Package: geostats (via r-universe)

October 18, 2024

Title An Introduction to Statistics for Geoscientists

Version 1.6

Date 2023-01-02

Description A collection of datasets and simplified functions for an introductory (geo)statistics module at University College London. Provides functionality for compositional, directional and spatial data, including ternary diagrams, Wulff and Schmidt stereonets, and ordinary kriging interpolation. Implements logistic and (additive and centred) logratio transformations. Computes vector averages and concentration parameters for the von-Mises distribution. Includes a collection of natural and synthetic fractals, and a simulator for deterministic chaos using a magnetic pendulum example. The main purpose of these functions is pedagogical. Researchers can find more complete alternatives for these tools in other packages such as 'compositions', 'robCompositions', 'sp', 'gstat' and 'RFOC'. All the functions are written in plain R, with no compiled code and a minimal number of dependencies. Theoretical background and worked examples are available at <`https://tinyurl.com/UCLgeostats/`>.

Author Pieter Vermeesch [aut, cre]

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

Depends R (>= 3.5.0)

Suggests MASS

License GPL-3

URL `https://github.com/pvermees/geostats/`

LazyData true

RoxygenNote 7.2.1

Encoding UTF-8

NeedsCompilation no

Date/Publication 2023-01-07 09:20:02 UTC

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

RemoteUrl https://github.com/cran/geostats

RemoteRef HEAD

RemoteSha 18f66dc9e25aff0c943e7a33d0281bef7b4ce7bb

Contents

`ACNK` *A-CN-K compositions*

Description

Synthetic A (Al_2O_3) – CN ($CaO+Na_2O$) – K (K_2O) data table.

Examples

```
data(ACNK,package='geostats')
ternary(ACNK,type='p',labels=c(expression('Al'[2]*'O'[3]),
                               expression('CaO+Na'[2]*'O'),
                               expression('K'[2]*'O')))
```

`alr` *additive logratio transformation*

Description

Maps compositional data from an n-dimensional simplex to an $(n-1)$-dimensional Euclidean space with Aitchison's additive logratio transformation.

Usage

```
alr(dat, inverse = FALSE)
```

Arguments

`dat`	an n column data frame or matrix
`inverse`	if `TRUE`, applies the inverse alr tranformation

Value

If `inverse=FALSE`, returns an $(n-1) \times m$ matrix of logratios; otherwise returns an $(n+1) \times m$ matrix of compositional data whose columns add up to 1.

Examples

```
xyz <- rbind(c(0.03,99.88,0.09),
             c(70.54,25.95,3.51),
             c(72.14,26.54,1.32))
colnames(xyz) <- c('a','b','c')
rownames(xyz) <- 1:3
uv <- alr(xyz)
XYZ <- alr(uv,inverse=TRUE)
xyz/XYZ
```

`boxcount` *box counting*

Description

Count the number of boxes needed to cover all the 1s in a matrix of 0s and 1s.

Count the number of boxes needed to cover all the 1s in a matrix of 0s and 1s.

Usage

```
boxcount(mat, size)
```


```
boxcount(mat, size)
```


Arguments

`mat`	a square square matrix of 0s and 1s, whose size should be a power of 2.
`size`	the size (pixels per side) of the boxes, whose size should be a power of 2.

Value

an integer

an integer

Examples

```
g <- sierpinski(n=5)
boxcount(mat=g,size=16)
g <- sierpinski(n=5)
boxcount(mat=g,size=16)
```

`Britain` *British coast*

Description

A 512×512 pixel image of the British coastline.

Examples

```
data(Britain,package='geostats')
p <- par(mfrow=c(1,2))
image(Britain)
fractaldim(Britain)
par(p)
```

`cantor` *Cantor set*

Description

Calculates or plots a Cantor set of fractal lines, which is generated using a recursive algorithm that is built on a line segment whose middle third is removed. Each level of recursion replaces each black line by the same pattern.

Usage

```
cantor(n = 5, plot = FALSE, add = FALSE, Y = 0, lty = 1, col = "black", ...)
```

Arguments

`n`	an integer value controling the number of recursive levels.
`plot`	logical. If `TRUE`, the Cantor set is plotted, otherwise a list of breaks and counts is returned.
`add`	logical (only used if `plot=TRUE`). If `add=FALSE`, then a brand new figure is created; otherwise the Cantor set is added to an existing plot.
`Y`	y-value for the plot (only used if `plot=TRUE`).
`lty`	line type (see `pars()` for details)
`col`	colour of the Cantor lines.
`...`	optional arguments to be passed on to `matplot` or `matlines`.

Value

a square matrix with 0s and 1s.

Examples

```
plot(c(0,1),y=c(0,1),type='n',bty='n',ann=FALSE,xaxt='n',yaxt='n',xpd=NA)
cantor(n=0,Y=1.00,plot=TRUE,add=TRUE)
cantor(n=1,Y=0.75,plot=TRUE,add=TRUE)
cantor(n=2,Y=0.50,plot=TRUE,add=TRUE)
cantor(n=3,Y=0.25,plot=TRUE,add=TRUE)
cantor(n=4,Y=0.00,plot=TRUE,add=TRUE)
```

---

catchments *properties of 20 river catchments*

Description

six different (three discrete, three continuous) measurements for twenty fictitious river catchments, containing their dominant lithology (categorical data), stratigraphic age (ordinal data), number of springs (count data), the pH of the river water (Cartesian quantity), its Ca/Mg ratio (Jeffreys quantity) and the percentage covered by vegetation (proportion).

Examples

```
data(catchments,package='geostats')
hist(catchments$pH)
```

---

circle.plot *plot circular data*

Description

Plots directional data as ticks on a circle, with angles plotting in a clockwise direction from the top.

Usage

```
circle.plot(a, degrees = FALSE, tl = 0.1, ...)
```

Arguments

`a`	angle(s), scalar or vector
`degrees`	logical. `TRUE` for degrees, `FALSE` for radians
`tl`	tick length (value between 0 and 1)
`...`	optional arguments to be passed on to the generic `matlines` function

Value

no return value

Examples

```
data(striations,package='geostats')
circle.plot(striations,degrees=TRUE)
```

`circle.points` *add points to a circular plot*

Description

Adds directional data as points on an existing circle plot, with angles plotting in a clockwise direction from the top.

Usage

```
circle.points(a, degrees = FALSE, ...)
```

Arguments

`a`	angle(s), scalar or vector
`degrees`	logical. `TRUE` for degrees, `FALSE` for radians
`...`	optional arguments to be passed on to the generic `points` function

Value

no return value

Examples

```
data(striations,package='geostats')
circle.plot(striations,degrees=TRUE)
md <- meanangle(striations,degrees=TRUE)
circle.points(md,pch=22,bg='black',degrees=TRUE)
```

clr — *centred logratio transformation*

Description

Maps compositional data from an n-dimensional simplex to an n-dimensional Euclidean space with Aitchison's centred logratio transformation.

Usage

```
clr(dat, inverse = FALSE)
```

Arguments

`dat`	an `n` x `m` matrix
`inverse`	logical. If `TRUE`, applies the inverse clr tranformation

Value

an `n` x `m` matrix

Examples

```
xyz <- rbind(c(0.03,99.88,0.09),
             c(70.54,25.95,3.51),
             c(72.14,26.54,1.32))
colnames(xyz) <- c('a','b','c')
rownames(xyz) <- 1:3
pc <- prcomp(clr(xyz))
biplot(pc)
```

colourplot — *colour plot*

Description

Adds a colour bar to a scatter plot and/or filled contour plot. This function, which is based on base `R`'s `filled.contour` function, is useful for visualising kriging results.

Usage

```
colourplot(
  x,
  y,
  z,
  X,
  Y,
  Z,
  nlevels = 20,
  colspec = hcl.colors,
  pch = 21,
  cex = 1,
  plot.title,
  plot.axes,
  key.title,
  key.axes,
  asp = NA,
  xaxs = "i",
  yaxs = "i",
  las = 1,
  axes = TRUE,
  frame.plot = axes,
  extra,
  ...
)
```

Arguments

`x`	numerical vector of n equally spaced values to be used in the contour plot.
`y`	numerical vector of m equally spaced values to be used in the contour plot.
`z`	an $n \times m$ matrix of numerical values to be used in the contour plot.
`X`	numerical vector of N values to be used in the scatter plot.
`Y`	numerical vector of N values to be used in the scatter plot.
`Z`	numerical vector of N values to be used in the scatter plot.
`nlevels`	number of levels to be used in the contour plot.
`colspec`	colour specification (e.g., `rainbow`, `grey.colors`, `heat.colors`, `topo.colors`).
`pch`	plot character (`21` – `25`).
`cex`	plot character magnification.
`plot.title`	statements that add titles to the main plot.
`plot.axes`	statements that draw axes on the main plot. This overrides the default axes.
`key.title`	statements that add titles for the plot key.
`key.axes`	statements that draw axes on the plot key. This overrides the default axis.
`asp`	the y/x aspect ratio, see `plot.window`.
`xaxs`	the x axis style. The default is to use internal labelling.

`yaxs`	the y axis style. The default is to use internal labelling.
`las`	the style of labelling to be used. The default is to use horizontal labelling.
`axes`	logicals indicating if axes should be drawn.
`frame.plot`	logicals indicating if a box should be drawn, as in `plot.default`.
`extra`	(optional) extra intructions to be carried out in the main plot window, such as text annotations.
`...`	additional graphical parameters

Value

no return value

Examples

```
data('meuse',package='geostats')
colourplot(X=meuse$x,Y=meuse$y,Z=log(meuse$zinc),
           plot.title=title(main='Meuse',xlab='Easting',ylab='Northing'),
           key.title=title(main='log(Zn)'))
```

`Corsica` *rivers on Corsica*

Description

A 512×512 pixel image of the river network on Corsica.

Examples

```
data(Corsica,package='geostats')
p <- par(mfrow=c(1,2))
image(Corsica)
fractaldim(Corsica)
par(p)
```

`countQuakes` *count the number of earthquakes per year*

Description

Counts the number of earthquakes per year that fall within a certain time interval.

Usage

```
countQuakes(qdat, minmag, from, to)
```

Arguments

`qdat`	a data frame containing columns named `mag` and `year`.
`minmag`	minimum magnitude
`from`	first year
`to`	last year

Value

a table with the number of earthquakes per year

Examples

```
data(declustered,package='geostats')
quakesperyear <- countQuakes(declustered,minmag=5.0,from=1917,to=2016)
table(quakesperyear)
```

`declustered` *declustered earthquake data*

Description

Dataset of 28267 earthquakes between 1769 and 2016, with aftershocks and precursor events removed.

References

Mueller, C.S., 2019. Earthquake catalogs for the USGS national seismic hazard maps. Seismological Research Letters, 90(1), pp.251-261.

Examples

```
data(declustered,package='geostats')
quakesperyear <- countQuakes(declustered,minmag=5.0,from=1917,to=2016)
table(quakesperyear)
```

`DZ` *detrital zircon U-Pb data*

Description

Detrital zircon U-Pb data of 13 sand samples from China.

References

Vermeesch, P. "Multi-sample comparison of detrital age distributions." Chemical Geology 341 (2013): 140-146.

Examples

```
data(DZ,package='geostats')
qqplot(DZ[['Y']],DZ[['5']])
```

`earthquakes` *earthquake data*

Description

Dataset of 20000 earthquakes between 2017 and 2000, downloaded from the USGS earthquake database (https://earthquake.usgs.gov/earthquakes/search/).

Examples

```
data(earthquakes,package='geostats')
gutenberg(earthquakes$mag)
```

`ellipse` *ellipse*

Description

Compute the x-y coordinates of an error ellipse.

Usage

```
ellipse(mean, cov, alpha = 0.05, n = 50)
```

Arguments

`mean`	two-element vector with the centre of the ellipse
`cov`	the 2 x 2 covariance matrix of `x` and `y`
`alpha`	confidence level of the ellipse
`n`	the number of points at which the ellipse is evaluated

Value

a two-column matrix of plot coordinates

Examples

```
X <- rnorm(100,mean=100,sd=1)
Y <- rnorm(100,mean=100,sd=1)
Z <- rnorm(100,mean=100,sd=5)
dat <- cbind(X/Z,Y/Z)
plot(dat)
ell <- ellipse(mean=colMeans(dat),cov=cov(dat))
polygon(ell)
```

`exp` *exponential transformation*

Description

Map a logged kernel density estimate from $[-\infty, +\infty]$ to $[0, \infty]$ by taking exponents.

Usage

```
## S3 method for class 'density'
exp(x)
```

Arguments

`x`	an object of class `density`

Value

an object of class `density`

Examples

```
data(catchments,package='geostats')
lc <- log(catchments$CaMg)
ld <- density(lc)
d <- exp(ld)
plot(d)
```

`FAM` *A-F-M data*

Description

FeO - (Na_2O + K_2O) - MgO compositions of 630 calc-alkali basalts from the Cascade Mountains and 474 tholeiitic basalts from Iceland. Arranged in F-A-M order instead of A-F-M for consistency with the `ternary` function.

Examples

```
data(FAM,package='geostats')
ternary(FAM[,-1])
```

`fault` *fault orientation data*

Description

Ten paired strike and dip measurements (in degrees), drawn from a von Mises - Fisher distribution with mean vector $\mu = \{-1, -1, 1\}/\sqrt{3}$ and concentration parameter $\kappa = 100$.

Examples

```
data(fault,package='geostats')
stereonet(trd=fault$strike,plg=fault$dip,option=2,degrees=TRUE,show.grid=FALSE)
```

`Finland` *Finnish lake data*

Description

Table of 2327 Finnish lakes, extracted from a hydroLAKES database.

References

Lehner, B., and Doll, P. (2004), Development and validation of a global database of lakes, reservoirs and wetlands, Journal of Hydrology, 296(1), 1-22, doi: 10.1016/j.jhydrol.2004.03.028.

Examples

```
data(Finland,package='geostats')
sf <- sizefrequency(Finland$area)
size <- sf[,'size']
freq <- sf[,'frequency']
plot(size,freq,log='xy')
fit <- lm(log(freq) ~ log(size))
lines(size,exp(predict(fit)))
```

forams *foram count data*

Description

Planktic foraminifera counts in surface sediments in the Atlantic ocean.

Examples

```
data(forams,package='geostats')
abundant <- forams[,c('quinqueloba','pachyderma','incompta',
                      'glutinata','bulloides')]
other <- rowSums(forams[,c('uvula','scitula')])
dat <- cbind(abundant,other)
chisq.test(dat)
```

fractaldim *calculate the fractal dimension*

Description

Performs box counting on a matrix of 0s and 1s.

Usage

```
fractaldim(mat, plot = TRUE, ...)
```

Arguments

`mat`	a square matrix of 0s and 1s. Size must be a power of 2.
`plot`	logical. If `TRUE`, plots the results on a log-log scale.
`...`	optional arguments to the generic `points` function.

Value

an object of class `lm`

Examples

```
g <- sierpinski(n=5)
fractaldim(g)
```

fractures *fractures*

Description

A 512×512 pixel image of a fracture network.

Examples

```
data(fractures,package='geostats')
p <- par(mfrow=c(1,2))
image(fractures)
fractaldim(fractures)
par(p)
```

geostats *library(geostats)*

Description

A list of documented functions may be viewed by typing `help(package='geostats')`. Detailed instructions are provided at https://github.com/pvermees/geostats/.

Author(s)

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

See Also

Useful links:

- https://github.com/pvermees/geostats/

gutenberg *create a Gutenberg-Richter plot*

Description

Calculate a semi-log plot with earthquake magnitude on the horizontal axis,and the cumulative number of earthquakes exceeding any given magnitude on the vertical axis.

Usage

```
gutenberg(m, n = 10, ...)
```

Arguments

`m`	a vector of earthquake magnitudes
`n`	the number of magnitudes to evaluate
`...`	optional arguments to the generic `points` function.

Value

the output of `lm` with earthquake magnitude as the independent variable (`mag`) and the logarithm (base 10) of the frequency as the dependent variable (`lfreq`).

Examples

```
data(declustered,package='geostats')
gutenberg(declustered$mag)
```

hills *hills*

Description

150 X-Y-Z values for a synthetic landscape that consists of three Gaussian mountains.

Examples

```
data(hills,package='geostats')
semivariogram(x=hills$X,y=hills$Y,z=hills$Z,model='gaussian')
```

koch *Koch snowflake*

Description

Calculates or plots a Koch set of fractal lines, which is generated using a recursive algorithm that is built on a triangular hat shaped line segment. Each level of recursion replaces each linear segment by the same pattern.

Usage

```
koch(n = 4, plot = TRUE, res = 512)
```

Arguments

n	an integer value controling the number of recursive levels.
plot	logical. If TRUE, the Koch flake is plotted.
res	the number of pixels in each side of the output matrix

Value

a res x res matrix with 0s and 1s

Examples

```
koch()
```

kriging *kriging*

Description

Ordinary kriging interpolation of spatial data. Implements a simple version of ordinary kriging that uses all the data in a training set to predict the z-value of some test data, using a semivariogram model generated by the semivariogram function.

Usage

```
kriging(x, y, z, xi, yi, svm, grid = FALSE, err = FALSE)
```

Arguments

x	numerical vector of training data
y	numerical vector of the same length as x
z	numerical vector of the same length as x
xi	scalar or vector with the x-coordinates of the points at which the z-values are to be evaluated.
yi	scalar or vector with the y-coordinates of the points at which the z-values are to be evaluated.
svm	output of the semivariogram function, a 3-element vector with the sill, nugget and range of the semivariogram fit.
grid	logical. If TRUE, evaluates the kriging interpolator along a regular grid of values defined by xi and yi.
err	logical. If TRUE, returns the variance of the kriging estimate.

Value

either a vector (if grid=FALSE) or a matrix (if grid=TRUE) of kriging interpolations. In the latter case, values that are more than 10% out of the data range are given NA values.

Examples

```
data(meuse,package='geostats')
x <- meuse$x
y <- meuse$y
z <- log(meuse$cadmium)
svm <- semivariogram(x=x,y=y,z=z)
kriging(x=x,y=y,z=z,xi=179850,yi=331650,svm=svm,grid=TRUE)
```

---

ksdist *Kolmogorov-Smirnov distance matrix*

Description

Given a list of numerical vectors, fills a square matrix with Kolmogorov-Smirnov statistics.

Usage

```
ksdist(dat)
```

Arguments

dat	a list of numerical data vectors

Value

an object of class dist

Examples

```
data(DZ,package='geostats')
d <- ksdist(DZ)
mds <- cmdscale(d)
plot(mds,type='n')
text(mds,labels=names(DZ))
```

---

logit *logistic transformation*

Description

Maps numbers from [0,1] to $[-\infty, +\infty]$ and back.

Usage

```
logit(x, ...)

## Default S3 method:
logit(x, inverse = FALSE, ...)

## S3 method for class 'density'
logit(x, inverse = TRUE, ...)
```

Arguments

x	a vector of real numbers (strictly positive if `inverse=FALSE`) or an object of class `density`.
...	optional arguments to the `log` function.
inverse	logical. If `inverse=FALSE`, returns $\ln\left[\frac{x}{1-x}\right]$; otherwise returns $\frac{\exp[x]}{\exp[x]+1}$.

Value

a vector with the same length of `x`

Examples

```
data(catchments,package='geostats')
lp <- logit(catchments$vegetation/100,inverse=FALSE)
ld <- density(lp)
d <- logit(ld,inverse=TRUE)
plot(d)
```

major — *composition of Namib dune sand*

Description

Major element compositions of 16 Namib sand samples.

References

Vermeesch, P. & Garzanti, E. “Making geological sense of ‘Big Data’ in sedimentary provenance analysis.” Chemical Geology 409 (2015): 20-27.

Examples

```
data(major,package='geostats')
comp <- clr(major)
pc <- prcomp(comp)
biplot(pc)
```

meanangle — *mean angle*

Description

Computes the vector mean of a collection of circular measurements.

Usage

```
meanangle(trd, plg = 0, option = 0, degrees = FALSE, orientation = FALSE)
```

Arguments

`trd`	trend angle, in degrees, between 0 and 360 (if `degrees=TRUE`) or between 0 and 2π (if `degrees=FALSE`).
`plg`	(optional) plunge angle, in degrees, between 0 and 90 (if `degrees=TRUE`) or between 0 and 2π (if `degrees=FALSE`).
`option`	scalar. If `option=0`, then `plg` is ignored and the measurements are considered to be circular; if `option=1`, then `trd` is the azimuth and `plg` is the dip; if `option=2`, then `trd` is the strike and `plg` is the dip; if `option=3` then `trd` is the longitude and `plg` is the latitude.
`degrees`	`TRUE` for degrees, `FALSE` for radians
`orientation`	logical. If `TRUE`, estimates the mean angle by eigen decomposition rather than by vector summation. This is the right thing to do for orientation data in which, for example, an angle of 45 degrees is equivalent to an angle of 225 degrees.

Value

a scalar of 2-element vector with the mean orientation, either in radians (if `degrees=FALSE`), or in degrees.

Examples

```
data(striations,package='geostats')
meanangle(striations,degrees=TRUE)
```

`meuse` *Meuse river data set*

Description

This data set gives locations and topsoil heavy metal concentrations, collected in a flood plain of the river Meuse, near the village of Stein (NL). Heavy metal concentrations are from composite samples of an area of approximately 15 m x 15 m. This version of the `meuse` dataset is a trimmed down version of the eponymous dataset from the `sp` dataset.

Examples

```
data(meuse,package='geostats')
semivariogram(x=meuse$x,y=meuse$y,z=log(meuse$cadmium))
```

`Mode` *get the mode of a dataset*

Description

Computes the most frequently occuring value in a sampling distribution.

Usage

```
Mode(x, categorical = FALSE)
```

Arguments

`x`	a vector
`categorical`	logical. If `TRUE`, returns the most frequently occuring value for categorical variables. If `FALSE`, returns the value corresponding to the maximimum kernel density for continuous variables

Value

a scalar

Examples

```
data(catchments,package='geostats')
m1 <- Mode(catchments$CaMg,categorical=TRUE)

m2 <- 1:50
for (i in m2){
   m2[i] <- Mode(rnorm(100),categorical=FALSE)
}
hist(m2)
```

`palaeomag` *palaeomagnetic data*

Description

Ten paired magnetic declination (azimuth) and inclination (dip) measurements, drawn from a von Mises - Fisher distribution with mean vector $\mu = \{2, 2, 1\}/3$ and concentration parameter $\kappa = 200$.

Examples

```
data(palaeomag,package='geostats')
stereonet(trd=palaeomag$decl,plg=palaeomag$incl,degrees=TRUE,show.grid=FALSE)
```

`PCA2D` *Principal Component Analysis of 2D data*

Description

Produces a 4-panel summary plot for two dimensional PCA for didactical purposes.

Usage

```
PCA2D(X)
```

Arguments

`X`	a matrix with two columns

Examples

```
X <- rbind(c(-1,7),c(3,2),c(4,3))
colnames(X) <- c('a','b')
PCA2D(X)
```

pebbles *pebble orientations*

Description

Orientations (in degrees) of 20 pebbles.

Examples

```
data(pebbles,package='geostats')
circle.plot(pebbles,degrees=TRUE)
m <- meanangle(pebbles,option=0,orientation=TRUE)
circle.points(m,degrees=TRUE,pch=22,bg='white')
```

pendulum *3-magnet pendulum experiment*

Description

Simulates the 3-magnet pendulum experiment, starting at a specified position with a given start velocity.

Usage

```
pendulum(
  startpos = c(-2, 2),
  startvel = c(0, 0),
  src = rbind(c(0, 0), c(0.5, sqrt(0.75)), c(1, 0)),
  plot = TRUE
)
```

Arguments

startpos	2-element vecotor with the initial position
startvel	2-element vector with the initial velocity
src	$n \times 2$ matrix with the positions of the magnets
plot	logical. If TRUE, generates a plot with the trajectory of the pendulum.

Value

the end position of the pendulum

Examples

```
p <- par(mfrow=c(1,2))
pendulum(startpos=c(2.1,2))
pendulum(startpos=c(1.9,2))
par(p)
```

randy — *generate bivariate random data*

Description

Returns bivariate datasets from four synthetic distributions that have the shape of a circle, arrow, square and ellipse.

Usage

```
randy(pop = 1, n = 250)
```

Arguments

pop	an integer from 1 to 4 marking the population of choice: 1 = circle, 2 = arrow, 3 = solid square, 4 = ellipse.
n	the number of random draws to be drawn from population pop

Value

a [2xn] matrix of random numbers

Examples

```
p <- par(mfrow=c(1,4))
for (i in 1:4){
   plot(randy(pop=i))
}
par(p)
```

Rbar — *calculate $\bar{R}$*

Description

Given n circular or spherical measurements, the length of their normalised vector sum ($\bar{R}$) serves as a measure of directional concentration.

Usage

```
Rbar(trd, plg = 0, option = 0, degrees = FALSE)
```

Arguments

`trd`	trend angle, in degrees, between 0 and 360 (if `degrees=TRUE`) or between 0 and 2π (if `degrees=FALSE`).
`plg`	(optional) plunge angle, in degrees, between 0 and 90 (if `degrees=TRUE`) or between 0 and 2π (if `degrees=FALSE`).
`option`	scalar. If `option=0`, then `plg` is ignored and the measurements are considered to be circular; if `option=1`, then `trd` is the azimuth and `plg` is the dip; if `option=2`, then `trd` is the strike and `plg` is the dip; if `option=3` then `trd` is the longitude and `plg` is the latitude.
`degrees`	`TRUE` for degrees, `FALSE` for radians

Value

a value between 0 and 1

Examples

```
data(striations,package='geostats')
Rbar(striations,degrees=TRUE)
```

Rbar2kappa — *$\bar{R}$ to κ conversion*

Description

Converts the empirical concentration parameter $\bar{R}$ to the von-Mises concentration parameter κ.

Usage

```
Rbar2kappa(R, p = 1)
```

Arguments

`R`	a scalar or vector of values between 0 and 1
`p`	the number of parameters

Details

$\bar{R}$ and κ are two types of concentration parameter that are commonly used in directional data analysis. κ is one of the parameters of the parametric von Mises distribution, which is difficult to estimate from the data. $\bar{R}$ is easier to calculate from data. `Rbar2kappa` converts $\bar{R}$ to $\bar{\kappa}$ using the following approximate empirical formula:

$\kappa = \frac{\bar{R}(p+1-\bar{R}^2)}{1-\bar{R}^2}$

where p marks the number of parameters in the data space (1 for circle, 2 for a sphere).

Value

value(s) between 0 and $+\infty$

References

Banerjee, A., et al. "Clustering on the unit hypersphere using von Mises-Fisher distributions." Journal of Machine Learning Research 6.Sep (2005): 1345-1382.

Examples

```
data(striations,package='geostats')
Rbar2kappa(Rbar(striations,degrees=TRUE))
```

---

rbsr *Rb-Sr data*

Description

Synthetic dataset of 8 Rb-Sr analysis that form a 1Ga isochron.

Examples

```
data(rbsr,package='geostats')
plot(rbsr[,'RbSr'],rbsr[,'SrSr'])
fit <- lm(SrSr ~ RbSr,data=rbsr)
abline(fit)
```

---

rwyxz *Spurious correlation*

Description

Calculate the 'null correlation' of ratios, using the the spurious correlation formula of Pearson (1897).

Usage

```
rwyxz(
  mw,
  mx,
  my,
  mz,
  sw,
  sx,
  sy,
  sz,
```

```
  rwx = 0,
  rwy = 0,
  rwz = 0,
  rxy = 0,
  rxz = 0,
  ryz = 0
)

ryxy(mx, my, sx, sy, rxy = 0)

rxzyz(mx, my, mz, sx, sy, sz, rxy = 0, rxz = 0, ryz = 0)
```

Arguments

mw	the mean of variable w
mx	the mean of variable x
my	the mean of variable y
mz	the mean of variable z
sw	the standard deviation of variable w
sx	the standard deviation of variable x
sy	the standard deviation of variable y
sz	the standard deviation of variable z
rwx	the correlation coefficient between w and x
rwy	the correlation coefficient between w and y
rwz	the correlation coefficient between w and z
rxy	the correlation coefficient between x and y
rxz	the correlation coefficient between x and z
ryz	the correlation coefficient between y and z

Value

the null correlation coefficient

References

Pearson, K. "Mathematical contributions to the theory of evolution. – on a form of spurious correlation which may arise when indices are used in the measurement of organs." Proceedings of the Royal Society of London 60.359-367 (1897): 489-498.

Examples

```
rxzyz(mx=100,my=100,mz=100,sx=1,sy=1,sz=10)
```

`semivariogram` *semivariogram*

Description

Plots the semivariance of spatial data against inter-sample distance, and fits a spherical equation to it.

Usage

```
semivariogram(
  x,
  y,
  z,
  bw = NULL,
  nb = 13,
  plot = TRUE,
  fit = TRUE,
  model = c("spherical", "exponential", "gaussian"),
  ...
)
```

Arguments

`x`	numerical vector
`y`	numerical vector of the same length as `x`
`z`	numerical vector of the same length as `x`
`bw`	(optional) the bin width of the semivariance search algorithm
`nb`	(optional) the maximum number of bins to evaluate
`plot`	logical. If `FALSE`, suppresses the graphical output
`fit`	logical. If `TRUE`, returns the sill, nugget and range.
`model`	the parametric model to fit to the empirical semivariogram (only used if `fit=TRUE`).
`...`	optional arguments to be passed on to the generic `plot` function

Value

returns a list with the estimated semivariances at different distances for the data, and (if `fit=TRUE`), a vector with the sill, nugget and range.

Examples

```
data(meuse,package='geostats')
semivariogram(x=meuse$x,y=meuse$y,z=log(meuse$cadmium))
```

`sierpinski` *Sierpinski carpet*

Description

Returns a matrix of 0s and 1s that form a Sierpinski carpet. This is a two dimensional fractal, which is generated using a recursive algorithm that is built on a grid of eight black squares surrounding a white square. Each level of recursion replaces each black square by the same pattern.

Usage

```
sierpinski(n = 5)
```

Arguments

`n`	an integer value controling the number of recursive levels.

Value

a square matrix with 0s and 1s.

Examples

```
g <- sierpinski(n=5)
image(g,col=c('white','black'),axes=FALSE,asp=1)
```

`sizefrequency` *calculate the size-frequency distribution of things*

Description

Count the number of items exceeding a certain size.

Usage

```
sizefrequency(dat, n = 10, log = TRUE)
```

Arguments

`dat`	a numerical vector
`n`	the number of sizes to evaluate
`log`	logical. If `TRUE`, uses a log spacing for the sizes at which the frequencies are evaluated

Value

a data frame with two columns `size` and `frequency`

Examples

```
data(Finland,package='geostats')
sf <- sizefrequency(Finland$area)
plot(frequency~size,data=sf,log='xy')
fit <- lm(log(frequency) ~ log(size),data=sf)
lines(x=sf$size,y=exp(predict(fit)))
```

`skew` *calculate the skewness of a dataset*

Description

Compute the third moment of a sampling distribution.

Usage

```
skew(x)
```

Arguments

`x`	a vector

Value

a scalar

Examples

```
data(catchments,package='geostats')
skew(catchments$vegetation)
```

stereonet *stereonet*

Description

Plots directional data on a Wulff or Schmidt stereonet. The Wulff equal angle polar Lambert projection preserves the shape of objects and is often used to visualise structural data. The Schmidt equal area polar Lambert projection preserves the size of objects and is more popular in mineralogy.

Usage

```
stereonet(
  trd,
  plg,
  coneAngle = rep(10, length(trd)),
  option = 1,
  wulff = TRUE,
  add = FALSE,
  degrees = FALSE,
  show.grid = TRUE,
  grid.col = "grey50",
  tl = 0.05,
  type = "p",
  labels = 1:length(trd),
  pch = 21,
  bg = c("black", "white"),
  lty = c(1, 2),
  ...
)
```

Arguments

`trd`	trend angle, in degrees, between 0 and 360 (if `degrees=TRUE`) or between 0 and 2π (if `degrees=FALSE`).
`plg`	plunge angle, in degrees, between 0 and 90 (if `degrees=TRUE`) or between 0 and 2π (if `degrees=FALSE`).
`coneAngle`	if `option=4`, controls the radius of a small circle around the pole with azimuth `trd` and dip `plg`.
`option`	scalar. If `option=1` or `option=4`, then `trd` is the azimuth and `plg` is the dip; if `option=2`, then `trd` is the strike and `plg` is the dip; if `option=3`, then `trd` is the longitude and `plg` is the latitude.
`wulff`	logical. If `FALSE`, produces a Schmidt net.
`add`	logical. If `TRUE`, adds to an existing stereonet.
`degrees`	logical. If `FALSE`, assumes that `azimuth` and `dip` are in radians.
`show.grid`	logical. If `TRUE`, decorates the plot with a grid of great and small circles.

`grid.col`	colour of the grid.
`tl`	tick length for the N, E, S, W markers (value between 0 and 1). Set to 0 to omit the markers.
`type`	if `option=1` or `3`, coordinates can be visualsed as points (`type='p'`), lines (`type='l'`) or decorated with text labels (`type='t'`).
`labels`	if `option=1` or `3` and `type='t'`, specifies the text labels to be used to mark the measurements on the stereonet.
`pch`	plot character: see 'points'.
`bg`	background colours of the plot characters. Vector of two colours, which are used to mark points that plot below and above the projection plane of the stereonet, respectively. Only relevant if `pch` falls in the range from 21:25.
`lty`	line type. Vector of two numbers, which are used to plot lines below and above the projection plane of the stereonet, respectively.
`...`	optional arguments to be passed on to the generic `points` function

Author(s)

based on a MATLAB script written by Nestor Cardozo.

References

Allmendinger, R.W., Cardozo, N., and Fisher, D.M. "Structural geology algorithms: Vectors and tensors". Cambridge University Press, 2011.

Examples

```
stereonet(trd=c(120,80),plg=c(10,30),degrees=TRUE,pch=16)
stereonet(trd=c(120,80),plg=c(10,30),degrees=TRUE,
          option=4,coneAngle=c(5,10),add=TRUE)
```

`striations` *directions of glacial striations*

Description

Directions (in degrees) of 30 glacial striation measurements from Madagascar.

Examples

```
data(striations,package='geostats')
circle.plot(striations,degrees=TRUE)
```

ternary — *ternary diagrams*

Description

Plot points, lines or text on a ternary diagram.

Usage

```
ternary(xyz = NULL, f = rep(1, 3), labels, add = FALSE, type = "p", ...)
```

Arguments

`xyz`	an `n x 3` matrix or data frame
`f`	a three-element vector of multipliers for `xyz`
`labels`	the text labels for the corners of the ternary diagram
`add`	if `TRUE`, adds information to an existing ternary diagram
`type`	one of `'n'` (empty plot), `'p'` (points), `'l'` (lines) or `'t'` (text).
`...`	optional arguments to the `points`, `lines` or `text` functions.

Examples

```
data(ACNK,package='geostats')
ternary(ACNK,type='p',labels=c(expression('Al'[2]*'O'[3]),
                               expression('CaO+Na'[2]*'O'),
                               expression('K'[2]*'O')))
```

test — *composition of a further 147 oceanic basalts*

Description

Major element compositions of 64 island arc basalts (IAB), 23 mid oceanic ridge basalts (MORB) and 60 ocean island basalts (OIB). This dataset can be used to test supervised learning algorithms.

References

Vermeesch, P. "Tectonic discrimination diagrams revisited." Geochemistry, Geophysics, Geosystems 7.6 (2006).

Examples

```
library(MASS)
data(training,package='geostats')
ld <- lda(x=alr(training[,-1]),grouping=training[,1])
data(test,package='geostats')
pr <- predict(ld,newdata=alr(test[,-1]))
table(test$affinity,pr$class)
```

---

training *composition of 646 oceanic basalts*

Description

Major element compositions of 227 island arc basalts (IAB), 221 mid oceanic ridge basalts (MORB) and 198 ocean island basalts (OIB). This dataset can be used to train supervised learning algorithms.

References

Vermeesch, P. “Tectonic discrimination diagrams revisited.” Geochemistry, Geophysics, Geosystems 7.6 (2006).

Examples

```
library(MASS)
data(training,package='geostats')
ld <- lda(x=alr(training[,-1]),grouping=training[,1])
pr <- predict(ld)
table(training$affinity,pr$class)
```

---

vonMises *von Mises distribution*

Description

Returns the probability density of a von Mises distribution, which describes probability distributions on a circle using the following density function:

$$\frac{\exp(\kappa\cos(x-\mu))}{2\pi I_0(\kappa)}$$

where $I_0(\kappa)$ is a zero order Bessel function.

Usage

```
vonMises(a, mu = 0, kappa = 1, degrees = FALSE)
```

Arguments

a	angle(s), scalar or vector
mu	scalar containing the mean direction
kappa	scalar containing the concentration parameter
degrees	TRUE for degrees, FALSE for radians

Value

a scalar or vector of the same length as angles

Examples

```
plot(x=c(-1,1.2),y=c(-1,1.2),type='n',
     axes=FALSE,ann=FALSE,bty='n',asp=1)
a <- seq(from=-pi,to=pi,length.out=200)
d <- vonMises(a=a,mu=pi/4,kappa=5)
symbols(x=0,y=0,circles=1,add=TRUE,inches=FALSE,xpd=NA,fg='grey50')
lines(x=(1+d)*cos(a),y=(1+d)*sin(a),xpd=NA)
```

---

worldpop *world population*

Description

The world population from 1750 until 2014.

Examples

```
data(worldpop,package='geostats')
plot(worldpop)
```

---

xyz2xy *get x,y plot coordinates of ternary data*

Description

Helper function to generate bivariate plot coordinates for ternary data.

Usage

```
xyz2xy(xyz)
```

Arguments

xyz	an n x 3 matrix or data frame

Value

an `n x 2` numerical matrix

Examples

```
xyz <- rbind(c(1,0,0),c(0,1,0),c(0,0,1),c(1,0,0))
xy <- xyz2xy(xyz)
plot(xy,type='l',bty='n')
```

---

york — *Linear regression of X,Y-variables with correlated errors*

Description

Implements the unified regression algorithm of York et al. (2004) which, although based on least squares, yields results that are consistent with maximum likelihood estimates of Titterington and Halliday (1979).

Usage

```
york(dat, alpha = 0.05, plot = TRUE, fill = NA, ...)
```

Arguments

`dat`	a 4 or 5-column matrix with the X-values, the analytical uncertainties of the X-values, the Y-values, the analytical uncertainties of the Y-values, and (optionally) the correlation coefficients of the X- and Y-values.
`alpha`	cutoff value for confidence intervals.
`plot`	logical. If true, creates a scatter plot of the data with the best fit line shown on it.
`fill`	the fill colour of the error ellipses. For additional plot options, use the `IsoplotR` package.
`...`	optional arguments for the scatter plot.

Details

Given n pairs of (approximately) collinear measurements X_i and Y_i (for $1 \leq i \leq n$), their uncertainties $s[X_i]$ and $s[Y_i]$, and their covariances cov$[X_i, Y_i]$, the `york` function finds the best fitting straight line using the least-squares algorithm of York et al. (2004). This algorithm is modified from an earlier method developed by York (1968) to be consistent with the maximum likelihood approach of Titterington and Halliday (1979).

Value

A two-element list of vectors containing:

coef the intercept and slope of the straight line fit

cov the covariance matrix of the coefficients

References

Titterington, D.M. and Halliday, A.N., 1979. On the fitting of parallel isochrons and the method of maximum likelihood. Chemical Geology, 26(3), pp.183-195.

York, Derek, et al., 2004. Unified equations for the slope, intercept, and standard errors of the best straight line. American Journal of Physics 72.3, pp.367-375.

Examples

```
data(rbsr,package='geostats')
fit <- york(rbsr)
```

Index