Package ‘provenance’

May 3, 2017

Title Statistical Toolbox for Sedimentary Provenance Analysis

Version 1.6

Description Bundles a number of established statistical methods to facilitate the visual interpretation of large datasets in sedimentary geology. Includes functionality for adaptive kernel density estimation, multidimensional scaling, generalised procrustes analysis and individual differences scaling using a variety of dissimilarity measures. Univariate provenance proxies, such as single-grain ages or (isotopic) compositions are compared with the Kolmogorov-Smirnov dissimilarity and Sircombe-Hazelton L2-norm. Categorical provenance proxies, such as mineralogical, petrographic or chemical compositions are compared with the Aitchison and Bray-Curtis distances. Also included are tools to plot compositional data on ternary diagrams, to calculate the sample size required for specified levels of statistical precision, and to assess the effects of hydraulic sorting on detrital compositions. Includes an intuitive query-based user interface for users who are not proficient in R.

Author Pieter Vermeesch [aut, cre]

Maintainer Pieter Vermeesch <p.vermeesch@ucl.ac.uk>

Depends R (>= 3.0.0)

Imports MASS, methods

License GPL-2

LazyData true

RoxygenNote 5.0.1

R topics documented:

`amalgamate` *Group components of a composition*

Description

Adds several components of a composition together into a single component

Usage

```
amalgamate(X, ...)

## Default S3 method:
amalgamate(X, ...)

## S3 method for class 'compositional'
amalgamate(X, ...)

## S3 method for class 'SRDcorrected'
amalgamate(X, ...)
```

Arguments

`X`	a compositional dataset
`...`	a series of new labels assigned to strings or vectors of strings denoting the components that need amalgamating

Value

an object of the same class as X with fewer components

Examples

```
data(Namib)
HMcomponents <- c("zr","tm","rt","TiOx","sph","ap","ep",
                  "gt","st","amp","cpx","opx")
am <- amalgamate(Namib$PTHM,feldspars=c("KF","P"),
                 lithics=c("Lm","Lv","Ls"),heavies=HMcomponents)
plot(ternary(am))
```

`as.acomp` *create an* `acomp` *object*

Description

Convert an object of class `compositional` to an object of class `acomp` for use in the `compositions` package

Usage

```
as.acomp(x)
```

Arguments

`x`	an object of class `compositional`

Value

a data.frame

Examples

```
data(Namib)
qfl <- ternary(Namib$PT,c('Q'),c('KF','P'),c('Lm','Lv','Ls'))
plot(qfl,type="QFL.dickinson")
qfl.acomp <- as.acomp(qfl)
## uncomment the next two lines to plot an error
## ellipse using the compositions package:
# library(compositions)
# ellipses(mean(qfl.acomp),var(qfl.acomp),r=2)
```

---

as.compositional   *create a* compositional *object*

Description

Convert an object of class matrix, data.fram or acomp to an object of class compositional

Usage

```
as.compositional(x, method = NULL, colmap = "rainbow")
```

Arguments

x	an object of class matrix, data.fram or acomp
method	dissimilarity measure, either 'aitchison' for Aitchison's CLR-distance or 'bray' for the Bray-Curtis distance.
colmap	the colour map to be used in pie charts.

Value

an object of class compositional

Examples

```
data(Namib)
PT.acomp <- as.acomp(Namib$PT)
PT.compositional <- as.compositional(PT.acomp)
print(Namib$PT$x - PT.compositional$x)
## uncomment the following lines for an illustration of using this
## function to integrate the \code{provenance} package with \code{compositions}
# library(compositions)
# data(Glacial)
# a.glac <- acomp(Glacial)
# c.glac <- as.compositional(a.glac)
# summaryplot(c.glac,ncol=8)
```

as.data.frame.compositional *create a* data.frame *object*

Description

Convert an object of class compositional to a data.frame for use in the robCompositions package

Usage

```
## S3 method for class 'compositional'
as.data.frame(x, ...)
```

Arguments

x	an object of class compositional
...	optional arguments to be passed on to the generic function

Value

a data.frame

Examples

```
data(Namib)
qfl <- ternary(Namib$PT,c('Q'),c('KF','P'),c('Lm','Lv','Ls'))
plot(qfl,type="QFL.dickinson")
qfl.frame <- as.data.frame(qfl)
## uncomment the next two lines to plot an error
## ellipse using the robCompositions package:
# library(robCompositions)
# pca <- pcaCoDa(qfl.frame)
# plot(pca,xlabs=rownames(qfl.frame))
```

botev *Compute the optimal kernel bandwidth*

Description

Uses the diffusion algorithm of Zdravko Botev (2011) to calculate the bandwidth for kernel density estimation

Usage

```
botev(x)
```

Arguments

x	a vector of ordinal data

Value

a scalar value with the optimal bandwidth

Author(s)

Dzdravko Botev

References

Botev, Z. I., J. F. Grotowski, and D. P. Kroese. "Kernel density estimation via diffusion." The Annals of Statistics 38.5 (2010): 2916-2957.

Examples

```
fname <- system.file("DZ.csv",package="provenance")
bw <- botev(read.distributional(fname)$x$N1)
print(bw)
```

---

bray.diss *Bray-Curtis dissimilarity*

Description

Calculates the Bray-Curtis dissimilarity between two samples

Usage

```
bray.diss(x, y)
```

Arguments

x	a vector containing the first compositional sample
y	a vector of length(x) containing the second compositional sample

Value

a scalar value

Examples

```
data(Namib)
print(bray.diss(Namib$HM$x["N1",],Namib$HM$x["N2",]))
```

CLR — *Centred logratio transformation*

Description

Calculates Aitchison's centered logratio transformation for a dataset of class `compositional`

Usage

```
CLR(x)
```

Arguments

`x`	an object of class `compositional`

Value

a matrix of CLR coordinates

Examples

```
# The following code shows that applying provenance's PCA function
# to compositional data is equivalent to applying R's built-in
# princomp function to the CLR transformed data.
data(Namib)
plot(PCA(Namib$Major))
dev.new()
clrdat <- CLR(Namib$Major)$x
biplot(princomp(clrdat))
```

combine — *Combine samples of distributional data*

Description

Lumps all single grain analyses of several samples together under a new name

Usage

```
combine(X, ...)
```

Arguments

`X`	a distributional dataset
`...`	a series of new labels assigned to strings or vectors of strings denoting the samples that need amalgamating

Value

a distributional data object with fewer samples than X

Examples

```
data(Namib)
combined <- combine(Namib$DZ,east=c('N3','N4','N5','N6','N7','N8','N9','N10'),
                    west=c('N1','N2','N11','N12','T8','T13'))
summaryplot(KDEs(combined))
```

---

densities *A list of rock and mineral densities*

Description

List of rock and mineral densities using the following abbreviations: Q (quartz), KF (K-feldspar), P (plagioclase), F (feldspar), Lvf (felsic/porfiritic volcanic rock fragments), Lvm (microlithic / porfiritic / trachitic volcanic rock fragments), Lcc (calcite), Lcd (dolomite), Lp (marl), Lch (chert), Lms (argillaceous / micaceous rock fragments), Lmv (metavolcanics), Lmf (metasediments), Lmb (metabasites), Lv (volcanic rock fragments), Lc (carbonates), Ls (sedimentary rock fragments), Lm (metamorphic rock fragments), Lu (serpentinite), mica, opaques, FeOx (Fe-oxides), turbids, zr (zircon), tm (tourmaline), rt (rutile), TiOx (Ti-oxides), sph (titanite), ap (apatite), mon (monazite), oth (other minerals), ep (epidote), othLgM (prehnite + pumpellyite + lawsonite + carpholite), gt (garnet), ctd (chloritoid), st (staurolite), and (andalusite), ky (kyanite), sil (sillimanite), amp (amphibole), px (pyroxene), cpx (clinopyroxene), opx (orthopyroxene), ol (olivine), spinel and othHM (other heavy minerals).

Author(s)

Alberto Resentini and Pieter Vermeesch

References

Resentini, A, Malusa M G and Garzanti, E. "MinSORTING: An Excel worksheet for modelling mineral grain-size distribution in sediments, with application to detrital geochronology and provenance studies." Computers & Geosciences 59 (2013): 90-97.

Garzanti, E, Ando, S and Vezzoli, G. "Settling equivalence of detrital minerals and grain-size dependence of sediment composition." Earth and Planetary Science Letters 273.1 (2008): 138-151.

See Also

restore, minsorting

Examples

```
data(Namib,densities)
N8 <- subset(Namib$HM,select="N8")
distribution <- minsorting(N8,densities,phi=2,sigmaphi=1,medium="air",by=0.05)
plot(distribution)
```

---

diss — *Calculate the dissimilarity matrix between two* distributional *or* compositional *datasets*

Description

Calculate the dissimilarity matrix between two datasets of class distributional or compositional using the Kolmogorov-Smirnov, Sircombe-Hazelton, Aitchison or Bray Curtis distance

Usage

```
diss(x, method)

## S3 method for class 'distributional'
diss(x, method = NULL)

## S3 method for class 'compositional'
diss(x, method = NULL)
```

Arguments

x	an object of class distributional or compositional
method	(optional) either "KS", "SH", "aitchison" or "bray"

Value

an object of class diss

Examples

```
data(Namib)
print(round(100*diss(Namib$DZ)))
```

---

endmembers — *Petrographic end-member compositions*

Description

A compositional dataset comprising the mineralogical compositions of the following end-members: undissected_magmatic_arc, dissected_magmatic_arc, ophiolite, recycled_clastic, undissected_continental_b transitional_continental_block, dissected_continental_block, subcreted_axial_belt and subducted_axial_belt

Author(s)

Alberto Resentini and Pieter Vermeesch

References

Resentini, A, Malusa M G and Garzanti, E. "MinSORTING: An Excel worksheet for modelling mineral grain-size distribution in sediments, with application to detrital geochronology and provenance studies." Computers & Geosciences 59 (2013): 90-97.

Garzanti, E, Ando, S and Vezzoli, G. "Settling equivalence of detrital minerals and grain-size dependence of sediment composition." Earth and Planetary Science Letters 273.1 (2008): 138-151.

See Also

minsorting

Examples

```
data(endmembers,densities)
ophiolite <- subset(endmembers,select="ophiolite")
plot(minsorting(ophiolite,densities,by=0.05))
```

`get.f` *Calculate the largest fraction that is likely to be missed*

Description

For a given sample size, returns the largest fraction which has been sampled with p x 100

Usage

```
get.f(n, p = 0.05)
```

Arguments

n	the number of grains in the detrital sample
p	the required level of confidence

Value

the largest fraction that is sampled with at least 100 x p certainty

References

Vermeesch, Pieter. "How many grains are needed for a provenance study?." Earth and Planetary Science Letters 224.3 (2004): 441-451.

Examples

```
print(get.f(60))
print(get.f(117))
```

get.n *Calculate the number of grains required to achieve a desired level of sampling resolution*

Description

Returns the number of grains that need to be analysed to decrease the likelihood of missing any fraction greater than a given size below a given level.

Usage

```
get.n(p = 0.05, f = 0.05)
```

Arguments

p	the probability that all n grains in the sample have missed at least one fraction of size f
f	the size of the smallest resolvable fraction (0<f<1)
n,	the number of grains in the sample

Value

the number of grains needed to reduce the chance of missing at least one fraction f of the total population to less than p

References

Vermeesch, Pieter. "How many grains are needed for a provenance study?." Earth and Planetary Science Letters 224.3 (2004): 441-451.

Examples

```
# number of grains required to be 99% that no fraction greater than 5% was missed:
print(get.n(0.01))
# number of grains required to be 90% that no fraction greater than 10% was missed:
print(get.n(p=0.1,f=0.1))
```

get.p — *Calculate the probability of missing a given population fraction*

Description

For a given sample size, returns the likelihood of missing any fraction greater than a given size

Usage

```
get.p(n, f = 0.05)
```

Arguments

n	the number of grains in the detrital sample
f	the size of the smallest resolvable fraction (0<f<1)

Value

the probability that all n grains in the sample have missed at least one fraction of size f

References

Vermeesch, Pieter. "How many grains are needed for a provenance study?." Earth and Planetary Science Letters 224.3 (2004): 441-451.

Examples

```
print(get.p(60))
print(get.p(117))
```

GPA — *Generalised Procrustes Analysis of configurations*

Description

Given a number of (2D) configurations, this function uses a combination of transformations (reflections, rotations, translations and scaling) to find a 'consensus' configuration which best matches all the component configurations in a least-squares sense.

Usage

```
GPA(X, scale = TRUE)
```

Arguments

X	a list of dissimilarity matrices
scale	boolean flag indicating if the transformation should include the scaling operation

Value

a two column vector with the coordinates of the group configuration

See Also

procrustes

`indscal` *Individual Differences Scaling of provenance data*

Description

Performs 3-way Multidimensional Scaling analysis using Carroll and Chang (1970)'s INdividual Differences SCALing method as implemented using De Leeuw and Mair (2011)'s stress majorization algorithm.

Usage

```
indscal(..., type = "ordinal")
```

Arguments

`...`	a sequence of datasets of class `distributional` or `compositional`
`type`	is either "ratio" or "ordinal"

Value

an object of class `INDSCAL`, i.e. a list containing the following items:

delta: Observed dissimilarities

obsdiss: List of observed dissimilarities, normalized

confdiss: List of configuration dissimilarities

conf: List of matrices of final configurations

gspace: Joint configurations aka group stimulus space

cweights: Configuration weights

stress: Stress-1 value

spp: Stress per point

sps: Stress per subject (matrix)

ndim: Number of dimensions

model: Type of smacof model

niter: Number of iterations

nobj: Number of objects

Author(s)

Jan de Leeuw and Patrick Mair

References

de Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: The R package smacof. Journal of Statistical Software, 31(3), 1-30, < http://www.jstatsoft.org/v31/i03/>

Examples

```
data(Namib)
plot(indscal(Namib$DZ,Namib$HM))
```

KDE *Create a kernel density estimate*

Description

Turns a vector of numbers into an object of class KDE using a combination of the Botev (2010) bandwidth selector and the Abramson (1982) adaptive kernel bandwidth modifier.

Usage

```
KDE(x, from = NA, to = NA, bw = NA, adaptive = TRUE, log = FALSE,
  n = 512, ...)
```

Arguments

x	a vector of numbers
from	minimum age of the time axis. If NULL, this is set automatically
to	maximum age of the time axis. If NULL, this is set automatically
bw	the bandwidth of the KDE. If NULL, bw will be calculated automatically using botev()
adaptive	boolean flag controlling if the adaptive KDE modifier of Abramson (1982) is used
log	transform the ages to a log scale if TRUE
n	horizontal resolution of the density estimate
...	optional arguments to be passed on to density

Value

an object of class KDE, i.e. a list containing the following items:

x: horizontal plot coordinates

y: vertical plot coordinates

bw: the base bandwidth of the density estimate

ages: the data values from the input to the KDE function

See Also

KDEs

Examples

```
data(Namib)
samp <- Namib$DZ$x[['N1']]
dens <- KDE(samp,0,3000,kernel="epanechnikov")
plot(dens)
```

KDEs — *Generate an object of class* `KDEs`

Description

Convert a dataset of class `distributional` into an object of class `KDEs` for further processing by the `summaryplot` function.

Usage

```
KDEs(x, from = NA, to = NA, bw = NA, samebandwidth = TRUE,
  adaptive = TRUE, pch = NA, normalise = FALSE, log = FALSE, n = 512,
  ...)
```

Arguments

`x`	an object of class `distributional`
`from`	minimum limit of the x-axis.
`to`	maximum limit of the x-axis.
`bw`	the bandwidth of the kernel density estimates. If bw = NA, the bandwidth will be set automatically using `botev()`
`samebandwidth`	boolean flag indicating whether the same bandwidth should be used for all samples. If samebandwidth = TRUE and bw = NULL, then the function will use the median bandwidth of all the samples.
`adaptive`	boolean flag switching on the adaptive bandwidth modifier of Abramson (1982)
`pch`	(optional) symbol to be used to mark the sample points along the x-axis
`normalise`	boolean flag indicating whether or not the KDEs should all integrate to the same value.
`log`	boolean flag indicating whether the data should by plotted on a logarithmic scale.
`n`	horizontal resolution of the density estimates
`...`	optional parameters to be passed on to `density`

Value

an object of class KDEs, i.e. a list containing the following items:

kdes: a named list with objects of class KDE

from: the beginning of the common time scale

to: the end of the common time scale

themax: the maximum probability density of all the KDEs

pch: the plot symbol to be used by plot.KDEs

xlabel: the x-axis label to be used by plot.KDEs

See Also

KDE

Examples

```
data(Namib)
KDEs <- KDEs(Namib$DZ,0,3000,pch=NA)
summaryplot(KDEs,ncol=3)
```

KS.diss *Kolmogorov-Smirnov dissimilarity*

Description

Returns the Kolmogorov-Smirnov dissimilarity between two samples

Usage

```
KS.diss(x, y)
```

Arguments

x	the first sample as a vector
y	the second sample as a vector

Value

a scalar value representing the maximum vertical distance between the two cumulative distributions

Examples

```
data(Namib)
print(KS.diss(Namib$DZ$x[['N1']],Namib$DZ$x[['T8']]))
```

MDS *Multidimensional Scaling*

Description

Performs classical or nonmetric Multidimensional Scaling analysis

Multidimensional Scaling of compositional data

Multidimensional Scaling of distributional data

Multidimensional Scaling of a dissimilarity matrix

Usage

```
MDS(x, ...)

## S3 method for class 'compositional'
MDS(x, classical = FALSE, k = 2, ...)

## S3 method for class 'distributional'
MDS(x, classical = FALSE, k = 2, ...)

## S3 method for class 'diss'
MDS(x, classical = FALSE, k = 2, ...)
```

Arguments

`x`	an object of class `distributional`, `compositional` or `diss`
`...`	optional arguments to be passed onto `diss` (if `x` is of class `compositional` or `distributional`) or onto `cmdscale` or `isoMDS` (if `x` is of class `dist`).
`classical`	boolean flag indicating whether classical (TRUE) or nonmetric (FALSE) MDS should be used
`k`	the desired dimensionality of the solution

Value

an object of class `MDS`, i.e. a list containing the following items:

`points`: a two column vector of the fitted configuration

`classical`: a boolean flag indicating whether the MDS configuration was obtained by classical (`TRUE`) or nonmetric (`FALSE`) MDS.

`diss`: the dissimilarity matrix used for the MDS analysis

`stress`: (only if `classical=TRUE`) the final stress achieved (in percent)

Examples

```
data(Namib)
plot(MDS(Namib$Major,classical=TRUE))
```

`minsorting` *Assess settling equivalence of detrital components*

Description

Models grain size distribution of minerals and rock fragments of different densities

Usage

```
minsorting(X, dens, sname = NULL, phi = 2, sigmaphi = 1,
  medium = "freshwater", from = -2.25, to = 5.5, by = 0.25)
```

Arguments

X	an object of class compositional
dens	a vector of mineral and rock densities
sname	sample name if unspecified, the first sample of the dataset will be used
phi	the mean grain size of the sample in Krumbein's phi units
sigmaphi	the standard deviation of the grain size distirbution, in phi units
medium	the transport medium, one of either "air", "freshwater" or "seawater"
from	the minimum grain size to be evaluated, in phi units
to	the maximum grain size to be evaluated, in phi units
by	the grain size interval of the output table, in phi units

Value

an object of class `minsorting`, i.e. a list with two tables:

mfract: the grain size distribution of each mineral (sum of the columns = 1)

mcomp: the composition of each mineral (sum of the rows = 1)

Author(s)

Alberto Resentini and Pieter Vermeesch

References

Resentini, A, Malusa, M G and Garzanti, E. "MinSORTING: An Excel worksheet for modelling mineral grain-size distribution in sediments, with application to detrital geochronology and provenance studies." Computers & Geosciences 59 (2013): 90-97.

Garzanti, E, Ando, S and Vezzoli, G. "Settling equivalence of detrital minerals and grain-size dependence of sediment composition." Earth and Planetary Science Letters 273.1 (2008): 138-151.

See Also

restore

Examples

```
data(endmembers,densities)
distribution <- minsorting(endmembers,densities,sname='ophiolite',phi=2,
                           sigmaphi=1,medium="seawater",by=0.05)
plot(distribution,cumulative=FALSE)
```

`Namib` *An example dataset*

Description

A large dataset of provenance data from Namibia comprised of 14 sand samples from the Namib Sand Sea and 2 samples from the Orange River.

Details

`Namib` is a list containing the following 6 items:

`DZ`: a `distributional` dataset containing the zircon U-Pb ages for ca. 100 grains from each sample, as well as their (1-sigma) analytical uncertainties.

`PT`: a `compositional` dataset with the bulk petrography of the samples, i.e. the quartz ('Q'), K-feldspar ('KF'), plagioclase ('P'), and lithic fragments of metamorphic ('Lm'), volcanic ('Lv') and sedimentary ('Ls') origin.

`HM`: a `compositional` dataset containing the heavy mineral composition of the samples, comprised of zircon ('zr'), tourmaline ('tm'), rutile ('rt'), Ti-oxides ('TiOx'), titanite ('sph'), apatite ('ap'), epidote ('ep'), garnet ('gt'), staurolite ('st'), andalusite ('and'), kyanite ('ky'), sillimanite ('sil'), amphibole ('amp'), clinopyroxene ('cpx') and orthopyroxene ('opx').

`PTHM`: a `compositional` dataset combining the variables contained in `PT` and `HM` plus 'mica', 'opaques', 'turbids' and 'other' transparent heavy minerals ('LgM'), normalised to 100.

`Major`: a `compositional` dataset listing the concentrations (in wt TiO_2, P_2O_5 and MnO.

`Trace`: a `compositional` data listing the concentrations (in ppm) of Rb, Sr, Ba, Sc, Y, La, Ce, Pr, Nd, Sm, Gd, Dy, Er, Yb, Th, U, Zr, Hf, V, Nb, Cr, Co, Ni, Cu, Zn, Ga and Pb.

Author(s)

Pieter Vermeesch and Eduardo Garzanti

References

Vermeesch, P. and Garzanti, E., Making geological sense of 'Big Data' in sedimentary provenance analysis, Chemical Geology 409 (2015) 20-27

Examples

```
data(Namib)
samp <- Namib$DZ$x[['N1']]
dens <- KDE(samp,0,3000)
plot(dens)
```

PCA *Principal Component Analysis*

Description

Performs PCA of compositional data using a centred logratio distance

Usage

```
PCA(x, ...)
```

Arguments

`x`	an object of class `compositional`
`...`	optional arguments to R's `princomp function`

Value

an object of classes `PCA`, which is synonymous to the stats packages' `princomp` class.

Examples

```
data(Namib)
plot(MDS(Namib$Major,classical=TRUE))
dev.new()
plot(PCA(Namib$Major),asp=1)
print("This example demonstrates the equivalence of classical MDS and PCA")
```

plot.compositional *Plot a pie chart*

Description

Plots an object of class `compositional` as a pie chart

Usage

```
## S3 method for class 'compositional'
plot(x, sname, annotate = TRUE, colmap = NULL, ...)
```

Arguments

`x`	an object of class `compositional`
`sname`	the sample name
`annotate`	a boolean flag controlling if the pies of the pie-chart should be labeled
`colmap`	an optional string with the name of one of R's built-in colour palettes (e.g., heat.colors, terrain.colors, topo.colors, cm.colors), which are to be used for plotting the data.
`...`	optional parameters to be passed on to the graphics object

Examples

```
data(Namib)
plot(Namib$HM,'N1',colmap='heat.colors')
```

---

plot.distributional *Plot continuous data as histograms or cumulative age distributions*

Description

Plot one or several samples from a `distributional` dataset as a histogram or Cumulative Age Distributions (CAD).

Usage

```
## S3 method for class 'distributional'
plot(x, snames = NULL, annotate = TRUE,
  CAD = FALSE, pch = NA, verticals = TRUE, colmap = NULL, ...)
```

Arguments

`x`	an object of class `distributional`
`snames`	a string or a vector of string with the names of the samples that need plotting if `snames` is a vector, then the function will default to a CAD.
`annotate`	boolean flag indicating whether the x- and y-axis should be labeled
`CAD`	boolean flag indicating whether the data should be plotted as a cumulative age distribution or a histogram. For multi-sample plots, the function will override this value with `TRUE`.
`pch`	an optional symbol to mark the sample points along the CAD
`verticals`	boolean flag indicating if the horizontal lines of the CAD should be connected by vertical lines
`colmap`	an optional string with the name of one of R's built-in colour palettes (e.g., heat.colors, terrain.colors, topo.colors, cm.colors), which are to be used for plotting the data.
`...`	optional arguments to the generic `plot` function

Examples

```
data(Namib)
plot(Namib$DZ,c('N1','N2'))
```

---

plot.GPA — *Plot a Procrustes configuration*

Description

Plots the group configuration of a Generalised Procrustes Analysis

Usage

```
## S3 method for class 'GPA'
plot(x, pch = NA, pos = NULL, col = "black", bg = "white",
  cex = 1, ...)
```

Arguments

`x`	an object of class `GPA`
`pch`	plot symbol
`pos`	position of the sample labels relative to the plot symbols if pch != NA
`col`	plot colour (may be a vector)
`bg`	background colour (may be a vector)
`cex`	relative size of plot symbols
`...`	optional arguments to the generic `plot` function

See Also

procrustes

Examples

```
data(Namib)
GPA <- procrustes(Namib$DZ,Namib$HM)
coast <- c('N1','N2','N3','N10','N11','N12','T8','T13')
snames <- names(Namib$DZ)
bgcol <- rep('yellow',length(snames))
bgcol[which(snames %in% coast)] <- 'red'
plot(GPA,pch=21,bg=bgcol)
```

plot.INDSCAL *Plot an INDSCAL group configuration and source weights*

Description

Given an object of class INDSCAL, generates two plots: the group configuration and the subject weights. Together, these describe a 3-way MDS model.

Usage

```
## S3 method for class 'INDSCAL'
plot(x, asp = 1, pch = NA, pos = NULL, col = "black",
  bg = "white", cex = 1, xlab = "X", ylab = "Y", xaxt = "n",
  yaxt = "n", ...)
```

Arguments

x	an object of class INDSCAL
asp	the aspect ratio of the plot
pch	plot symbol (may be a vector)
pos	position of the sample labels relative to the plot symbols if pch != NA
col	plot colour (may be a vector)
bg	background colour (may be a vector)
cex	relative size of plot symbols
xlab	a string with the label of the x axis
ylab	a string with the label of the y axis
xaxt	if = 'y', adds ticks to the x axis
yaxt	if = 'y', adds ticks to the y axis
...	optional arguments to the generic plot function

See Also

indscal

Examples

```
data(Namib)
coast <- c('N1','N2','N3','N10','N11','N12','T8','T13')
snames <- names(Namib$DZ)
pch <- rep(21,length(snames))
pch[which(snames %in% coast)] <- 22
plot(indscal(Namib$DZ,Namib$HM),pch=pch)
```

plot.KDE *Plot a kernel density estimate*

Description

Plots an object of class KDE

Usage

```
## S3 method for class 'KDE'
plot(x, pch = "|", xlab = "age [Ma]", ylab = "", ...)
```

Arguments

x	an object of class KDE
pch	the symbol used to show the samples. May be a vector. Set pch = NA to turn them off.
xlab	the label of the x-axis
ylab	the label of the y-axis
...	optional parameters to be passed on to the graphics object

See Also

KDE

Examples

```
data(Namib)
samp <- Namib$DZ$x[['N1']]
dens <- KDE(samp,from=0,to=3000)
plot(dens)
```

plot.MDS *Plot an MDS configuration*

Description

Plots the coordinates of a multidimensional scaling analysis as an X-Y scatter plot or 'map' and, if x$classical = FALSE, a Shepard plot.

Usage

```
## S3 method for class 'MDS'
plot(x, nnlines = FALSE, pch = NA, pos = NULL, cex = 1,
  col = "black", bg = "white", ...)
```

Arguments

`x`	an object of class `MDS`
`nnlines`	if TRUE, draws nearest neighbour lines
`pch`	plot character (see ?plot for details). May be a vector.
`pos`	position of the sample labels relative to the plot symbols if pch != NA
`cex`	relative size of plot symbols (see ?par for details)
`col`	plot colour (may be a vector)
`bg`	background colour (may be a vector)
`...`	optional arguments to the generic `plot` function

See Also

MDS

Examples

```
data(Namib)
mds <- MDS(Namib$DZ)
coast <- c('N1','N2','N3','N10','N11','N12','T8','T13')
snames <- names(Namib$DZ)
bgcol <- rep('yellow',length(snames))
bgcol[which(snames %in% coast)] <- 'red'
plot(mds,pch=21,bg=bgcol)
```

`plot.minsorting` *Plot inferred grain size distributions*

Description

Plot the grain size distributions of the different minerals under consideration

Usage

```
## S3 method for class 'minsorting'
plot(x, cumulative = FALSE, components = NULL, ...)
```

Arguments

`x`	an object of class `minsorting`
`cumulative`	boolean flag indicating whether the grain size distribution should be plotted as a density or cumulative probability curve.
`components`	string or list of strings with the names of a subcomposition that needs plotting
`...`	optional parameters to be passed on to graphics::matplot (see ?par for details)

See Also

minsorting

Examples

```
data(endmembers,densities)
OPH <- subset(endmembers,select="ophiolite")
distribution <- minsorting(OPH,densities,phi=2,sigmaphi=1,medium="air",by=0.05)
plot(distribution,components=c('F','px','opaques'))
```

---

plot.PCA *Compositional biplot*

Description

Plot the results of a principal components analysis as a biplot

Usage

```
## S3 method for class 'PCA'
plot(x, ...)
```

Arguments

x	an object of class PCA
...	optional arguments of the biplot function

See Also

PCA

Examples

```
data(Namib)
plot(PCA(Namib$Major))
```

plot.ternary *Plot a ternary diagram*

Description

Plots triplets of compositional data on a ternary diagram

Usage

```
## S3 method for class 'ternary'
plot(x, type = "empty", pch = NA, pos = NULL,
  labels = names(x), showpath = FALSE, bg = NA, col = "cornflowerblue",
  ...)
```

Arguments

`x`	an object of class `ternary`
`type`	adds annotations to the ternary diagram, one of either `empty`, `QFL.descriptive`, `QFL.folk` or `QFL.dickinson`
`pch`	plot character, see `?par` for details (may be a vector)
`pos`	position of the sample labels relative to the plot symbols if pch != NA
`labels`	vector of strings to be added to the plot symbols
`showpath`	if `x` has class `SRDcorrected`, and `showpath`==TRUE, the intermediate values of the SRD correction will be plotted on the ternary diagram as well as the final composition
`bg`	background colour for the plot symbols (may be a vector)
`col`	colour to be used for the background lines (if applicable)
`...`	optional arguments to the generic `points` function

See Also

ternary

Examples

```
data(Namib)
tern <- ternary(Namib$PT,'Q',c('KF','P'),c('Lm','Lv','Ls'))
plot(tern,type='QFL.descriptive',pch=21,bg='red',labels=NULL)
```

`procrustes` *Generalised Procrustes Analysis of provenance data*

Description

Given a number of input datasets, this function performs an MDS analysis on each of these and the feeds the resulting configurations into the `GPA()` function.

Usage

```
procrustes(...)
```

Arguments

`...`	a sequence of datasets of classes `distributional` and `compositional`

Value

an object of class `GPA`, i.e. a list containing the following items:

`points`: a two column vector with the coordinates of the group configuration

`labels`: a list with the sample names

Author(s)

Pieter Vermeesch

References

Gower, J.C. (1975). Generalized Procrustes analysis, Psychometrika, 40, 33-50.

See Also

GPA

Examples

```
data(Namib)
gpa <- procrustes(Namib$DZ,Namib$HM)
plot(gpa)
```

provenance — *Menu-based interface for* provenance

Description

For those less familiar with the syntax of the R programming language, the provenance() function provides a user-friendly way to access the most important functionality in the form of a menu-based query interface. Further details and examples are provided on http://provenance.london-geochron.com

Usage

```
provenance()
```

Author(s)

Pieter Vermeesch

References

Vermeesch, P., Resentini, A. and Garzanti, E., an R package for statistical provenance analysis, Sedimentary Geology, doi:10.1016/j.sedgeo.2016.01.009.

See Also

http://provenance.london-geochron.com

read.compositional — *Read a .csv file with categorical data*

Description

Reads a data table containing categorical data (e.g. petrographic, heavy mineral or geochemical data)

Usage

```
read.compositional(fname, method = NULL, colmap = "rainbow")
```

Arguments

fname	a string with the path to the .csv file
method	either "bray" (for the Bray-Curtis distance) or "aitchison" (for Aitchison's central logratio distance). If omitted, the function defaults to 'aitchison', unless there are zeros present in the data.
colmap	an optional string with the name of one of R's built-in colour palettes (e.g., heat.colors, terrain.colors, topo.colors, cm.colors), which are to be used for plotting the data.

Value

an object of class `compositional`, i.e. a list with the following items:

`x`: a data frame with the samples as rows and the categories as columns

`method`: either "aitchison" (for Aitchison's centred logratio distance) or "bray" (for the Bray-Curtis distance)

Examples

```
fname <- system.file("Major.csv",package="provenance")
Major <- read.compositional(fname)
plot(PCA(Major))
```

`read.densities` *Read a .csv file with mineral and rock densities*

Description

Reads a data table containing densities to be used for hydraulic sorting corrections (minsorting and srd functions)

Usage

```
read.densities(fname)
```

Arguments

`fname`	a string with the path to the .csv file

Value

a vector with mineral and rock densities

Examples

```
data(Namib,densities)
N8 <- subset(Namib$HM,select="N8")
distribution <- minsorting(N8,densities,phi=2,sigmaphi=1,medium="air",by=0.05)
plot(distribution)
```

read.distributional *Read a .csv file with continuous (detrital zircon) data*

Description

Reads a data table containing continuous data (e.g. detrital zircon ages)

Usage

```
read.distributional(fname, errorfile = NA, method = "KS",
  xlab = "age [Ma]", colmap = "rainbow")
```

Arguments

fname	the path of a .csv file with the input data, arranged in columns.
errorfile	the (optional) path of a .csv file with the standard errors of the input data, arranged by column in the same order as fname. Must be specified if the data are to be compared with the Sircombe-Hazelton dissimilarity.
method	an optional string specifying the dissimilarity measure which should be used for comparing this with other datasets. Should be one of either "KS" (for Kolmogorov-Smirnov) or "SH" (for Sircombe and Hazelton). If method = "SH", then errorfile should be specified. If method = "SH" and errorfile is unspecified, then the program will default back to the Kolmogorov-Smirnov dissimilarity.
xlab	an optional string specifying the nature and units of the data. This string is used to label kernel density estimates.
colmap	an optional string with the name of one of R's built-in colour palettes (e.g., heat.colors, terrain.colors, topo.colors, cm.colors), which are to be used for plotting the data.

Value

an object of class distributional, i.e. a list with the following items:

x: a named list of vectors containing the numerical data for each sample

err: an (optional) named list of vectors containing the standard errors of x

method: either "KS" (for Kolmogorov-Smirnov) or "SH" (for Sircombe Hazelton)

breaks: a vector with the locations of the histogram bin edges

xlab: a string containing the label to be given to the x-axis on all plots

Examples

```
agefile <- system.file("DZ.csv",package="provenance")
errfile <- system.file("DZerr.csv",package="provenance")
DZ <- read.distributional(agefile,errfile)
plot(KDE(DZ$x$N1))
```

`restore` *Undo the effect of hydraulic sorting*

Description

Restore the detrital composition back to a specified source rock density (SRD)

Usage

```
restore(X, dens, target = 2.71)
```

Arguments

`X`	an object of class `compositional`
`dens`	a vector of rock and mineral densities
`target`	the target density (in g/cm3)

Value

an object of class `SRDcorrected`, i.e. an object of class `compositional` which is a daughter of class `compositional` containing the restored composition, plus one additional member called `restoration`, containing the intermediate steps of the SRD correction algorithm.

Author(s)

Alberto Resentini and Pieter Vermeesch

References

Garzanti E, Ando, S and Vezzoli, G. "Settling equivalence of detrital minerals and grain-size dependence of sediment composition." Earth and Planetary Science Letters 273.1 (2008): 138-151.

See Also

minsorting

Examples

```
data(Namib,densities)
rescomp <- restore(Namib$PTHM,densities,2.71)
HMcomp <- c("zr","tm","rt","sph","ap","ep","gt",
            "st","amp","cpx","opx")
amcomp <- amalgamate(rescomp,Plag="P",HM=HMcomp,Opq="opaques")
plot(ternary(amcomp),showpath=TRUE)
```

`SH.diss` *Sircombe and Hazelton distance*

Description

Calculates Sircombe and Hazelton's L2 distance between the Kernel Functional Estimates (KFEs, not to be confused with Kernel Density Estimates!) of two samples with specified analytical uncertainties

Usage

```
SH.diss(x, i, j, c.con = 0)
```

Arguments

`x`	an object of class `distributional`
`i`	index of the first sample
`j`	index of the second sample
`c.con`	smoothing bandwidth of the kernel functional estimate

Value

a scalar value expressing the L2 distance between the KFEs of samples i and j

Author(s)

Keith Sircombe and Martin Hazelton

References

Sircombe, K. N., and M. L. Hazelton. "Comparison of detrital zircon age distributions by kernel functional estimation." Sedimentary Geology 171.1 (2004): 91-111.

See Also

KS.diss

Examples

```
datfile <- system.file("DZ.csv",package="provenance")
errfile <- system.file("DZerr.csv",package="provenance")
DZ <- read.distributional(datfile,errfile)
d <- SH.diss(DZ,1,2)
print(d)
```

subset.compositional *Get a subset of compositional data*

Description

Return a subset of provenance data according to some specified indices

Usage

```
## S3 method for class 'compositional'
subset(x, subset = NULL, select = NULL,
  components = NULL, ...)
```

Arguments

x	an object of class compositional
subset	logical expression indicating elements or rows to keep: missing values are taken as false.
select	a vector of sample names.
components	a vector specifying a subcomposition
...	optional arguments for the generic subset function

Value

an object of class compositional

See Also

read.compositional

subset.distributional *Get a subset of distributional data*

Description

Return a subset of provenance data according to some specified indices

Usage

```
## S3 method for class 'distributional'
subset(x, subset = NULL, select = NULL, ...)
```

Arguments

`x`	an object of class `distributional`
`subset`	logical expression indicating elements or rows to keep: missing values are taken as false.
`select`	a vector of sample names
`...`	optional arguments for the generic subset function

Value

an object of class `distributional`

See Also

read.distributional

Examples

```
data(Namib)
coast <- subset(Namib$HM,select=c("N1","N2","T8","T13","N12","N13"))
summaryplot(coast,ncol=2)
```

`summaryplot` *Joint plot of several provenance datasets*

Description

Arranges kernel density estimates and pie charts in a grid format

Usage

```
summaryplot(..., ncol = 1)
```

Arguments

`...`	a sequence of datasets of class `compositional`, `KDEs`, or `distributional`
`ncol`	the number of columns

Value

a summary plot of all the data comprised of KDEs for the datasets of class `KDEs`, pie charts for those of class `compositional` and histograms for those of class `distributional`.

See Also

KDEs

Examples

```
data(Namib)
KDEs <- KDEs(Namib$DZ,0,3000)
summaryplot(KDEs,Namib$HM,Namib$PT,ncol=2)
```

`ternary` *Define a ternary composition*

Description

Create an object of class `ternary`

Usage

```
ternary(X, x = NULL, y = NULL, z = NULL)
```

Arguments

`X`	an object of class `compositional`
`x`	string or a vector of strings indicating the variables making up the first subcomposition of the ternary system. If omitted, the first component of X is used instead.
`y`	second (set of) variables
`z`	third (set of) variables

Value

an object of class `ternary`, i.e. a list containing:

x: a three column matrix (or vector) of ternary compositions.

and (if X is of class `SRDcorrected`)

restoration: a list of intermediate ternary compositions inherited from the SRD correction

See Also

restore

Examples

```
data(Namib)
tern <- ternary(Namib$PT,c('Q'),c('KF','P'),c('Lm','Lv','Ls'))
plot(tern,type="QFL")
```

Index