Package ‘geostats’

December 30, 2020

Title An Introduction to Statistics for Geoscientists

Version 1.0

Date 2020-12-30

Description

Provides example datasets and code for the introductory statistics module for geoscientists at University College London (UCL). Includes functionality for compositional data, fractals and chaos.

Author Pieter Vermeesch [aut, cre]

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

Depends R (>= 3.5.0)

Imports MASS

License GPL-3

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

R topics documented:

ACNK — *A-CN-K compositions*

Description

Synthetic A (Al2O3), CN (CaO+Na2O), K (K2O) 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')))
```

AFM — *A-F-M data*

Description

(Na2O + K2O) - FeO - MgO compositions of 630 calc-alkali basalts from the Cascade Mountains and 474 tholeiitic basalts from Iceland.

Examples

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

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` x `m` matrix
`inverse`	if `TRUE`, applies the inverse alr tranformation

Value

if `inverse=FALSE`, returns an `(n-1) x m` matrix of logratios; otherwise returns an `(n+1) x 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
pc <- prcomp(alr(xyz))
biplot(pc)
```

`boxcount` *box counting*

Description

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

Usage

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

Arguments

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

Examples

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

`Britain` *British coast*

Description

a 512 x 512 pixel image of the British coast line

Examples

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

`cantor` *Cantor set*

Description

Calculates or plots a Cantor set of fractal lines.

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`.

Details

The Cantor set 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.

Value

a square matrix with 0s and 1s.

Examples

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

`circle.plot` *plot circular data*

Description

Plots directional data as ticks on a circle

Usage

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

Arguments

`a`	angle(s), scalar or vector
`degrees`	`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

Details

Produces a circle with angles plotting in a clockwise direction from the top

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

Usage

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

Arguments

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

Details

adds points to a circle with angles plotting in a clockwise direction from the top

Examples

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

`clasts` *clast size data*

Description

20 clast size measurements, in cm

Examples

```
data(clasts,package='geostats')
d <- density(log(clasts))
plot(d)
```

`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`	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

combines a filled contour plot and filled scatter plot for 3-dimensional measurements

Usage

```
colourplot(
  x,
  y,
  z,
  X,
  Y,
  Z,
  nlevels = 20,
  colspec = rainbow,
  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`	if `levels` is not specified, the range of `z`, values is divided into approximately this many levels.
`colspec`	colour specification (e.g., `rainbow`, `hsv`, `hcl`, `rgb`)
`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 labeling.
`yaxs`	the y axis style. The default is to use internal labeling.
`las`	the style of labeling to be used. The default is to use horizontal labeling.
`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

Details

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.

Examples

```
data('meuse',package='geostats')
colourplot(X=meuse$x,Y=meuse$y,Z=log(meuse$zinc))
```

`Corsica` *rivers on Corsica*

Description

a 512 x 512 pixel image of the river network on Corsica

Examples

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

countQuakes *count the number of earthquakes per year*

Description

counts the number of earthquakes per year that fall between two magnitude limits

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

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 5 sand samples from China

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 confidence ellipse
`n`	the number of points at which the ellipse is evaluated

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 the input 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(clasts,package='geostats')
lc <- log(clasts)
ld <- density(lc)
d <- exp(ld)
plot(d)
```

`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(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.

Examples

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

`fractures` *fractures*

Description

a 512 x 512 pixel image of a fracture network

Examples

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

`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>

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

This data set contains 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)
```

koch — *Koch snowflake*

Description

Calculates or plots a Koch set of fractal lines.

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

Details

The Koch set 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.

Value

a `res` x `res` matrix with 0s and 1s

Examples

```
k <- koch(n=5)
d <- fractaldim(k,plot=FALSE)
print(d)
```

kriging — *kriging*

Description

ordinary kriging interpolation of spatial data

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 x-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.

Details

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, given a spherical variogram.

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)
xi <- seq(from=min(x),to=max(x),length.out=50)
yi <- seq(from=min(y),to=max(y),length.out=50)
zi <- kriging(x=x,y=y,z=z,xi=xi,yi=yi,svm=svm,grid=TRUE)
contour(xi,yi,zi)
```

---

ksdist *Kolmogorov-Smirnov distance matrix*

Description

fills a square matrix with Kolmogorov-Smirnov statistics

Usage

```
ksdist(dat)
```

Arguments

`dat`	a list of numerical data vectors

Examples

```
data(DZ,package='geostats')
d <- ksdist(DZ)
plot(cmdscale(d))
```

`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`)
`...`	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(porosity,package='geostats')
lp <- logit(porosity,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

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)
```

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` or `4`, then `trd` is the longitude and `plg` is the latitude.
`degrees`	`TRUE` for degrees, `FALSE` for radians

Details

averages angles by taking their vector sum

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

compute 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(clasts,package='geostats')
m1 <- Mode(clasts,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)
```

pendulum *3-magnet pendulum experiment*

Description

simulate the 3-magnet pendulum experiment

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 x 2 matrix with the positions of the magnets
plot	logical. If TRUE, generates a plot with the trajectory of the pendulum.

Details

start a pendulumn at a specified position and with a start velocity.

Value

the end position of the pendulum

Examples

```
par(mfrow=c(1,2))
pendulum(startpos=c(2,2))
pendulum(startpos=c(1.9,2))
```

pH *pH data*

Description

pH measurements in 20 samples of rain water

Examples

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

porosity *porosity data*

Description

20 porosity measurements, as fractions

Examples

```
data(porosity,package='geostats')
plot(density(logit(porosity)))
```

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

returns $\bar{R}$, 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 codeoption=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; if `option=4`, then `trd` is the longitude and `plg` is the latitude.
`degrees`	`TRUE` for degrees, `FALSE` for radians

Details

Given n circular or spherical measurements, their length of the normalised vector sum takes serves as a measure of concentration.

Value

a value between 0 and 1

Examples

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

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

Description

converts concentration parameter $\bar{R}$ to 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 $+\infty$

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

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

Details

Implements the spurious correlation formula of Pearson (1897)

Value

the null correlation coefficient

Examples

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

---

semivariogram *semivariogram*

Description

computes, plots, and fits the semivariogram of spatial data

Usage

```
semivariogram(
  x,
  y,
  z,
  bw = NULL,
  nb = 13,
  plot = TRUE,
  fit = TRUE,
  model = c("spherical", "linear", "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

Details

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

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 fractal.

Usage

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

Arguments

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

Details

The Sierpinski carpet is 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.

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

calculate 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(porosity,package='geostats')
skew(porosity)
```

---

stereonet *stereonet*

Description

Plots directional data on a Schmidt equal area or Wulff equal angle stereonet.

Usage

```
stereonet(
  trd,
  plg,
  coneAngle = 0,
  option = 1,
  wulff = TRUE,
  add = FALSE,
  degrees = FALSE,
  show.grid = TRUE,
  grid.col = "grey50",
  tl = 0.05,
  type = "p",
  labels = 1:length(trd),
  ...
)
```

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`, 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` or `4`, 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.
...	optional arguments to be passed on to the generic points function

Details

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.

Author(s)

based on the Matlab script by Gerry Middleton

Examples

```
stereonet(azimuth=c(120,80),dip=c(10,30),degrees=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 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.

Examples

```
library(MASS)
data(training,package='geostats')
data(test,package='geostats')
qd <- qda(affinity ~ ., data=training)
pr <- predict(qd,newdata=test[,-1])
table(test$affinity,pr$class)
```

training — *composition of 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.

Examples

```
library(MASS)
data(training,package='geostats')
qd <- qda(affinity ~ ., data=training)
pr <- predict(qd)
table(training$affinity,pr$class)
```

vonMises — *von Mises distribution*

Description

returns the probability density of a von Mises distribution

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

Details

the von Mises distribution 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

Value

a scalar or vector of the same length as `angles`

Examples

```
plot(x=c(-1.2,1.2),y=c(-1.2,1.2),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/2,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

```
X <- c(1.550,12.395,20.445,20.435,20.610,24.900,
       28.530,50.540,51.595,86.51,106.40,157.35)
Y <- c(.7268,.7849,.8200,.8156,.8160,.8322,
       .8642,.9584,.9617,1.135,1.230,1.490)
n <- length(X)
sX <- X*0.01
sY <- Y*0.005
rXY <- rep(0.8,n)
dat <- cbind(X,sX,Y,sY,rXY)
fit <- york(dat)
```

Index