Package: SPARTAAS (via r-universe)

August 21, 2024

Type Package

Title Statistical Pattern Recognition and daTing using Archaeological Artefacts assemblageS

Version 1.2.4

Maintainer Arthur Coulon <arthur-coulon@outlook.fr>

URL https://spartaas.gitpages.huma-num.fr/r-package/

Description Statistical pattern recognition and dating using archaeological artefacts assemblages. Package of statistical tools for archaeology. hclustcompro(perioclust): Bellanger Lise, Coulon Arthur, Husi Philibrary(SPARTlippe (2021, ISBN:978-3-030-60103-4). mapclust: Bellanger Lise, Coulon Arthur, Husi Philippe (2021) <doi:10.1016/j.jas.2021.105431>. seriograph: Desachy Bruno (2004) <doi:10.3406/pica.2004.2396>. cerardat: Bellanger Lise, Husi Philippe (2012) <doi:10.1016/j.jas.2011.06.031>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports FactoMineR, grDevices,dplyr,tidyr,ggplot2,plotly,stringr,colorspace, shiny,shinydashboard,shinyjs,shinyjqui,fpc,ggdendro,htmltools, rstudioapi,htmlwidgets,shinythemes,explor,shinyWidgets,scatterD3, ks,foreign,grid,cluster,leaflet,ape, MASS,ade4,lmtest,nor1mix,shinycssloaders,scales,fastcluster

Suggests knitr, rmarkdown,sf

RoxygenNote 7.1.1

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

RemoteUrl https://github.com/arliph/spartaas

RemoteRef HEAD

RemoteSha 7db7515aebe5129677d997539cd52ec3283a627b

Contents

`SPARTAAS-package` *SPARTAAS*

Description

Statistical **Pa**ttern **R**ecognition and da**T**ing using **A**rcheological **A**rtefacts assemblage**S**.

Details

Statistical pattern recognition and dating using archaeological artefacts assemblages. Package of statistical tools for archaeology. hclustcompro(perioclust): Bellanger Lise, Coulon Arthur, Husi Philibrary(SPARTlippe (2021, ISBN:978-3-030-60103-4). mapclust: Bellanger Lise, Coulon Arthur, Husi Philippe (2021) <doi:10.1016/j.jas.2021.105431>. seriograph: Desachy Bruno (2004) <doi:10.3406/pica.2004.2396>. cerardat: Bellanger Lise, Husi Philippe (2012) <doi:10.1016/j.jas.2011.06.031>.

Author(s)

Arthur Coulon [aut, cre], Lise Bellanger [aut], Philippe Husi [aut], Bruno Desachy [ctb], Benjamin Martineau [ctb]

Maintainer: Arthur Coulon <arthur-coulon@outlook.fr>

References

Bellanger L., Coulon A., Husi P., 2020 – Perioclust: a new Hierarchical agglomerative clustering method including temporal or spatial ordering constraints. Springer Series, Studies in Classification, Data Analysis, and Knowledge Organization. <doi: 10.1007/978-3-030-60104-1>

Bellanger L., Husi P., Laghzali Y. (2015). Spatial statistic analysis of dating using pottery: an aid to the characterization of cultural areas in West Central France. In : Traviglia A. ed., Across Space and Time, Proceedings of the 41th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA-2013), Perth (Australie), Amsterdam University Press : 276-282.

`adjacency` *Dissimilarity matrix based on connectivity information.*

Description

From the data of a network, we build a contiguity matrix. Based on this matrix, we generate a dissimilarity matrix. The matrix contains only 0 or 1, 1 if there is no relationship and 0 if there is a relationship. The network object is a data frame with two columns. The first column contains the elements of the network and the second column contains a list of all other elements related to it. The list is a string consisting of the names of the elements separated by commas (see example).

Usage

```
adjacency(network)
```

Arguments

`network`	Data frame with 2 columns. The first contains all elements (nodes) and the second a string with all related elements (links).

Value

`D`	Dissimilarity matrix based on adjacency.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--  or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data(datangkor)

## network stratigraphic data (Network)
network <- datangkor$stratigraphy

dissimilarity <- adjacency(network)
dissimilarity
```

`arrondi` *Returns the rounded value.*

Description

Always returns the upper value if the next digit is 5.

Usage

```
arrondi(x, acc)
```

Arguments

`x`	The number to round.
`acc`	Accuracy (number of digits). A negative number of digits means rounding to the power of ten, e.g. arrondi(x, digits = -2) will round to the nearest hundred.

Value

`res`	Value or vector of values rounded.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
  library(SPARTAAS)

  x1 <- c(15,25,35,45,55)
  round(x1,-1)
  arrondi(x1,-1)
```

```
x2 <- c(-15,-25,-35,-45,-55)
round(x2,-1)
arrondi(x2, -1)

x3 <- 1.125
round(x3,2)
arrondi(x3, 2)

x4 <- seq(-0.55,0.55,0.1)
data.frame(
  val = x4,
  round = round(x4,1),
  arrondi = arrondi(x4, 1),
  equal = (arrondi(x4, 1) == round(x4,1))
)
```

`CAdist` *Distance matrix based on correspondence analysis results*

Description

Perform a correspondence analysis on a contingency table and then return the distance matrix of the coordinates (you can choose the number of axes to use to build the distance matrix with the nCP parameter).

Usage

```
CAdist(df, nPC = NULL, graph = TRUE)
```

Arguments

`df` Data.frame, matrix or table with the data for the correspondence analysis.

`nPC` Number of principal components to be retained for the construction of the distance matrix. Must be between 1 and the minimum of ncol - 1 and nrow - 1. Could also be "max".

`graph` Logical parameter for plotting the Correspondence Analysis (Axis1, Axis2).

Value

`D` The distance matrix

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data(datangkor)

## contingency table
cont <- datangkor$contingency

distance <- CAdist(cont, nPC = "max")
distance

## run without printing the plot
distance <- CAdist(cont, nPC = "max", graph=FALSE)
```

`cerardat` *cerardat*

Description

The methodology is based on a statistical and visual approach using two estimated density curves to date each archaeological context. The statistical procedure required two steps, each leading to the construction of a density curve. The first allowed us to estimate a date corresponding to the terminus post quem of the context, a cursor reflecting an event dated in calendar time. This statistical tool allows the archaeologist to easily visualise and analyse chronological patterns.

Usage

```
cerardat(df, row.sup, date, nf = NULL, confidence = 0.95, graph = T)
```

Arguments

`df` The data (data.frame) is a contingency table with the observations in the rows and the technical groups in the columns.

`row.sup` Index of supplementary rows in df (vector).

`date` The dates of each observation or NA (vector).

`nf` an integer representing the number of axes retained in the correspondence analysis. If NULL, it is automatically chosen to keep a number corresponding to at least 60% of the inertia.

`confidence` The desired confidence interval (0.95 for 95%).

`graph` logical to display the plots or not.

Details

The corpus data is a contingency table with the observations in the rows and the technical groups in the columns. There are two types of observations: the reference corpus observations and the supplementary observations. The supplementary rows (observations) are identified by the argument ‘row.sup‘.

step 1: modelling events dated in calendar time (dateEv)
This step involves estimating the date of an event recorded in the ground (an archaeological context for the archaeologist) from the pottery assemblage of which it is composed, by fitting a regression model that relates a known date in calendar time, such as the date of issue of a coin, to its pottery profile. The reference corpus used to fit the regression model. We then used the previously fitted model to calculate a predicted value for contexts not included in the reference corpus, sometimes stratigraphically separated or poorly documented, with a 95% confidence interval for the predicted date.

A correspondence analysis (CA) was carried out to summarize the information in the reference corpus data. We then kept only the first factorial axes. In this way, our contingency table becomes a reduced size table, an incomplete reconstruction of the data. This principle is used in many factor analysis techniques to reduce the number of explanatory variables in the linear regression model.

After estimating the beta parameters of the model using the classical results of multiple regression analysis and checking that the model fits the data correctly, we can deduce the estimated date of an observation and also predict the date of another observation that has no coins and is therefore not dated.

step 2: from event time (dateEv) to accumulation time (dateAc)
We used the results of the first step and the properties of the CA to obtain an estimate of the date of each fabric. We could then define the archaeological time represented as dateAc, in other words the accumulation time of a context, as the weighted sum of the fabric dates; the weights being the proportions of MINVC of each fabric in the context. Assuming that the random variables dateEvj are independent, the distribution of the accumulation time of each context can be approximated by the Gaussian mixture. In this way, for each context, we obtained a plurimodal density curve representing the estimated law of accumulation time based on the mixture of normal densities (dates of each tissue). By definition, the area under the density curve has a value of 1 (i.e. 100%).

date
In order to estimate a date for the context, it is essential to refer to objects that have been dated by another source, such as coins. These contexts were selected on a very strict basis for their chronostratigraphic reliability, level of domestic occupation or enclosures with long urban stratigraphic sequences, thereby minimising any bias associated with the disparity between the date of the coin and that of the context.

Value

`prediction` Estimated date for archaeological context (event: dateEV and accumulation: dateAc) with confidence interval. The first two columns show the total count and the number of categories per line (only for columns used in CA). Then a date column shows the known dates. Each dateEv and dateAC model has three columns (value, lower bound, upper bound).

`date_gt` Estimated date for technical groups with confidence interval (use for dateAc). The first column show the total count per category in the reference data (only for rows used in CA).

`lm` Linear model on the components of the correspondance analysis.

`predict_obj_row`
date prediction of archaeological contexts (rows) using predict.lm.

`predict_obj_col`
date prediction of technical groups (columns) using predict.lm.

`cont_gt` Contingency table of the reference corpus.

`statistical.summary`
Statistical summary of the model:
Adjusted R-squared
R-squared
sigma (Residual standard error)
The Shapiro-Wilks test is used to verify the normality of the residuals.
The Durbin-Watson test checks for first order autocorrelation.
The Breusch-Pagan test checks for heteroscedasticity.

`obs_ca_eval` Quality of row representation in the correspondence analysis.

`check_ref` Plot of estimated dates (with confidence interval) and real dates of reference data. Only when the real date is known.

`check_sup` Plot of estimated dates (with confidence interval) and real dates of supplementary data. Only when the real date is known.

`Shapiro_Wilks` Summary of the Shapiro-Wilks test. see shapiro.test.

`Durbin_Watson` Summary of the Durbin-Watson test. see dwtest.

`Breusch_Pagan` Summary of the Breusch-Pagan test. see bptest.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

References

Bellanger L. and Husi P. (2012) Statistical tool for dating and interpreting archaeological contexts using pottery. Journal of Archaeological Science, Elsevier, 39 (4), pp.777-790. doi:10.1016/j.jas.2011.06.031.

Examples

```
data("datacerardat")

resultat = cerardat(df = datacerardat$df,
           row.sup = datacerardat$row.sup,
           date = datacerardat$date,
           nf = NULL,
           confidence = 0.95,
           graph = TRUE
        )

resultat
```

```
#The Shapiro-Wilks test is used to verify the normality of the residuals.
#The Durbin-Watson test checks for first order autocorrelation.
#The Breusch-Pagan test checks for heteroscedasticity.



#See the first plot
plot(resultat,
     which = 1,
     col1=rgb(0.93,0.23,0.23,0.5),
     col2="black",
     xlim=NULL,
     ylim=c(0,0.03)
    )

#See the first ten plots
#plot(resultat,
#     which = 1:10,
#     col1=rgb(0.93,0.23,0.23,0.5),
#     col2="black",
#     xlim=NULL,
#     ylim=c(0,0.03)
#    )

#See all plots
#plot(resultat,
#     which = NULL,
#     col1=rgb(0.93,0.23,0.23,0.5),
#     col2="black",
#     xlim=NULL,
#     ylim=c(0,0.03)
#    )

#You can extract the plots and find them in the directory :
paste0(getwd(),"/figures")
#With the 'extract_results' function
#extract_results(resultat,width=480, height=480, path="figures")
```

---

cerardat_app *Launch the shiny application.*

Description

see cerardat.

Usage

```
cerardat_app()
```

Value

No return value

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.

  library(SPARTAAS)
  if(interactive()){
    cerardat_app()
  }
```

cerardat_estim_nf *Number of axes to keep*

Description

Estimate the correct number of axes to keep in the regression model.

Usage

```
cerardat_estim_nf(df, row.sup, date)
```

Arguments

`df`	The data (data.frame) is a contingency table with the observations in rows and the technical groups in columns.
`row.sup`	Index of supplementary rows in df (vector).
`date`	The dates of each observation or NA (vector).

Value

`nf`	Number of axes to keep (minimal PRESS value)
`MSE`	plot of the Mean Squared Error.
`PRESS`	plot of the PRediction Error Sum Of Squares.
`Pvalue`	plot of p-values from statistical tests on the hypotheses.

`adj.R_sq` plot of the Coefficient of determination R^2.

`data` data frame of MSE, PRESS and R_sq values.

`data_stat` data frame of the p-values.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

References

Bellanger L. and Husi P. (2012) Statistical tool for dating and interpreting archaeological contexts using pottery. Journal of Archaeological Science, Elsevier, 39 (4), pp.777-790. doi:10.1016/j.jas.2011.06.031.

Examples

```
library(SPARTAAS)
data(datacerardat)
res = cerardat_estim_nf(datacerardat$df, datacerardat$row.sup, datacerardat$date)

#Number of axes to keep (minimal PRESS value)
res$nf

#the plots
res$MSE
res$PRESS
res$adj.R_sq
```

datacancer *Data set of cancerology.*

Description

Longitude, latitude, number of thyroid cancers. The data concern two French departments (Loire-Atlantique and Vendee) between 1998 and 2012. For confidentiality reasons, the data are simulated.

Usage

```
data("datacancer")
```

Format

List of two objects with 359 observations on the following 3 variables.

$coord (data.frame):

`longitude` a numeric vector: geographical coordinate

`latitude` a numeric vector: geographical coordinate

$var (vector):

`var` a numeric vector: number of thyroid cancers (simulated)

Author(s)

M. Karakachoff (IR CHU - l'institut du Thorax INSERM UMR 1087 - CNRS UMR 6291) Nantes, France

F. Molinie (resp. Loire-Atlantique-Vendee cancer registry - `registre-des-cancers`) France

Examples

```
library(SPARTAAS)
data(datacancer)
str(datacancer)
head(datacancer$coord)
str(datacancer$var)
```

`dataceram` *Data set of archeology*

Description

This important dataset comes from the Collective Research Project (CRP) on Medieval and Modern pottery in the Middle Loire Basin. This is a long-term project, since it began in 1996 and has already been the subject of two books on the subject (Husi dir. 2003 and 2013), as well as an online logical publication (Husi dir. 2022).

Usage

```
data("dataceram")
```

Format

List of three objects with 226 observations.

$contingency (data.frame) Contingency table of the quantities of 183 types of pottery sherds in the 226 sets.

$timerange (data.frame) The first column corresponds to the identifier (sets), the second and the third to the lower and upper limits of the temporal range of the observations.

$geographic_area Vector of the geographical area of each observation.

Author(s)

Husi P. (dir.) – La céramique médiévale et moderne du bassin de la Loire moyenne, chrono-typologie et transformation des aires culturelles dans la longue durée (6e —19e s.), XXe Supplément à la Revue Archéologique du Centre de la France, FERACF, Tours, 2022.

Examples

```
library(SPARTAAS)
data(dataceram)
str(dataceram)
str(dataceram$contingency)
head(dataceram$timerange)
head(dataceram$geographic_area)
```

`datacerardat` *Data set of archaeology.*

Description

Data sets on Tours pottery. Contains three objects: - Pottery sherd contingency table (data.frame 280 sets and 391 technical groups) - The date (if known) of the sets (vector) - Column index for supplementary sets (vector)

Usage

```
data("datacerardat")
```

Format

A list with 3 objects.

`df` Ceramic contingency table. a integer data.frame

`date` The date (if known) of the sets. a numeric or NA vector

`col.sup` Column index for supplementary sets. a numeric vector

Examples

```
data(datacerardat)
datacerardat
## maybe str(datacerardat);
```

datangkor *Data set of archeology*

Description

The archaeological data come from excavations carried out at Angkor Thom (Cambodia), the capital of the Khmer Empire between the 9th and 15th centuries (Gaucher, 2004). The dataset consists of the pottery assemblages (quantities of different types of pottery sherds contained in the sets -..$contingency) and the stratigrpahy of the sets from 3 disconnected archaeological sites (..$stratigraphy).

Usage

```
data("datangkor")
```

Format

List of two objects with 17 observations.

$contingency (data.frame) Contingency table of the quantities of 12 types of pottery sherds in the 17 sets.

$stratigraphy (data.frame) Saves the stratigraphic network. The first column corresponds to the nodes (sets) and the second to the edges by listing the nodes connected to it.

Author(s)

Gaucher, J. (2004). Angkor Thom, une utopie réalisée ? Structuration de l'espace et modèle indien d'urbanisme dans le Cambodge ancien. Arts Asiatiques, Volume 59, pp. 58-86.

Examples

```
library(SPARTAAS)
data(datangkor)
str(datangkor)
str(datangkor$contingency)
str(datangkor$stratigraphy)
```

datarcheo *Data set of archeology*

Description

Latitude, longitude, absolute difference between two dates and the name of the archaeological site. The data concern the dating of archaeological contexts in West-central France based on a large collection of medieval pottery finds. Two original statistical models are developed to estimate context dates using pottery. The absolute difference is calculated for each context. The data are based on a collective research on medieval pottery directed by P. Husi (`"La céramique médiévale dans la vallée de la Loire moyenne"`).

Usage

```
data("datarcheo")
```

Format

List of three objects with 240 observations on the following 4 variables.

$coord (data.frame):

`longitude` a numeric vector: geographical coordinate

`latitude` a numeric vector: geographical coordinate **$var (vector):**

`regionalized_var` a numeric vector: difference between two dating model **$label (vector):**

`noms` a character vector(factor): name of archeological site

Author(s)

P. Husi IR CNRS, UMR CITERES-LAT, CNRS/Tours University, France :

Examples

```
library(SPARTAAS)
data(datarcheo)
str(datarcheo)
head(datarcheo$coord)
str(datarcheo$var)
levels(datarcheo$label)
```

`extract_results` *Generates all plots (in jpeg format) in a subfolder*

Description

Generates all plots (in jpeg format) in a subfolder (relative path from the working directory).

Usage

```
extract_results(cerardat, width=480, height=480, path="figures",
                  col1 = rgb(0.93,0.23,0.23,0.5), col2 = "black",
                  xlim=NULL, ylim=NULL)
```

Arguments

`cerardat`	a cerardat output.
`width`	the width of the graphics region in pixels.
`height`	the width of the graphics region in pixels.
`path`	directory where plots are exported (relative path from the working directory). Automatically generates two sub-folders 'ref' for the reference plot and 'sup' for the supplementary plot.

`col1` Color of the the Event curve (color name, hex code, rgb function {grDevices} for transparency).

`col2` Color of the the Accumulation curve (color name, hex code, rgb function {grDevices} for transparency).

`xlim` Two numeric values, specifying the left limit and the right limit of the scale.

`ylim` Two numeric values, specifying the lower limit and the upper limit of the scale.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
data("datacerardat")

resultat = cerardat(df = datacerardat$df,
           row.sup = datacerardat$row.sup,
           date = datacerardat$date,
           nf = NULL,
           confidence = 0.95,
           graph = TRUE
        )
#Generates all plots (in jpeg format) in a folder 'figures'
#You will find the jpeg in 'ref' and 'sup' subfolder
#extract_results(resultat,width=480, height=480, path="figures",
#     col1=grDevices::rgb(0.93,0.23,0.23,0.5),
#     col2="black",
#     xlim=NULL,
#     ylim=c(0,0.03)
#    )

#You can extract the plots and find them in the directory :
paste0(getwd(),"/figures")
```

---

hclust *Hierarchical clustering of up to two datasets (Compromised clustering).*

Description

Overload of hclust for dealing with two dissimilarities matrices. Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

Usage

```
hclust(d, method = ”complete”, members = NULL, d2 = NULL, alpha = NULL)
```

Arguments

`d` a dissimilarity structure as produced by dist.

`method` the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

`members` NULL or a vector with length size of d. See the ‘Details’ section.

`d2` a second dissimilarity structure as produced by dist.

`alpha` The mixing parameter in order to generate the D_alpha matrix on which the classical hclust method is applied. Formula: D_alpha = alpha * d + (1-alpha) * d2.

Details

Data fusion (parameter alpha: optimal value see hclustcompro_select_alpha). It is necessary to define the appropriate proportion for each data source. This is the first sensitive point of the method that the user has to consider. A tool is provided to help him in his decision.

Value

hclust

Author(s)

The hclust function is based on Fortran code contributed to STATLIB by F. Murtagh.

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (d, method = ”complete”, members = NULL, d2 = NULL,
    alpha = NULL)
{
    if (!is.null(d2)) {
        if (!length(d) == length(d2)) {
            stop(”d and d2 have not the same size.”)
        }
        if (is.null(alpha)) {
            sa <- hclustcompro_select_alpha(d, d2, method = method,
                resampling = FALSE)
            alpha <- sa$alpha[1]
        }
        alpha <- as.numeric(alpha)
        if (!(alpha > 0 & alpha < 1)) {
            warning("Alpha must be between 0 and 1.")
            sa <- hclustcompro_select_alpha(d, d2, method = method,
                resampling = FALSE)
            alpha <- sa$alpha[1]
        }
        d <- dist(alpha * d + (1 - alpha) * d2)
    }
    stats::hclust(d, method, members)
  }
```

hclustcompro *hclustcompro*

Description

Compromised Hierarchical bottom-up clustering method. The method uses two sources of information. The merging of the two data sources is done by a parameter (α) that allows to weight each source.

$$D_\alpha = \alpha D_1 + (1-\alpha) D_2$$

Usage

```
hclustcompro(
  D1,
  D2,
  alpha="EstimateAlphaForMe",
  k=NULL,
  title="notitle",
  method="ward.D2",
  suppl_plot=TRUE
)
```

Arguments

`D1` First dissimilarity matrix (square matrix) or distance matrix. Could be a contingency table (see CAdist). A factorial correspondence analysis is performed using the distances (chi-square metric).

`D2` Second dissimilarity matrix (square matrix), same size as D1, or distance matrix.

`alpha` The mixing parameter in order to generate the D_alpha matrix (in [0;1]). Formula: D_alpha = alpha * D1 + (1-alpha) * D2 `k` The number of clusters you want.

`title` The title to be displayed on the dendogram plot.

`method` The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

`suppl_plot` Logical defines whether additional plots are to be displayed (WSS and average sil plot).

Details

CAH

Data fusion (parameter α optimal value see hclustcompro_select_alpha). It is necessary to define the appropriate proportion for each data source. This is the first sensitive point of the method that the user has to consider. A tool is provided to help him in his decision.

Cut dendrogram

The division into classes and subclasses is the second crucial point. It has to be done based on the knowledge of the study area and some decision support tools such as the cluster silhouette or the calculation of the intra-cluster variability (WSS: Within Sum of Square). You can use hclustcompro_subdivide to subdivide a cluster into sub-clusters.

Value

The function returns a list (class: hclustcompro_cl).

`D1` First dissimilarity matrix (square matrix)

`D2` Second dissimilarity matrix (square matrix)

`D_alpha` The matrix use in the CAH resulting from the mixing of the two matrices (D1 and D2)

`alpha` Alpha

`tree` An object of class hclust, describing the tree generated by the clustering process (see hclust)

`cluster` The cluster number vector of the selected partition

`cutree` Plot of the cut dendrogram

`call` How you call the function

`cont` Original contingency data (if D1 is a contingency table)

Author(s)

The hclust function is based on Fortran code contributed to STATLIB by F. Murtagh.

A. COULON

L. BELLANGER

P. HUSI

Examples

```
library(SPARTAAS)
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
constraint <- adjacency(network)

#You can also run hclustcompro with the dist matrix directly
hclustcompro(D1 = distance, D2 = constraint, alpha = 0.7, k = 4)
```

hclustcompro_app *Launch the shiny application.*

Description

see hclustcompro, hclustcompro_select_alpha, seriograph. You can also check the wiki on the application.

Usage

```
hclustcompro_app()
```

Value

No return value

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.

library(SPARTAAS)
if(interactive()){
  hclustcompro_app()
}
```

hclustcompro_detail_resampling

Resampling process in detail (one curve by set of clone).

Description

Based on a resampling process, we generate clone and we check the CorCrit_alpha. The function shows each set of clone curves.

Usage

```
hclustcompro_detail_resampling(D1, D2 = NULL, acc = 2, method = "ward.D2", iter = 5)
```

Arguments

`D1`	First dissimilarity matrix or contingency table (square matrix). You can replace D1 by a hclustcompro object (Don't use D2 in this case).
`D2`	Second dissimilarity matrix or network data (square matrix) same size than D1. If D1 is a hclustcompro object D2 is set to NULL.
`acc`	Number of digits after the comma for the alpha value.
`method`	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
`iter`	The number of clones cheked for each observation.

Details

Definition of the criterion:

A criterion for choosing alpha IN [0;1] must be determined by balancing the weights between the two information sources in the final classification. To obtain alpha, we define the following criterion:

$$CorCrit_alpha = |Cor(dist_cophenetic, D1) - Cor(dist_cophenetic, D2)|$$

$$equation(1)$$

The CorCrit_alpha criterium in (1) represents the difference in absolute value between two cophenetic correlation (Cophenetic correlation is defined as the correlation between two distances matrices. It is calculated by considering the half distances matrices as vectors. It measures of how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points).

The first correlation is associated with the comparison between D1 and ultrametric distances from the HAC with alpha fixed; while the second compares D2 and ultrametric distances from the HAC with alpha fixed. Then, in order to compromise between the information provided by D1 and D2, we decided to estimate alpha with hat(alpha) such that:

$$hat(alpha) = minCorCrit_alpha$$

$$equation(2)$$

Resampling strategy:

To do this, a set of "clones" is created for each observation i. A clone c of observation i is a copy of observation i for which the adjacency relationships to others have been modified. The clone has none conection exept with j. A set is generated by varying j for all observations except i. A HAC is then carried out using the combination defined in (1) with D1(c) a (n+1)X(n+1) matrix where the observations i and c are identical and D2(c) a (n+1)X(n+1) matrix where the clone c of i has different neighbourhood relationships from those of i. We can create a set of n-1 clones for each element i in n, so n(n-1) clones in total.

Intuitively, by varying alpha between 0 and 1, we will be able to identify when the clone and the initial observation will be separated on the dendrogram. This moment will correspond to the value of alpha above which the weight given to information on the connection between observations contained in D2 has too much impact on the results compared to that of D1.

Let CorCrit_alpha(c) defines the same criterion as in (1) in which D1 ans D2 are replaced respectively by D1(c) and D2(c). The estimated alpha is the average of estimated values for each clone. For each clone (c):

$$hat(alpha)(c) = minCorCrit_alpha(c)$$

$$equation(3)$$

hat(alpha)^* is the average of the hat(alpha)(c). In the same spirit as confidence intervals based on bootstrap percentiles (Efron & Tibshirani, 1993), a percentile confidence interval based on replication is also be obtained using the empirical percentiles of the distribution of hat(alpha)(c).

$$hat(alpha)* = (1/n(n-1)) * sumhat(alpha)(c)$$

$$equation(4)$$

$$cIN[1; n(n-1)].$$

Value

`plot`	The interactive plot: CorCrit_alpha criterium for each resampling dataset

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
###################################
#      For view the equation      #
###################################

plot(
  c(.6,.6,.6,.6),
  c(.9,.5,-.3,-.7),
  xlim = c(.6,1.4),
  ylim = c(-1.1,1),
  axes = FALSE,
  main = "Equations:",
  xlab = "",
  ylab = "",
  pch = 1
)
text(.65, .9, "( 1 )")
text(.65, .5, "( 2 )")
text(.65,-.3, "( 3 )")
text(.65,-.7, "( 4 )")

text(1, .9,
 expression(CorCrit[alpha] == abs(Cor(dist[cophenetic],dist[ceramic]) - Cor(dist[cophenetic],
  dist[stratigraphic])
)))
text(1, .5, expression(hat(alpha) == min(CorCrit[alpha], alpha)))

text(1,-.3, expression(hat(alpha)^(c) == min(CorCrit[alpha]^(c), alpha)))
text(1,-.7, expression(hat(alpha)^"*" == frac(1,n(n-1)) * sum(hat(alpha)^(c),c==1,n(n-1))))

#################################


##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.
#library(SPARTAAS)
#data(datangkor)

#network stratigraphic data (Network)
#network <- datangkor$stratigraphy

#contingency table
#cont <- datangkor$contingency


#hclustcompro_detail_resampling(D1 = CAdist(cont, nPC="max"), D2 = adjacency(network))
```

hclustcompro_select_alpha

Estimate the optimal value(s) of the α parameter.

Description

The following criterion "balances" the weight of D_1 and D_2 in the final clustering. The α value is only a point estimate but the confidence interval gives a range of possible values.

Based on a resampling process, we generate clones and recalculate the criteria according to α (see below).

Usage

```
hclustcompro_select_alpha(
    D1,
    D2,
    acc=2,
    resampling=TRUE,
    method="ward.D2",
    iter=5,
    suppl_plot=TRUE
)
```

Arguments

`D1`	First dissimilarity matrix or contingency table (square matrix).
`D2`	Second dissimilarity matrix or network data (square matrix) of the same size as D1.
`acc`	Number of digits after the decimal point for the alpha value.
`resampling`	Logical for estimating the confidence interval with a resampling strategy. If you have a lot of data, you can save computation time by setting this option to FALSE.
`method`	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
`iter`	The number of clones checked per observation. (200 observations iter=1: ~30 sec, 1000 observations iter=1: ~40 min).
`suppl_plot`	Logical defines whether additional plots should be displayed.

Details

Definition of the criterion:

A criterion for choosing $\alpha \in [0;1]$ must be determined by balancing the weights between the two sources of information in the final classification. To obtain α, we define the following criterion

$$CorCrit_{\alpha} = |Cor(dist_{cophenetic}, D_1) - Cor(dist_{cophenetic}, D_2)|$$

$$Equation(1)$$

The $CorCrit_{\alpha}$ criterion in (1) represents the difference in absolute value between two cophenetic correlations (cophenetic correlation is defined as the correlation between two distance matrices. It is calculated by considering the half distance matrices as vectors. It measures how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points). The first correlation is related to the comparison between D_1 and the ultrametric distances of the clustering with α fixed, while the second compares D_2 and the ultrametric distances of the clustering with α fixed. Then, in order to compromise between the information provided by D_1 and D_2, we decided to estimate α with $\hat{\alpha}$ such that:

$$\hat{\alpha} = minCorCrit_{\alpha}$$

$$Equation(2)$$

Resampling strategy:

This is done by creating a set of "clones" for each observation i. A clone c of observation i is a copy of observation i for which the distances from the second source have been modified. The modification is made by copying the distances for the second source from another observation j. A clustering is then performed using the combination defined in (1) with $D_1^{(c)}$ an $(n+1) \times (n+1)$ matrix where observations i and c are identical and $D_2^{(c)}$ an $(n+1) \times (n+1)$ matrix where the clone c of i has different distances from those of i. A set of clones is generated by varying j for all observations except i. We can generate a set of $n-1$ clones for each element i in n, so $n(n-1)$ clones in total.

Intuitively, by varying α between 0 and 1, we will be able to identify when the clone and the original observation are separated on the dendrogram. This moment will correspond to the value of alpha above which the weight given to the information about the connection between observations contained in D_2 has too much influence on the results compared to that of D_1.

Let $CorCrit_{\alpha}^{(c)}$ define the same criterion as in (1), where D_1 and D_2 are replaced by $D_1^{(c)}$ and $D_2^{(c)}$ respectively. The estimated α is the mean of the estimated values for each clone.
For each clone c:

$$\hat{\alpha}^{(c)} = minCorCrit_{\alpha}^{(c)}$$

$$Equation(3)$$

$\hat{\alpha}^{*}$ is the mean of $\hat{\alpha}^{(c)}$. In the same spirit as confidence intervals based on bootstrap percentiles (Efron & Tibshirani, 1993), a percentile confidence interval based on replication is also be obtained using the empirical percentiles of the distribution of $\hat{\alpha}^{(c)}$.

$$\hat{\alpha}^{*} = \frac{1}{n(n-1)} \sum \hat{\alpha}^{(c)}$$

$$Equation(4)$$

$$c \in [1; n(n-1)]$$

Warnings:

It is possible to observe an α value outside the confidence interval. In some cases, this problem can be solved by increasing the number of iterations or by changing the number of axes used to construct the matrix D1 after the correspondence analysis. If the α value remains outside the interval, it means that the data are noisy and the resampling procedure is affected.

Value

The function returns a list (class: selectAlpha_obj).

`alpha`	The estimated value of the alpha parameter (min CorCrit_alpha)
`alpha.plot`	The CorCrit curve for alpha between 0 and 1

If resampling = TRUE

`sd`	The standard deviation
`conf`	The confidence interval of alpha
`boxplot`	The boxplot of alpha estimation with resampling
`values`	All potential alpha values obtained from each clone

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
#################################

##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

dissimilarity <- CAdist(cont,nPC="max",graph=FALSE)
constraint <- adjacency(network)

hclustcompro_select_alpha(D1 = dissimilarity, D2 = constraint)
hclustcompro_select_alpha(D1 = dissimilarity, D2 = constraint, acc = 3, resampling = TRUE)
```

hclustcompro_subdivide

Subdivide a cluster after running hclustcompro.

Description

Allow the user to split one cluster into sub-clusters. This function only works with 'hclustcompro_cl' object returned by the hclustcompro function.

Usage

```
hclustcompro_subdivide(hclustcompro_cl,cluster,nb_class)
```

Arguments

`hclustcompro_cl`	A hclustcompro_cl object
`cluster`	The number of the cluster. Numbered from left to right on the dendrogram (1, 2, ...)
`nb_class`	The number of sub-clusters you want

Value

`hclustcompro_cl`	A new hclustcompro_cl object updated see hclustcompro

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data(datangkor)

#network stratigraphic data (Network)
network <- datangkor$stratigraphy

#contingency table
cont <- datangkor$contingency

#obtain the dissimilarities matrices
distance <- CAdist(cont, nPC = 11)
```

```
constraint <- adjacency(network)

#You can also run hclustcompro with the dist matrix directly
clustering <- hclustcompro(D1 = distance, D2 = constraint, alpha = 0.7, k = 7) #number of cluster 7
clustering <- hclustcompro_subdivide(clustering,cluster = 5, nb_class = 2)

#subdivide more than one cluster
clustering2 <- hclustcompro(D1 = distance, D2 = constraint,0.7,k=7) #number of cluster 7
clustering2 <- hclustcompro_subdivide(clustering2,cluster = c(5,7), nb_class = c(2,2))
```

mapclust *Divise hierarchical Clustering using Spatialpatches algorithm.*

Description

This function performs a divisive hierarchical clustering on a regionalised variable using the spatial patches algorithm (Woillez et al. 2007; Woillez, Rivoirard and Petitgas 2009). It is a top-down hierarchical clustering with a geographical constraint. It is possible to cut the tree by clicking on the dendrogram at the desired level. The results include a description of the clusters and graphics. When slicing the dendrogram, you can look at the two plots (WSSPlot and AveSilPlot) that show the relatively good quality of the partitions. The first shows the Within Sum of Square (WSS) for each partition and you can use the elbow approach to select a partition. The second graph shows the average silhouette width. This index is between -1 and 1. The closer it is to 1, the better the partition. See silhouette.

If you want to cut the dendogram to a different dlim or number of clusters, you can do so without re-running `mapclust()` with `mapclust_cut_tree`.

Usage

```
mapclust(
    coord,
    var,
    label = "Nolabel",
    iter = 20,
    Plabel = TRUE,
    lonlat = TRUE,
    positive_var = FALSE,
    n = NULL
)
```

Arguments

`coord` The x (longitude) and y (latitude) coordinates of the data.frame or matrix dimension 2.

`var` The regionalised variable(s) of interest

`label` (optional) The names of the samples or an identifier. Must be a factor.

`iter` The number of iterations. The number of different dlim you want to test (must be greater than 10).

`Plabel` Logical parameter to activate or not the printing of labels on the dendrogram.

`lonlat` Logical parameter to activate or not the cartography in lonlat system with leaflet (base map).

`positive_var` logical parameter that defines whether your variable of interest is positive or not.

`n` Number of groups. If NULL, you can select the number of groups by clicking on the dendrogram.

Details

Dlim is the selected minimum distance from the sample to the patch centre: to identify patches (units are those of coordinates). The dlim is automatically initialised and does not need to be set by the user. The minimum data is a data frame or matrix with at least 3 columns.

Value

the function returns a list.

Plot:

`dendrogram` The global dendrogram (hclust object)

`dendrogram_ggplot` The global dendrogram (ggplot2 object)

`cuttree` The cut dendrogram

`map` The map of the selected partition

`AveSilPlot` The average silhouette width plot (for each partition)

`WSSPlot` The Within Sum of Square plot (for each partition)

`silhouette` The silhouette plot of the selected partition

Value:

`X` The x-coordinate data you have used

`Y` The y-coordinate data you have used

`var` The regionalized variable(s) data you have used

`label` The label vector you have used

`density` The estimated density based on var. Equal to var if you used a unidimensional density variable.

`cluster` The cluster vector of the selected partition

`Plabel` Logical parameter to activate or not the printing of labels on the dendrogram

`fullhist` The compositional cluster for each observation

`hist` The compositional cluster without duplicates (corresponds to the split on the dendogram)

`dlim` The vector of the different limit distances

`cutdlim` The select dlim for the cut of the current partition

DiMatrix The weighted euclidean distance matrix

silhouetteData The silhouette data of the selected partition

AveSilData The average silhouette value for each partition

Moran The Moran index for each group for each partition

lonlat Logical parameter, whether your coordinates are in latitude-longitude format or not

Author(s)

A. COULON L. BELLANGER P. HUSI

References

Bellanger L., Coulon A. and Husi P. (2021) Determination of cultural areas based on medieval pottery using an original divisive hierarchical clustering method with geographical constraint (MapClust), Journal of Archaeological Science, Volume 132 doi:10.1016/j.jas.2021.105431.

Bellanger L., Husi P., Laghzali Y. (2015). Spatial statistic analysis of dating using pottery: an aid to the characterization of cultural areas in West Central France. In : Traviglia A. ed., Across Space and Time, Proceedings of the 41th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA-2013), Perth (Australie), Amsterdam University Press : 276-282.

Woillez M., Poulard J.C., Rivoirard J., Petitgas P., Bez N. (2007). Indices for capturing spatial patterns and their evolution in time, with application to European hake (Merluccius merluccius) in the Bay of Biscay. ICES J. Mar. Sci. 64, 537-550.

Woillez M., Rivoirard J. and Petitgas P. (2009) Notes on survey-based spatial indicators for monitoring fish populations, Aquatic Living Resources, 22 :155-164.

Examples

```
###################################
## loading data
 library(SPARTAAS)
 data(datarcheo)
 data(datacancer)

###################################
### Example: 1
## Function "mapclust"
# object <- mapclust( coord = ..., var = ..., label = ...)

classification <- mapclust(datarcheo$coord, datarcheo$var, datarcheo$label, n=4)

#Global dendrogram
 classification$dendrogram
#Cut dendrogram
 classification$cuttree
#silhouette of selected partition
 classification$silhouette
```

```
#You can cut the dendrogram for another dlim
 NewCut <- mapclust_cut_tree(classification, dlim=0.30)

#See evaluation using Silhouette width by running:
 NewCut$silhouette
 #If the plot is empty try to increase the height of the window (full screen)

#See summary of the data by running:
 summary(NewCut$silhouetteData)


####################################
## kmeans comparison
# pepare data (only geographical data)
 datakmeans <- datarcheo$coord

#kmeans
 number_cluster <- 4
 cl <- kmeans(datakmeans, number_cluster)
 plot(datakmeans, col = cl$cluster)
```

`mapclust_app` *Shiny application for Mapclust method*

Description

see mapclust You can also check the wiki on the application.

Usage

```
mapclust_app()
```

Value

No return value

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
#open application
#library(SPARTAAS)
#if(interactive()){
  #mapclust_app()
#}
```

`mapclust_cut_tree` *Function to cut the dendrogram for a new height (distance limit) or a new number of clusters and map the new partition*

Description

The function returns the new map, one dendrogram with the cut line, the silhouette width and the new vector cluster. You must call mapclust beforehand to get a `mapclust_cl` object that can be used by mapclust_cut_tree.

Usage

```
mapclust_cut_tree(classification, nb_grp = NA, dlim = NA)
```

Arguments

`classification`	The object return by mapclust `Class: mapclust_cl`
`nb_grp`	The number of groups you want to have on the partition. Must be an integer. (don't use dlim in this case)
`dlim`	The value of dlim at which you wish to cut the dendrogram. You can enter any value (numerical) and the function will select the nearest lower dlim with the same partition. (do not use nb_grp in this case).

Value

the function returns a list.

Plot:

`dendrogram`	The global dendrogram (hclust object)
`dendrogram_ggplot`	The global dendrogram (ggplot2 object)
`cuttree`	The cut dendrogram
`map`	The map of the selected partition
`AveSilPlot`	The average silhouette width plot (for each partition)
`WSSPlot`	The Within Sum of Square plot (for each partition)
`silhouette`	The silhouette plot of the selected partition

Value:

`X` The x-coordinate data you have used

`Y` The y-coordinate data you have used

`var` The regionalised variable data you have used

`label` The label vector you have used

`density` The estimated density based on var. Equal to var if you used a unidimensional density variable

`cluster` The cluster vector of the selected partition

`Plabel` Logical parameter to activate or not the printing of labels on the dendrogram

`fullhist` The compositional cluster for each observation

`hist` The compositional cluster without duplicates (corresponds to the split on the dendogram)

`dlim` The vector of the different limit distances

`cutdlim` The select dlim for the cut of the current partition

`DiMatrix` The weighted euclidean distance matrix

`silhouetteData` The silhouette data of the selected partition

`AveSilData` The average silhouette value for each partition

`Moran` The Moran index for each group for each partition

`lonlat` Logical parameter, whether your coordinates are in latitude-longitude format or not

Author(s)

A. COULON L. BELLANGER P. HUSI

Examples

```
## loading data
library(SPARTAAS)
data(datarcheo)

##First you need to run the mapclust function.
#This function allow you to obtain one partition
# object <- mapclust( coord = ..., var = ..., label = ...)
OldClassif <- mapclust(datarcheo$coord, datarcheo$var, datarcheo$label, n = 4)

##In order to cut the dendrogram for another dlim
NewCut <- mapclust_cut_tree(classification = OldClassif, dlim = 0.37)
##In order to cut the dendrogram for another number of cluster
NewCut2 <- mapclust_cut_tree(classification = OldClassif, nb_grp = 4)

#See evaluation using Silhouette width by running:
NewCut$silhouette
#If the plot is empty try to increase the height of the window (full screen).

#See summary of the data by running:
summary(NewCut$silhouetteData)
```

`overlap` *Temporal overlap index*

Description

The overlap index is the ratio between internal overlap and total overlap over time. We define the limit of total overlap as: the minimum of the lower limits of the pair of individuals and the maximum of the upper limits. We define the internal overlap limit as the maximum of the lower limits and the minimum of the upper limits.

Usage

```
overlap(temporal)
```

Arguments

`temporal`	A data frame with tree columns: the name of the element, the lower limit and the upper limit.

Details

The lower and upper limits must be numeric.

The dissimilarity between time periods is calculated as the ratio of the overlap of the time periods (distance in the case of disjoint time periods) to the cumulative extent of the two time periods.

As the ratio is bounded between -1 and 1, we add 1 to make it positive and normalise it so that it is between 0 and 1.

This overlap index then needs to be transformed into a dissimilarity index between sets. To do this we use the 1 - ratio. It is equal to 0 if the two time periods are identical and 1 if they are infinitely different.

Value

`D`	The dissimilarity matrix base on the overlap index.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data(dataceram)
dist <- overlap(dataceram$timerange)
```

---

plot.cerardat_obj *plot cerardat model*

Description

plot cerardat model

Usage

```
## S3 method for class 'cerardat_obj'
plot(x, which = NULL,
                    col1 = rgb(0.93,0.23,0.23,0.5), col2 = "black",
                    xlim=NULL, ylim=NULL,...)
```

Arguments

`x` a cerardat output.

`which` Vector containing the plots you want (row number on the contingence table: numeric).

`col1` Color of the Event curve (color name, hex code, rgb function {grDevices} for transparency).

`col2` Color of the Accumulation curve (color name, hex code, rgb function {grDevices} for transparency).

`xlim` Two numeric values, specifying the left limit and the right limit of the scale.

`ylim` Two numeric values, specifying the lower limit and the upper limit of the scale.

`...` other parameters to be passed through to plotting functions.

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.

## The function is currently defined as
```

---

seriograph *Plot seriograph (B. DESACHY).*

Description

Visualization of contingency data over time. **Rows** must be individuals (archaeological site,...) and **columns** must be categories (type,...).

Usage

```
seriograph(cont, order, insert, show, permute)
```

Arguments

`cont`	Contingency table or hclustcompro object. Note: Your contingency table must have the rows sorted in chronological order (the order parameter allows you to change the order of the rows if necessary).
`order`	Vector to change the order of the rows (use row names or cluster names if cont is a hclustcompro object, as a character vector). The oldest (at the bottom) must be at the end of the vector. Missing names are not plotted. You can remove a row by simply removing the name in the vector.
`show`	The element to be plotted. This should be (a unique abbreviation of) one of 'both', 'EPPM' or 'frequency'.
`permute`	Logical for permute columns in order to show seriation.
`insert`	Vector with the position after which you want to insert one or more hiatuses. Could be a list with two vectors: position and label to be printed instead of hiatus (see last examples).

Details

Seriograph

We have chosen the serigraph (DESACHY 2004). This tool makes it possible to highlight the evolution of ceramics over time as well as to understand the commercial relations thanks to the imported ceramics. The percentages of each category of ceramics per set are displayed. The percentages are calculated independently for each set (row). The display of the percentages allows comparison of the different sets but does not provide information on the differences in numbers.

To fill this gap, the proportion of the numbers in each class is displayed on the seriograph (weight column).

We can generalized this representation for other contingency data or with hclustcompro object.

The visualization of positive deviations from the average percentage allows us to observe a series that results from changes in techniques and materials dedicated to ceramic manufacturing over time.

In order to facilitate the exploitation of the data tables, we propose here a computerised graphic processing tool (EPPM serigraph - for Ecart Positif aux Pourcentages Moyens - positive deviation from the average percentage), which does not require specialised statistical skills and is adapted to the case of stratified sites, where the study of the evolution of artefacts can be based on the relative chronology provided by the excavation.

The treatment consists firstly of transforming this table of counts into a table of percentages, the total number in each set (each row) being reduced to 100; these are the proportions, or frequencies, of the types in the sets are thus compared.

The display of positive deviations from the mean percentages (EPPM) shows in black on a lighter background the percentage shares that are higher than the mean percentage of the variable, so as to highlight the most significant part of the values in the table.This display is simply adapted to the seriograph: when a percentage is greater than the average percentage of the type, the excess share (called here EPPM: positive deviation from the average percentage) is shown in black, centred around the axis of the type, on the grey background of the percentage bar.

The table is then transformed into a graphic matrix where these percentages are expressed, for each type, by horizontal bars centred on the column axis. When the rows are ordered chronologically, the silhouette formed by the superposition of these frequency bars bars makes it possible to visualise the evolution over time of the type concerned.

The display of the percentages allows comparison of the different sets but does not provide information on the differences in numbers. To fill this gap, the proportion of the numbers in each class is displayed on the seriograph (weight column).

The processing technique applies to sets whose chronological order is not known; the lines of the graph are to be reorganised so as to obtain boat-shaped silhouettes following the hypothesis of a chronological evolution corresponding to the seriation model.

Positive deviation from the average percentage (EPPM in French)

The average percentage is calculated for each ceramic category (columns) on the total number of accounts (all classes combined). From the average percentage we recover for each category and for each rows the difference between the percentage of the category in the class with the average percentage. The EPPM corresponds to the notion of independence deviation (between rows and columns, between categories and time classes) in a chi-square test approach. Although this approach is fundamental in statistical analysis, independence deviations are here purely indicative and are not associated with a p_value that could determine the significance of deviations.

Weight

Weight is the number of observations divided by the total number of observations. It indicates for each row the percentage of the data set used to calculate the frequencies of the elements (row).

Permutation

order argument:

The rows of the contingency table are initially in the order of appearance (from top to bottom). It must be possible to re-order the classes in a temporal way (You can also order as you want your contingency table).

permute argument:
In addition, it is possible to swap ceramic categories (contingency table columns) in order to highlight a serialization phenomenon. Matrix permutation uses an algorithm called "reciprocal averages". Each line is assigned a rank ranging from 1 to n the number of lines. A barycentre is calculated for each column by weighting according to the row rank. Finally, the columns are reorganized by sorting them by their barycentre.

Insert

It's possible to insert a row in the seriograph in order to represent a archeological hiatus or other temporal discontinuities.

Value

The function returns a list (class: seriograph).

`seriograph` The seriograph plot

`dendrogram` If cont is a hclustcompro object return the dendrogram with the period order as label

`contingency` Data frame of the contingencies data group by cluster

`frequency` Data frame of the frequencies data group by cluster

`ecart` Data frame of the gap data group by cluster

Author(s)

B. DESACHY

A. COULON

L. BELLANGER

P. HUSI

References

Desachy B. (2004). Le sériographe EPPM : un outil informatisé de sériation graphique pour tableaux de comptages. In: Revue archéologique de Picardie, n°3-4, 2004. Céramiques domestiques et terres cuites architecturales. Actes des journées d'étude d'Amiens (2001-2002-2003) pp. 39-56 doi:10.3406/pica.2004.2396

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data(datangkor)
```

```
## network stratigraphic data (Network)
network <- datangkor$stratigraphy

## contingency table
cont <- datangkor$contingency

## default
seriograph(cont)

seriograph(cont,show = "EPPM")
seriograph(cont,show = "frequency")

## Don't allow permutation of columns
seriograph(cont, permute = FALSE)

## insert Hiatus (position, 1 -> after first row from bottom: oldest)
seriograph(cont,insert = 2)
seriograph(cont,insert = c(2,3))

## insert costum label element
insert <- list(
  position = c(2,3),
  label = c("Hiatus.100years","Missing data")
)
seriograph(cont,insert = insert)

## change order with cluster name (letters on dendrogram) to sort them in a chronological order
seriograph(cont,order=c("AI03","AI09","AI01","AI02","AI04","APQR01","AO05",
"AO03","AI05","AO01","APQR02","AI07","AI08","AO02","AI06","AO04","APQR03"))
## by omitting the row names, you delete the corresponding rows
seriograph(cont,order=c("AI02","AI08","APQR03","AI09"))
```

serio_app *Launch the shiny application.*

Description

see seriograph. You can also check the wiki on the application.

Usage

```
serio_app()
```

Value

No return value

Author(s)

A. COULON

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.

library(SPARTAAS)
if(interactive()){
  serio_app()
}
```

timerange — *Plot the time range of observations sorted by cluster.*

Description

Cluster time range visualisation.

Usage

```
timerange(
  data,
  cluster = NULL,
  add = NULL,
  density = NULL,
  color = NULL,
  reorder = FALSE
)
```

Arguments

`data`	data.frame containing the identifier, lower and upper bound (id, inf, sup) for each observation
`cluster`	vector containing the cluster number of each observation.
`add`	data.frame containing the information to be displayed on hover.
`density`	vector of the density for each observation.
`color`	vector of the colors for each observation (if you want the colors to correspond to something else than clusters). Character vector of the same length as the number of observations.

`reorder`	Logical to rearrange the colors. If TRUE, the first color corresponds to the leftmost cluster on the plot. If FALSE, the first color is that of cluster number 1, wherever it is.

Value

The function returns a list.

`plot`	The timerange plot.

Author(s)

A. COULON

B. MARTINEAU

L. BELLANGER

P. HUSI

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##-- or do  help(data=index)  for the standard data sets.
library(SPARTAAS)
data <- data.frame(
  id = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),
  Lower_bound = c(400,401,401,350,500,460,482,432,399,
    489,750,740,704,700,758,789,802,755,750,820),
  Upper_bound = c(550,689,755,700,780,700,700,699,650,
    850,1100,1100,1010,889,999,999,1050,1002,1000,1100)
)

cluster = c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2)

add <- data.frame(
  Site = c("Angers","Angers","Angers","Angers","Angers",
    "Angers","Angers","Angers","Angers","Angers",
    "Blois","Blois","Blois","Blois","Blois",
    "Blois","Blois","Blois","Blois","Blois")
)

timerange(data, cluster, add)

## with sub group (cluster 1 is sub divided in 2: 1.1 and 1.2)
cluster_with_sub = c(1.1,1.1,1.1,1.1,1.1,1.2,1.2,1.2,1.2,1.2,2,2,2,2,2,2,2,2,2,2)

timerange(data, cluster_with_sub, add)

## with density
density <- c(32,34,35,19,9,25,19,29,28,18,10,13,9,10,9,6,3,7,7,1)
timerange(data=data, cluster=cluster, density=density)
```

Index