysis (ASMDA). Vilnius, Lithuania

Examples

```
## Not run:
data(perfume)
## Example of FAST results
res.fast<-fast(perfume,sep.words=";")
res.consensual<-ConsensualWords(res.fast)

## End(Not run)
```

fcp *Free choice profiling*

Description

Free choice profiling with confidence ellipses

Usage

```
fcp(X,group,scale=TRUE, ncp = NULL, axes=c(1,2), name.group = NULL, level.conf = 0.95,
    nbsim=500, nbchoix=NULL, cex=1, color=NULL, title=NULL, new.plot=TRUE,
 graph=c("ind","var","ellipse"))
```

Arguments

X	data.frame
group	a list indicating the number of variables in each group; used when method="freechoice" or method="hsort"
scale	boolean, used when method="freechoice"; if TRUE, the variables are scaled
ncp	number of components used to procrustes the virtual subspaces on the true subspace; NULL by default and the number of components is estimated
axes	a length 2 vector specifying the components to plot
name.group	the names of each group of variables
level.conf	confidence level used to construct the ellipses. By default, 0.95
nbsim	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
cex	cf. function par in the **graphics** package
color	a vector with the colors used; by default there are 35 colors defined
title	string corresponding to the title of the graph you draw (by default NULL and a title is chosen)
new.plot	boolean, if TRUE, a new graphical device is created
graph	list with the graphs to draw; "ind", "var" and "ellipse" by default

Details

Perform MFA on the data frame and calculate confidence ellipses around the products

Value

Returns a list with the result of the MFA and the bootstraped results that can be used with the plotellipse function.

Author(s)

Francois Husson

Examples

```
## Not run:
data(perfume_fcp)
res <- fcp(perfume_fcp, group = c(12,7,7,7,6,8))

## End(Not run)
```

`format_holos` *Format Holos data*

Description

Format the data exported from the Holos platform.

Usage

```
format_holos(path.data)
```

Arguments

`path.data`	The path corresponding to the folder containing all Holos data (i.e. one folder per subject with different files: "X_comment.txt", "X_data.txt", "X_last.txt", "X_txt").

Value

A list of 4 objects: `IDsubjects`, a dataframe containing the concordance between the names of the subjects as given in the Holos experiment and their ID; `datadigit`, a list of S (S = number of subjects) dataframes corresponding to the digit-tracking data; `datafinal_coord`, a list of S (S = number of subjects) dataframes corresponding to the final configurations data; and `datafinal_verb`, a list of S (S = number of subjects) dataframes corresponding to the final verbalization data.

See Also

`analyse_holos`

Examples

```
## Not run:
data <- format_holos(path.data = "C:/MyDirectory/")
## 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)

F Husson, S Le

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

Applied mathematics department, Institut Agro

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

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)

S Le

See Also

`density`, `hist`

Examples

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

---

hsortplot *Plot consumers' hierarchical sorting*

Description

Plot consumers' hierarchical sorting

Usage

```
hsortplot(don, group, numr = 2, numc = 2)
```

Arguments

`don`	a data frame with n rows (products) and p columns (nested partitions for all consumers)
`group`	a list indicating the number of levels (nested partitions) for each consumer
`numr`	the number of hierarchical sorting per row (by default 2)
`numc`	the number of hierarchical sorting per column (by default 2)

Details

The data used here refer to a specific experiment, where children were asked to provide hierarchical sorting (several nested partitions) from 16 cards.

Value

Returns as many graphs as there are consumers, each graph represents hierarchical sorting provided by a consumer

Author(s)

Marine Cadoret, Sebastien Le <sebastien.le@institut-agro.fr>

See Also

fahst

Examples

```
## Not run:
data(cards)
group.cards<-c(2,3,3,2,2,4,2,3,2,1,3,2,3,3,3,2,3,3,2,3,3,3,3,3,3,3,3,3,3,3)
hsortplot(cards,group.cards)

## End(Not run)
```

IdMap *Ideal Mapping (IdMap)*

Description

Create the ideal map, a map based on the ideal profiles provided by the consumers.

Usage

```
IdMap(dataset, col.p, col.j, col.lik, id.recogn, nbsimul=500, nbchoix=NULL,
      alpha=0.05, coord=c(1,2), precision=0.1, levels.contour=NULL,
   color=FALSE, cons.eq=FALSE)
```

Arguments

dataset	A matrix with at least two qualitative variables (*consumer* and *products*) and a set of quantitative variables containing at least 2*A variables (for both *perceived* and *ideal* intensities)
col.p	The position of the *product* variable
col.j	The position of the *consumer* variable
col.lik	The position of the *liking* variable
id.recogn	The sequence in the variable names which distinguish the ideal variables from the sensory variables. This sequence should be fixed and unique. Each ideal variable should be preceded by the corresponding perceived intensity variable.
nbchoix	The number of consumers forming a virtual panel, by default the number of panelists in the original panel
nbsimul	The number of simulations (corresponding to the number of virtual panels) used to compute the ellipses

alpha	The confidence level of the ellipses
coord	A length 2 vector specifying the components to plot
precision	The value defining the step when gridding the space
levels.contour	The levels (between 0 and 1) to consider for the colors on the surface plot. By default, they are set automatically based on the results
color	Boolean, define whether the map is in color or in black and white
cons.eq	Boolean, define whether the IdMap (by default) or the wIdMap is performed

Details

The IdMap, step by step:
Step 1: the sensory and ideal variables are separated into two tables.
Step 2: the product space is created by PCA on the averaged sensory table (averaged by product).
Step 3: the averaged ideal product of each consumer is projected as supplementary entities in this space.
Step 4: confidence ellipses are created around each individual averaged ideal product using truncated total bootstrap.
Step 5: for each consumer, the space is grid and the position where the ideal area is defined is marked: individual surfaces of response are created.
Step 6: (optional) the ellipses can be balanced by applying individual weight (all the ellipses have a weigth of 1, however the size of the ellipse). wIdMap is then performed.
Step 7: all the individual surface plots are added together and a surface plot is created.

Value

A list containing the following components:

PCA	the results from the PCA used to create the sensory space
idmap	a list containing the results of the IdMap (*data*), the weight for each consumer (*j.weight*) and the precision used.
ideal	a list containing the estimated profile of the ideal of reference (not available for the wIdMap) as well as the percentage of consumers concerned

Author(s)

Worch Thierry (thierry@qistatistics.co.uk)

References

Worch, T., Le, S., Punter, P., Pages, J. (2012). Construction of an Ideal Map (IdMap) based on the ideal profiles obtained directly from consumers. *Food Quality and Preference*, 26, 93-104.

See Also

plot.IdMap, carto, boot

Examples

```
## Not run:
data(perfume_ideal)

#! For the IdMap
res.IdMap <- IdMap(perfume_ideal, col.p=2, col.j=1,
   col.lik=ncol(perfume_ideal), id.recogn="id_")
plot.IdMap(res.IdMap, xlim=c(-7,10), ylim=c(-5,7), levels.contour=NULL, color=TRUE)

#! For the wIdMap
res.wIdMap <- IdMap(perfume_ideal, col.p=2, col.j=1, col.lik=ncol(perfume_ideal),
   id.recogn="id_", cons.eq=TRUE)

## End(Not run)
```

`IdMapConsumer` *Ideal Mapping by categories (IdMapConsumer)*

Description

Create the ideal map and plot the ideal areas of the categories of qualitative variables. And perform 2 tests: a global test in order to highlight the significance of the difference between ideals of all the categories of the same variable; a pair comparison test to highlight the significance between 2 categories of the same variable.

Usage

```
IdMapConsumer(dataset.id, dataset.signa, col.p, col.j, col.lik,
 num.col.var.signa, conf.level=0.95, id.recogn, nbchoix = NULL,
    nbsimul = 500, alpha = 0.05, coord = c(1, 2), precision = 0.1,
    levels.contour = NULL, color = FALSE, simusigni = 500)
```

Arguments

`dataset.id`	A matrix with at least two qualitative variables (*consumer* and *products*) and a set of quantitative variables containing at least 2*A variables (for both *perceived* and *ideal* intensities)
`dataset.signa`	a data frame with n rows (individuals) and p columns (categorical variables)
`col.p`	The position of the *product* variable in the dataframe *dataset.id*
`col.j`	The position of the *consumer* variable in the dataframe *dataset.id*
`col.lik`	The position of the *liking* variable in the dataframe *dataset.id*
`id.recogn`	The sequence in the variable names which distinguish the ideal variables from the sensory variables. This sequence should be fixed and unique. Each ideal variable should be preceeded by the corresponding perceived intensity variable.

`num.col.var.signa`
: The position of the categorical variables in the dataframe *dataset.signa* you want to plot the ideal area of the different modalities/you want to know if the ideal product of the different modalities is significantly different

`conf.level`
: Threshold used for the tests

`nbchoix`
: The number of consumers forming a virtual panel, by default the number of panelists in the original panel

`nbsimul`
: The number of simulations (corresponding to the number of virtual panels) used to compute the ellipses

`alpha`
: The confidence level of the ellipses

`coord`
: A length 2 vector specifying the components to plot

`precision`
: The value defining the step when gridding the space

`levels.contour`
: The levels (between 0 and 1) to consider for the colors on the surface plot. By default, they are set automatically based on the results

`color`
: Boolean, define whether the map is in color or in black and white

`simusigni`
: The number of simulations used to perform the global and the pair comparison test

Details

The IdMapConsumer, step by step: Step 1: the classical IdMap is plotted with the method "ellipses"
Step 2: for each modality of the categorical variable, the optimum of the ideal area is calculated with the method "density"
Step 3: for each categorical variable given in *num.col.var.signa*, simulations are performed giving the p-value for the global ant the pair comparison test.
Step 4: if the global test is significant for a variable, the ideal areas of its modalities are plotted on the IdMap

This function needs the KernSmooth package.

Value

A list containing the following components:

`PCA`
: the results from the PCA used to create the sensory space

`idmap`
: a list containing the results of the IdMap (*data*), the weight for each consumer (*j.weight*) and the precision used.

`ideal`
: a list containing the estimated profile of the ideal of reference (not available for the wIdMap) as well as the percentage of consumers concerned

`coordobs`
: The coordinates of all the ideals of all the categories on the sensory space

`test.global`
: The results for the global test for each variables (observed inertia, critical inertia, P-value)

`test.paires`
: The results for the pair comparison test for each variables, between its ideal's categories(observed distance between two categories, critical distance, P-value)

The three last components are provided only if the user choose "color = FALSE", else no test and no ideal map with categories' ideal are performed.

Author(s)

Melodie Sanchez, Sarah Sanchez, francois.husson@institut-agro.fr

References

Worch, T., Le, S., Punter, P., & Pages, J. (2012). Construction of an Ideal Map (IdMap) based on the ideal profiles obtained directly from consumers. *Food Quality and Preference*, 26, 93-104.

See Also

IdMap

Examples

```
## Not run:
###Load the two datasets
data(cream_id)
data(cream_signa)

###Run the analysis and test the ideals of the variables from 1 to 12
## for example with a confidence level of 90
res.idmap <- IdMapConsumer(cream_id, cream_signa, col.p=2, col.j=1, col.lik=29,
   num.col.var.signa=c(1:12),conf.level=0.90,id.recogn="id_")

## End(Not run)
```

---

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
Francois 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)
prefpls(cbind(resindscal$points, napping.words))
pmfa(napping.don, napping.words, mean.conf = resindscal$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:Pan`
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)

Francois Husson

See Also

`aov`

Examples

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

## End(Not run)
```

JAR *JAR*

Description

Just About Right

Usage

```
JAR(x, col.p, col.j, col.pref, jarlevel="jar")
```

Arguments

x	data.frame
col.p	the position of the *product* variable
col.j	the position of the *panelist* variable
col.pref	the position of the *preference* variable
jarlevel	a string corresponding to the jar level (the level must be the same for all the jar variables)

Details

Perform the penalty analysis. Two models are constructed.
The one-dimensional model is constructed descriptor by descriptor. For descriptor_j the model is:
Hedonic score = Descriptor_j_Not enough+ Descriptor_j_Too much
The multi-dimensional model is constructed with all descriptors simultaneously:
Hedonic score = Descriptor_1_Not enough+ Descriptor_1_Too much +...+ Descriptor_p_Not enough+ Descriptor_p_Too much+ Product + Judge

Value

Returns a list of 3 objects.
The penalty1 object corresponds to the one-dimensional penalty results: a data-frame with the penalty coefficient in the first column, the standard deviation and the p-value for the test that the penalty is significantly different from 0.
The penalty2 object corresponds to the mutli-dimensional penalty results: a data-frame with the penalty coefficient in the first column, the standard deviation and the p-value for the test that the penalty is significantly different from 0. The Frequency object gives the percentage of times the non-jar categories are given for each product: a matrix with the non-jar categories in rows and the products in columns

Author(s)

Francois Husson

See Also

plot.JAR

Examples

```
## Not run:
data(JAR)
res.jar <- JAR(JAR,col.p=13,col.j=1,col.pref=2)
plot(res.jar,name.prod="284", model=1)

## 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)

F Husson, S Le

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")
```

MultiIdeal *Single vs. Multiple Ideal*

Description

By the use of confidence ellipses, this procedure checks whether consumers associate the different products tested to a single or to multiple ideals.

Usage

```
MultiIdeal(dataset, col.p, col.j, id.recogn, level.search.desc=0.2, correct=FALSE,
            nbchoix=NULL, nbsimul=500, coord=c(1,2))
```

Arguments

dataset	A matrix with at least two qualitative variables (*consumer* and *products*) and a set of quantitative variables containing at least 2*A variables (for both *perceived* and *ideal* intensities)
col.p	The position of the *product* variable
col.j	The position of the *consumer* variable
id.recogn	The sequence in the variable names which distinguish the ideal variables from the sensory variables. This sequence should be fixed and unique. Each ideal variable should be preceeded by the corresponding perceived intensity variable.
level.search.desc	the threshold above which a descriptor is not considered as discriminant according to AOV model "descriptor=Product+Panelist".
correct	Boolean, define whether the ideal products should be corrected from the difference in the use of the scale or not

nbchoix	The number of consumers forming a virtual panel, by default the number of panelists in the original panel
nbsimul	The number of simulations (corresponding to the number of virtual panels) used to compute the ellipses
coord	A length 2 vector specifying the components to plot

Details

The procedure of MultiIdeal, step by step:
Step 1: the sensory and ideal variables are separated into two tables.
Step 2: the product space is created by PCA on the averaged sensory table (averaged by product).
Step 3: the ideal information (Product x Consumer) is projected as supplementary entities in this space.
Step 4: confidence ellipses are created around the averaged ideal points associated to each product (using the consumer variability).

Value

Returns a matrix with the P-values of the Hotelling's T2 tests for each pair of products.

Author(s)

Worch Thierry (thierry@qistatistics.co.uk)

References

Worch, T., & Ennis, J.M. (2013). Investigating the single ideal assumption using Ideal Profile Method. *Food Quality and Preference*.

See Also

panellipse

Examples

```
## Not run:
data(perfume_ideal)
res <- MultiIdeal(perfume_ideal, col.p=2, col.j=1, id.recogn="id_",
    level.search.desc=0.2, nbsimul=500, coord=c(1,2))

# To run the analysis with all the attributes
res <- MultiIdeal(perfume_ideal, col.p=2, col.j=1, id.recogn="id_",
    level.search.desc=1, nbsimul=500, coord=c(1,2))

## End(Not run)
```

`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)

Francois Husson

References

Pages 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)
```

`nappesortplot` *Plot consumers' sorted tablecloth*

Description

Plot consumers' sorted tablecloth.

Usage

```
nappesortplot(donnee,first="nappe", numr = 2, numc = 2, lim = c(60,40))
```

Arguments

`donnee`	a data frame of dimension (*p,3j*), where *p* represents the number of products and *j* the number of consumers
`first`	2 possibilities: "nappe" if the napping variables first appear for each consumer or "catego" if it is the categorization variable
`numr`	the number of tablecloth per row (by default 2)
`numc`	the number of tablecloth per column (by default 2)
`lim`	the size of the tablecloth

Details

The data used here refer to a specific experiment, where consumers are asked to position products on a tablecloth of dimension `lim`, by default (60,40) and to categorize them.

Value

Returns as many graphs as there are consumers, each graph represents products positioned by a given consumer on a tablecloth and colored according to the categorization

Author(s)

Marine Cadoret, Sebastien Le <sebastien.le@institut-agro.fr>

References

Pages, J., Le, S., Cadoret, M. (2010) *The Sorted Napping: a new holistic approach in sensory evaluation*. Journal of Sensory Studies

See Also

fasnt

Examples

```
## Not run:
data(smoothies)
nappesortplot(smoothies,first="nappe")

## 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

Applied mathematics department, Institut Agro

Examples

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

Applied mathematics department, Institut Agro

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
res <- pmfa(napping.don, napping.words)
res2 <- boot(napping.don,method="napping")

## 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

Applied mathematics department, Institut Agro

Examples

```
## Not run:
data(napping)
nappeplot(napping.don)
dev.new()
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=nbProd, seed=NULL)
```

Arguments

`nbPanelist`	Maximum number of panelists
`nbProd`	Number of products
`nbProdByPanelist`	Number of products that each panelist will evaluate
`seed`	initialization of the algorithm

Value

List with

`design`	design with the products evaluated by each panelist
`rank`	contingency table with the number of times each product is seen at each rank
`succ`	contingency table with the number of times each product follow each product

Author(s)

F. Husson

Examples

```
## Not run:
optimaldesign(nbPanelist=10,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)

F Husson, S Le

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 = 300,
    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,
 graph.type = c("ggplot","classic"))
```

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
graph.type	a character that gives the type of graph used: "ggplot" or "classic"

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 significantly 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.
`correl`	a list with: the matrix of the estimated correlation coefficients and two matrices corresponding to the confidence intervals, min and max, of the correlation coefficients calculated by bootstrap.

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)

Francois Husson

References

Husson F., Le Dien S. & Pages J. (2005). Confidence ellipse for the sensory profiles obtained by Principal Components Analysis. *Food Quality and Preference*. 16 (3), 245-250.
Pages 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., Le S. & Pages 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,
 graph.type = c("ggplot","classic"))
```

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
graph.type	a character that gives the type of graph used: "ggplot" or "classic"

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)

F Husson, S Le

References

Husson F., Le Dien S. & Pages J. (2005). Confidence ellipse for the sensory profiles obtained by Principal Components Analysis. *Food Quality and Preference*. 16 (3), 245-250.
Pages 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., Le S. & Pages 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

```
## Not run:
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=""))

## End(Not run)
```

---

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)

Francois Husson

References

Husson F., Le Dien S. & Pages J. (2005). Confidence ellipse for the sensory profiles obtained by Principal Components Analysis. *Food Quality and Preference*. 16 (3), 245-250.
Pages 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[as.numeric(sensochoc[,1])<11,]
Panel2=sensochoc[as.numeric(sensochoc[,1])<21 & as.numeric(sensochoc[,1])>10,]
Panel3=sensochoc[as.numeric(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)

F Husson, S Le

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 for the consumer, or, in the case of "qualified" categorization, to the sequence of words associted with the group to which the product belongs for the consumer.

Source

Applied Mathematics Department, Institut Agro Centre de Rennes

Examples

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

## End(Not run)
```

perfume_fcp *Perfume data obtained by free choice profiling*

Description

The data used here refer to 12 luxury perfumes described by 6 experts.

Usage

```
data(perfume_fcp)
```

Format

A data frame with 12 rows (the number of perfumes) and 47 columns.

Examples

```
## Not run:
data(perfume_fcp)
res <- fcp(perfume_fcp, group = c(12,7,7,7,6,8))

## End(Not run)
```

---

perfume_ideal *Perfume Ideal Data*

Description

The data used here refer to the sensory description of twelve perfumes (2 were duplicated).

Each perfume was evaluated once by 103 Dutch consumers and described on 21 attributes according to the Ideal Profile Method.
Both perceived and ideal intensities were asked. In addition, the overall liking is asked.

Usage

```
data(perfume_ideal)
```

Format

A data frame made of 103*14=1442 rows and 2 qualitative variables (set and product), 21*2 attributes (perceived and ideal intensities) and overall liking.

Source

OP&P Product Research, Utrecht, The Netherlands

Examples

```
## Not run:
data(perfume_ideal)
decat(perfume_ideal, formul = "~product+user", firstvar = 3, graph = FALSE)

## End(Not run)
```

plot.fahst *Make Factorial Approach for Hierarchical Sorting Task data (FAHST) graphs*

Description

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

Usage

```
## S3 method for class 'fahst'
plot(x,choix="ind", axes = c(1, 2), xlim = NULL, ylim = NULL,
   invisible = NULL, col.ind = "blue", col.var = "red", lab.ind=TRUE,lab.var=TRUE,
 cex = 1, lab.lev=TRUE,lab.grpe = TRUE, title = NULL, habillage = "none",
 habillage.lev = "none", traj = FALSE, palette = NULL, new.plot = TRUE, ...)
```

Arguments

x	an object of class fahst
choix	the graph to plot ("ind" for the products and the categories, "group" for the consumers and "level" for the levels)
axes	a length 2 vector specifying the components to plot
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'
invisible	string indicating if some points should not be drawn ("ind" or "var")
col.ind	a color for the products
col.var	a color for the categories
lab.ind	boolean, if TRUE, the products are labelled
lab.var	boolean, if TRUE, the categories associated with categorization are labelled
cex	cf. function par in the **graphics** package
lab.lev	boolean, if TRUE, the levels are labelled
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)
habillage	give no color for the individuals ("none"), or color the products according to one of the levels of a consumer (give the number of the colomn corresponding to the level)
habillage.lev	give no color for the levels ("none"), color the levels according to consumer ("subject") or color the levels according to the number of the level ("level")
traj	boolean, if TRUE, trajectories are drawn between levels of the same consumer

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, the levels factor map and the consumers factor map.

Author(s)

Marine Cadoret, Sebastien Le <sebastien.le@institut-agro.fr>

See Also

fahst

Examples

```
## Not run:
data(cards)
group<-c(2,3,3,2,2,4,2,3,2,1,3,2,3,3,3,2,3,3,2,3,3,3,3,3,3,3,3,3,3,3)
res.fahst <- fahst(cards,group,graph=FALSE)
plot.fahst(res.fahst,choix="ind",invisible="var",habillage=2,
    title="Cards colored according to level 2 of subject 1")
plot.fahst(res.fahst,choix="level",traj=TRUE)

## End(Not run)
```

---

plot.fasnt *Make Factorial Approach for Sorting Napping Task data (FASNT) graphs*

Description

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

Usage

```
## S3 method for class 'fasnt'
plot(x,choix="ind", axes = c(1, 2), xlim = NULL, ylim = NULL,
   invisible = NULL, col.ind = "blue", col.var = "red", lab.ind=TRUE,lab.var=TRUE,
 lab.coord=TRUE, lab.partial=TRUE, 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 and the categories, "group" for the consumers and "partial" for the partial products)
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
lab.ind	boolean, if TRUE, the products are labelled
lab.var	boolean, if TRUE, the categories associated with categorization are labelled
lab.coord	boolean, if TRUE, the napping variables are labelled
lab.partial	boolean, if TRUE, the partial products are labelled
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, the coordinates factor map and the consumers factor map.

Author(s)

Marine Cadoret, Sebastien Le <sebastien.le@institut-agro.fr>

See Also

fasnt

Examples

```
## Not run:
data(smoothies)
res.fasnt <- fasnt(smoothies, first="nappe",graph=FALSE)
plot.fasnt(res.fasnt,choix="ind",invisible="var",habillage=15,
  title="Objects colored according to the groups provided by consumer 5")
plot.fasnt(res.fasnt,choix="partial",lab.partial=FALSE)

## 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

```
## S3 method for class 'fast'
plot(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, Sebastien Le <sebastien.le@institut-agro.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)
```

---

plot.IdMap *Plot for the Ideal Mapping*

Description

Option to replot by zooming on the ideal map.

Usage

```
## S3 method for class 'IdMap'
plot(x, xlim, ylim, levels.contour=NULL, color=FALSE, inverse=FALSE, ...)
```

Arguments

x	An object of class *IdMap*
xlim	The lower and upper limits of interest on the X-axis
ylim	The lower and upper limits of interest on the Y-axis
levels.contour	The levels (between 0 and 1) to consider for the colors on the surface plot. By default, they are set automatically based on the results
color	Boolean, define whether the map is in color or in black and white
inverse	Boolean, define whether the Black/While code should be inversed in the graph
...	further arguments passed to or from other methods

Details

Replot the Ideal Map by zooming on the area of interest

Value

Return the IdMap zoomed on the area of interest.

Author(s)

Worch Thierry (thierry@qistatistics.co.uk)

See Also

IdMap

Examples

```
## Not run:
data(perfume_ideal)
res <- IdMap(perfume_ideal, col.p=2, col.j=1, col.lik=ncol(perfume_ideal), id.recogn="id_")
plot.IdMap(res, xlim=c(-7,10), ylim=c(-5,7), levels.contour=NULL, color=TRUE)
plot.IdMap(res, xlim=c(-7,10), ylim=c(-5,7), levels.contour=NULL, color=FALSE, inverse=TRUE)

## End(Not run)
```

---

plot.JAR *Make penalty graph*

Description

Plot the graphs for the penalty analysis.

Usage

```
## S3 method for class 'JAR'
plot(x, name.prod, model=1, confidence=TRUE, level=0.05, ...)
```

Arguments

x	an object of class fast
name.prod	a string corresonding to the name of the product for which penalty is drawn
model	an integer: 1 means that the one-dimensional penalties are drawn, 2 means that the multidimensional penalties are drawn
confidence	A boolean: if true, the confidence interval is drawn for each penalty
level	threshold for the selection of the categories that are plotted
...	further arguments passed to or from other methods, such as cex, cex.main, ...

Value

Draw a plot of the penalty against the ferquency of the categories.

Author(s)

Francois Husson

See Also

JAR

Examples

```
## Not run:
data(JAR)
res.jar <- JAR(JAR,col.p=13,col.j=1,col.pref=2)
plot.JAR(res.jar,name.prod="284", model=1)

## End(Not run)
```

---

plot.WordCountAna *Word-Count based methods Analysis (WordCountAna) graphs*

Description

Draw the Word-Count based methods Analysis (WordCountAna) graphs

Usage

```
## S3 method for class 'WordCountAna'
plot(x,axes=c(1,2),choix="prod",lab=TRUE,color=NULL,pch=NULL,
    proba=0.05,xlim=NULL,ylim=NULL,cex=1,title=NULL,new.plot=TRUE,...)
```

Arguments

x	an object of class VerbAna
axes	a length 2 vector specifying the components to plot
choix	a string corresponding to the graph that you want to do ("prod" for the products graph, "panel" for the panellists graph, "dist" for the distinct-words graph, "cons" for the consensual words graph)
lab	boolean, if TRUE, the labels are drawn
color	the color to use to draw the graph
pch	either an integer specifying a symbol or a single character to be used as the default in plotting points. See points for possible values and their interpretation
proba	significance threshold considered to define consensual words (by default 0.05)
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'
cex	numerical value giving the amount by which plotting text and symbols should be magnified relative to the default (by default 1)
title	string corresponding to the title of the graph you draw
new.plot	boolean, if TRUE, a new graphical device is created
...	further arguments passed to or from other methods, such as cex, cex.main, ...

Value

Returns one of the four following factor maps depending on the choice: products, panellists, distinct-words and consensual words.

Author(s)

Belchin Kostov <badriyan@clinic.ub.es>, Francois Husson <francois.husson@institut-agro.fr>, Monica Becue-Bertaut

References

Kostov, B., Becue-Bertaut, M., Husson, F., Pages, J., Cadoret, M., Torrens, J. and Urpi, P. (2012). A tool for detecting words with consensual meaning in verbalization tasks. 11th Sensometrics Conference, July 10-13, 2012, Rennes, France.

See Also

WordCountAna

Examples

```
data(perfume)
res<-WordCountAna(base=perfume,sep.word=";",graph=FALSE)
plot.WordCountAna(res,choix="prod")
plot.WordCountAna(res,choix="panel")
plot.WordCountAna(res,choix="dist")
```

```
plot.WordCountAna(res,choix="cons")
plot.WordCountAna(res,choix="cons",proba=0.1)
```

---

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,
  graph.type = c("ggplot","classic"))
```

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
graph.type	a character that gives the type of graph used: "ggplot" or "classic"

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)

Francois 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)

Francois Husson, Sebastien Le

References

Morand, E., Pages, 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)
dev.new()
pmfa(napping.don, napping.words)

## End(Not run)
```

`print.fahst` *Print Factorial Approach for Hierarchical Sorting Task data (FAHST) results*

Description

Print Factorial Approach for Hierarchical Sorting Task data (FAHST) results.

Usage

```
## S3 method for class 'fahst'
print(x, file = NULL, sep = ";", ...)
```

Arguments

x	an object of class fahst
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, Sebastien Le <sebastien.le@institut-agro.fr>

See Also

fahst

Examples

```
## Not run:
data(cards)
group<-c(2,3,3,2,2,4,2,3,2,1,3,2,3,3,3,2,3,3,2,3,3,3,3,3,3,3,3,3,3,3)
res.fast <- fahst(cards,group,graph=F)
print.fahst(res.fast, file="c:/fahst.csv", sep = ";")

## End(Not run)
```

`print.fasnt` *Print Factorial Approach for Sorting Napping Task data (FASNT) results*

Description

Print Factorial Approach for Sorting Napping Task data (FASNT) results.

Usage

```
## S3 method for class 'fasnt'
print(x, file = NULL, sep = ";", ...)
```

Arguments

`x`	an object of class fasnt
`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, Sebastien Le <sebastien.le@institut-agro.fr>

See Also

fasnt

Examples

```
## Not run:
data(smoothies)
res.fasnt <- fasnt(smoothies, first="nappe",graph=F)
print.fasnt(res.fasnt, file="c:/fasnt.csv", sep = ";")

## 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

```
## S3 method for class 'fast'
print(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, Sebastien Le <sebastien.le@institut-agro.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)

Francois 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)

Francois 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

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

Applied mathematics department, Institut Agro

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

Agrocampus Rennes

Examples

```
data(chocolates)
```

`smoothies` *Smoothies*

Description

The data used here refer to 8 smoothies on which 24 consumers performed a sorted napping task.

Usage

```
data(smoothies)
```

Format

A data frame with 8 rows (the number of smoothies) and 72 columns (the number of consumers * 3). For each consumer, we have the coordinates of the products on the tablecloth associated with napping on the one hand and the partitionning variable associated with categorization on the other hand. The columns are grouped by consumer.

Source

Applied mathematics department, Institut Agro

Examples

```
## Not run:
data(smoothies)
## Example of FASNT
res.fasnt<-fasnt(smoothies,first="nappe")

## End(Not run)
```

`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)

Francois 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)

Francois 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)

Francois 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)
```

---

videos *Videos data obtained with Holos*

Description

Example of Holos data, as obtained with the format_holos function.

Usage

```
data(videos)
```

Format

A list of 4 objects: name.subjects, a dataframe containing the concordance between the names of the subjects as given in the Holos experiment and their ID; datadigit, a list of S (S = number of subjects) dataframes corresponding to the digit-tracking data; datafinal_coord, a list of S (S = number of subjects) dataframes corresponding to the final configurations data; and datafinal_verb, a list of S (S = number of subjects) dataframes corresponding to the final verbalization data.

See Also

format_holos analyse_holos

Examples

```
data(videos)
```

`WilliamsDesign` *Construct a Williams design*

Description

Construct a Williams design

Usage

```
WilliamsDesign(nbprod,seed=NULL)
```

Arguments

`nbprod`	nombre de produits
`seed`	a number to intialize the generator of the random number

Value

Retrun the design

Author(s)

Francois Husson <husson@agrocampus-rennes.fr>

Examples

```
exemple1 <- WilliamsDesign(12)
```

`WordCountAna` *Word-Count based methods Analysis (WordCountAna)*

Description

Sensory methods as labelled sorting task, check-all-that-apply (CATA), ultra-flash profiling (UFP) and open-ended questions can be used to collect free-text descriptions of products through word-count based methods. A data frame with rows-products and columns-panellists is considered for the analysis. WordCountAna performs a multiple factor analysis for contingency tables keeping all the information in the comparison of the products. The identification of the consensual words which have the same meaning for most of the panellists eases the interpretation of the word-count based methods and solves the problems arising from the large diversity of vocabulary as the different meanings possibly associated to a same word. A test, based on resampling techniques, allows for assessing the significance of the consensus.

Usage

```
WordCountAna (base, sep.word = NULL, ncp = Inf, nb.panel = 3, nb.simul = 500,
    proba = 0.05, graph = TRUE, axes = c(1,2))
```

Arguments

base	a data frame with n rows (products) and p columns (panellists). Each cell corresponds to a free-text description used to describe a product by a panellist
sep.word	a string with all the characters which correspond to separator of words (by default, NULL and is considered equal to "; (),?./:'!$=+;<>[]@-")
ncp	number of dimensions kept in the results and to compute the within-inertia
nb.panel	minimum number of panellists who used the same word in order to define consensual words (by default 3)
nb.simul	number of bootstrap simulations (by default 500)
proba	significance threshold considered to define consensual words (by default 0.05)
graph	boolean, if TRUE a graph is displayed
axes	a length 2 vector specifying the components to plot

Value

mfact	a list of matrices containing all the results for multiple factor analysis for contingency tables
dist.words	a matrix containing the results for distinct words (number of times that used and number of panellists that pronounced)
centroids	a matrix containing the coordinates of the centroids of distinct-words
cons	a matrix containing the results of bootstrap resampling for distinct-words pronounced by at least "nb.panel" panellists (number of times that used, number of panellists that pronounced and the significance of the consensus)
cons.words	a vector of consensual words assessed by bootstrap resampling

Returns the products factor map, panellists factor map, distinct-words factor map and consensual words factor map.

Author(s)

Belchin Kostov <badriyan@clinic.ub.es>, Francois Husson <francois.husson@institut-agro.fr>, Monica Becue-Bertaut

References

Becue-Bertaut, M. and Pages, J. (2008). Multiple factor analysis and clustering of a mixture of quantitative, categorical and frequency data. *Computational Statistice and Data Analysis*, 52, 3255-3268. Kostov, B., Becue-Bertaut, M. and Husson, F. (2012). Multiple Factor Analysis for Contingency Tables in FactoMineR Package. *The R journal* Kostov, B., Becue-Bertaut, M., Husson, F., Pages, J., Cadoret, M., Torrens, J. and Urpi, P. (2012). A tool for detecting words with consensual meaning in verbalization tasks. 11th Sensometrics Conference, July 10-13, 2012, Rennes, France.

See Also

textual, MFA, plot.WordCountAna

Examples

```
data(perfume)
res<-WordCountAna(base=perfume,sep.word=";")
```

Index