Package ‘depth’

January 7, 2017

Type Package

Title Nonparametric Depth Functions for Multivariate Analysis

Version 2.1-1

Date 2017-01-07

Author Maxime Genest <maxime.genest@clevislauzon.qc.ca>, Jean-Claude Masse <jcmasse@mat.ulaval.ca>, Jean-Francois Plante <jfplante@hec.ca>.

Maintainer Jean-Francois Plante <jfplante@hec.ca>

Description Tools for depth functions methodology applied to multivariate analysis. Besides allowing calculation of depth values and depth-based location estimators, the package includes functions or drawing contour plots and perspective plots of depth functions. Euclidian and spherical depths are supported.

Depends R (>= 3.2.0), abind, grDevices, circular, rgl

Suggests robustbase, MASS

License GPL-2

Repository CRAN

Date/Publication 2017-01-07 10:20:43

NeedsCompilation yes

R topics documented:

`depth-package` *Depth functions tools for multivariate analysis*

Description

This is a collection of functions applying depth functions methodology to multivariate analysis. Besides allowing calculation of depth values and depth-based location estimators, the package includes functions for drawing contour plots and perspective plots of depth functions.

Details

Package:	depth
Type:	Package
Version:	2.0
Date:	2012-08-12
License:	GPL-2
LazyLoad:	yes

All functions apply to a multivariate data set. Function `depth` calculates the depth of a point with respect to the data set. Depth functions covered are Tukey's, Liu's and Oja's. Functions `med`, `trmean` and `ctrmean` return depth-based medians, classical-like trimmed means and centroid trimmed means, respectively. Functions `perspdepth` and `isodepth` draw perspective and contour plots, respectively. Functions `sdepth`, `smed`, `strmeasure` and `scontour` give equivalent results for directional data.

Author(s)

Maxime Genest <maxime.genest@clevislauzon.qc.ca>, Jean-Claude Masse <jcmasse@mat.ulaval.ca>, Jean-Francois Plante <jfplante@hec.ca>.

Maintainer: Jean-Francois Plante <jfplante@hec.ca>

References

Liu, R.Y., Parelius, J.M. and Singh, K. (1999), Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion), *Ann. Statist.*, **27**, 783–858.

Liu, R.Y. and Singh, K. (1992), Directional data: Concepts of data depth on circles and spheres, *Ann. Statist.*, **20**, 1468–1484.

Mardia, K.V. and Jupp, E.J. (1999). *Directional Statistics*, Wiley.

Small, C.G. (1990), A survey of multidimensional medians, *Int. Statist. Rev.*, **58**, 263–277.

Zuo, Y. amd Serfling, R. (2000), General Notions of Statistical Depth Functions, *Ann. Statist.*, **28**, no. 2, 461–482.

Examples

```
set.seed(159); library(MASS)
mu1 <- c(0,0); mu2 <- c(6,0); sigma <- matrix(c(1,0,0,1), nc = 2)
mixbivnorm <- rbind(mvrnorm(80, mu1, sigma), mvrnorm(20, mu2, sigma))
depth(c(0,0),mixbivnorm)
med(mixbivnorm)
trmean(mixbivnorm, 0.2)
library(rgl)
perspdepth(mixbivnorm, col = "magenta")
isodepth(mixbivnorm, dpth = c(35,5), col = rainbow(2))
```

`ctrmean` *Centroid trimmed mean*

Description

Computes the centroid of a Tukey depth-based trimmed region.

Usage

```
ctrmean(x ,alpha, eps = 1e-8, mustdith = FALSE, maxdith = 50,
   dithfactor = 10 ,factor = .8)
```

Arguments

`x`	Bivariate data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, both components must be numerical vectors of equal length (coordinates of observations).
`alpha`	Outer trimming fraction (0 to 0.5). Observations whose depth is less than `alpha` to be trimmed.
`eps`	Error tolerance to control the calculation.
`mustdith`	Logical. Should dithering be applied? Used when data set is not in general position or a numerical problem is encountered.
`maxdith`	Positive integer. Maximum number of dithering steps.
`dithfactor`	Scaling factor used for horizontal and vertical dithering.
`factor`	Proportion (0 to 1) of outermost contours computed according to a version of the algorithm ISODEPTH of Rousseeuw and Ruts (1998); remaining contours are derived from an algorithm in Rousseeuw *et al.* (1999).

Details

Dimension 2 only. Centroid trimmed mean is defined to be the centroid of a Tukey depth-based trimmed region relative to the uniform measure. Contours are derived from algorithm ISODEPTH by Ruts and Rousseeuw (1996) or, more exactly, revised versions of this algorithm which appear in Rousseeuw and Ruts (1998) and Rousseeuw *et al.* (1999). Argument `factor` determines which version to use. If n is the number of observations, contours of depth $\leq$ `factor` $n/2$ are obtained from the 1998 version, while the remaining contours are derived from the 1999 version.

When the data set is not in general position, dithering can be used in the sense that random noise is added to each component of each observation. Random noise takes the form `eps` times `dithfactor` times U for the horizontal component and `eps` times `dithfactor` times V for the vertical component, where U, V are independent uniform on [-.5, 5.]. This is done in a number of consecutive steps applying independent U's and V's.

Value

Centroid trimmed mean vector

Author(s)

Jean-Claude Masse and Jean-Francois Plante, based on Fortran code by Ruts and Rousseeuw from University of Antwerp.

References

Masse, J.C. (2008), Multivariate Trimmed means based on the Tukey depth, *J. Statist. Plann. Inference*, in press.

Ruts, I. and Rousseeuw, P.J. (1996), Computing depth contours of bivariate point clouds, *Comput. Statist. Data Anal.*, **23**. 153–168.

Rousseeuw, P.J. and Ruts, I. (1998), Constructing the bivariate Tukey median, *Stat. Sinica*, **8**, 828–839.

Rousseeuw, P.J., Ruts, I., and Tukey, J.W. (1999), The Bagplot: A Bivariate Boxplot, *The Am. Stat.*, **53**, 382–387.

See Also

`med` for multivariate medians and `trmean` for classical-like depth-based trimmed means.

Examples

```
## exact centroid trimmed mean
set.seed(345)
xx <- matrix(rnorm(1000), nc = 2)
ctrmean(xx, .2)

## second example of an exact centroid trimmed mean
set.seed(159); library(MASS)
mu1 <- c(0,0); mu2 <- c(6,0); sigma <- matrix(c(1,0,0,1), nc = 2)
mixbivnorm <- rbind(mvrnorm(80, mu1 ,sigma), mvrnorm(20, mu2, sigma))
ctrmean(mixbivnorm, 0.3)
```

```
## dithering used for data set not in general position
data(starsCYG, package = "robustbase")
ctrmean(starsCYG, .1, mustdith = TRUE)
```

`depth` *Depth calculation*

Description

Computes the depth of a point with respect to a multivariate data set.

Usage

```
depth(u, x, method = "Tukey", approx = FALSE,
   eps = 1e-8, ndir = 1000)
```

Arguments

`u`	Numerical vector whose depth is to be calculated. Dimension has to be the same as that of the observations.
`x`	The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one multivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
`method`	Character string which determines the depth function used. `method` can be "Tukey" (the default), "Liu" or "Oja".
`approx`	Logical. If dimension is 3, should an approximate Tukey depth be computed? Useful when sample size is large.
`eps`	Error tolerance to control the calculation.
`ndir`	Number of random directions used when Tukey depth is approximated.

Details

`method` "Tukey" refers to the Tukey or halfspace depth. In dimension 2, exact calculation is based on Fortran code from Rousseeuw and Ruts (1996). In dimensions higher than 2, calculation utilises Fortran code from Struyf and Rousseeuw (1998). This yields exact calculation when dimension is 3 and `approx = FALSE`, and approximate calculation when dimension is higher than 3.

The Liu (or simplicial) depth is computed in dimension 2 only. Calculation is exact and based on Fortran code from Rousseeuw and Ruts (1996).

The Oja depth is derived from a location measure considered by Oja. If p is the dimension and n the size of the data set, it is defined to be $0.5(1 + \binom{n}{p}^{-1} \sum(\mathrm{Volume}(S(u, x[i_1,], \dots, x[i_p,])))^{-1}$, where $S(args)$ denotes the simplex generated by $args$, and sum and average are taken over all p-plets $x[i_1,], \dots, x[i_p,]$ such that $1 <= i_1 < \dots . < i_p <= n$. Calculation is exact.

Value

Returns the depth of multivariate point u with respect to data set x.

Author(s)

Jean-Claude Masse and Jean-Francois Plante, based on Fortran code by Rousseeuw, Ruts and Struyf from University of Antwerp.

References

Liu, R.Y., Parelius, J.M. and Singh, K. (1999), Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion), *Ann. Statist.*, **27**, 783–858.

Rousseeuw, P.J. and Ruts, I. (1996), AS 307 : Bivariate location depth, *Appl. Stat.-J. Roy. S. C*, **45**, 516–526.

Rousseeuw, P.J. and Struyf, A. (1998), Computing location depth and regression depth in higher dimensions, *Stat. Comput.*, **8**, 193–203.

Zuo, Y. amd Serfling, R. (2000), General Notions of Statistical Depth Functions, *Ann. Statist.*, **28**, no. 2, 461–482.

See Also

perspdepth and isodepth for depth graphics.

Examples

```
## calculation of Tukey depth
data(starsCYG, package = "robustbase")
depth(apply(starsCYG,2,mean), starsCYG)

## Tukey depth applied to a large bivariate data set.
set.seed(356)
x <- matrix(rnorm(9999), nc = 3)
depth(rep(0,3), x)

## approximate calculation much easier
depth(rep(0,3), x, approx = TRUE)
```

---

isodepth *Contour plots for depth functions*

Description

Draws a contour plot of Tukey's depth function.

Usage

```
isodepth(x, dpth = NULL, output = FALSE, twodim = TRUE,
   mustdith = FALSE, maxdith = 50, dithfactor = 10,
   trace.errors = TRUE, eps = 1e-8, factor = 0.8, xlab = "X",
   ylab = "Y", zlab = "Tukey's depth", colcontours = NULL, ...)
```

Arguments

`x`	Bivariate data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, both components must be numerical vectors of equal length (coordinates of observations).
`dpth`	Vector of positive integers. Numbers $1, 2, \ldots$ refer to contours of depth $1/n, 2/n, \ldots$, where n is the number of observations. Useful to draw particular contours. Default `dpth = NULL` corresponds to the set of all contours.
`output`	Logical. Default FALSE produces a contour plot; otherwise a list of contour vertices.
`twodim`	Logical. `twodim = FALSE` returns a transparent perspective plot making use of the `rgl` package.
`mustdith`	Logical. Should dithering be applied? Used when data set is not in general position or a numerical problem is encountered.
`maxdith`	Positive integer. Maximum number of dithering steps.
`dithfactor`	Scaling factor used for horizontal and vertical dithering.
`trace.errors`	Logical. Should all contours be considered? Used when a numerical problem is encountered for some inner contours. Default `trace.errors = FALSE` means those contours are left out.
`eps`	Error tolerance to control the calculation.
`factor`	Proportion (0 to 1) of outermost contours computed according to a version of the algorithm ISODEPTH of Rousseeuw and Ruts (1998); remaining contours are derived from an algorithm in Rousseeuw *et al.* (1999).
`xlab`	Title for x-axis. Must be a character string.
`ylab`	Title for y-axis. Must be a character string.
`zlab`	Title for z-axis. Used jointly with `twodim = FALSE`.
`colcontours`	Vector of color names of some or all of the contours. Recycling is used when necessary. Colors can be specified in different ways, see color specification in `par`,
`...`	Any additional graphical parameters (see `par`).

Details

Tukey's depth and dimension 2 only. Contours are computed according to algorithm ISODEPTH by Ruts and Rousseeuw (1996) or, more exactly, revised versions of this algorithm which appear in Rousseeuw and Ruts (1998) and Rousseeuw *et al.* (1999). Argument `factor` determines which version to use. If n is the number of observations, contours of depth $\leq$ `factor` $n/2$ are obtained from the 1998 version, while the remaining contours are derived from the 1999 version.

When the data set is not in general position, dithering can be used in the sense that random noise is added to each component of each observation. Random noise takes the form `eps` times `dithfactor` times U for the horizontal component and `eps` times `dithfactor` times V for the vertical component, where U, V are independent uniform on [-.5, 5.]. This is done in a number of consecutive steps applying independent U's and V's.

Value

Default `output = FALSE` yields a contour plot. If not, the function returns a list of m components, where m is the number of contours and component i is a matrix whose rows are the vertices of contour i.

Author(s)

Jean-Claude Masse and Jean-Francois Plante, based on Fortran code by Rousseeuw, Ruts and Struyf from University of Antwerp.

References

Ruts, I. and Rousseeuw, P.J. (1996), Computing depth contours of bivariate point clouds, *Comput. Stat. Data An.*, **23**, 153–168.

Rousseeuw, P.J. and Ruts, I. (1998), Constructing the bivariate Tukey median, *Stat. Sinica*, **8**, 828–839.

Rousseeuw, P.J., Ruts, I., and Tukey, J.W. (1999), The Bagplot: A Bivariate Boxplot, *The Am. Stat.*, **53**, 382–387.

See Also

`depth`, `perspdepth`

Examples

```
## exact contour plot with 10 contours
set.seed(601) ; x = matrix(rnorm(48), nc = 2)
isodepth(x)

## exact colored contours
set.seed(159); library(MASS)
mu1 <- c(0,0); mu2 <- c(6,0); sigma <- matrix(c(1,0,0,1), nc = 2)
mixbivnorm <- rbind(mvrnorm(80, mu1 ,sigma), mvrnorm(20, mu2, sigma))
isodepth(mixbivnorm, dpth = c(35,5), col = rainbow(2))

## vertices of each contour
set.seed(601)
x <- matrix(rnorm(48), nc = 2)
isodepth(x, output = TRUE)

## data set not in general position
data(starsCYG, package = "robustbase")
isodepth(starsCYG, mustdith = TRUE)
```

```
## colored contours
set.seed(601)
x <- matrix(rnorm(48), nc = 2)
isodepth(x, colcontours= rainbow(10))

# perspective plot
library(rgl)
set.seed(601)
x <- matrix(rnorm(48), nc = 2)
isodepth(x, twodim = FALSE)
```

med *Multivariate median*

Description

Computes the median of a multivariate data set.

Usage

```
med(x, method = "Tukey", approx = FALSE, eps = 1e-8, maxit = 200,
   mustdith = FALSE, maxdith = 50, dithfactor = 10, factor = 0.8,
   nstp = NULL, ntry = NULL, nalt = NULL,
   ndir = 1000)
```

Arguments

x	The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one multivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
method	Character string which determines the depth function used. `method` can be "Tukey" (the default), "Liu", "Oja", "Spatial" or "CWmed".
approx	Logical. Should an approximate Tukey median be computed? Useful in dimension 2 only when sample size is large.
eps	Error tolerance to control the calculation.
maxit	Number of Newton-Raphson iterations in case `method` is "Spatial".
mustdith	Logical.Should dithering be applied? Used to compute the Tukey median when data set is not in general position or a numerical problem is encountered.
maxdith	Integer. Maximum number of dithering steps.
dithfactor	Scaling factor used for horizontal and vertical dithering.
factor	Proportion (0 to 1) of outermost contours computed according to algorithm HALFMED of Rousseeuw and Ruts (1998); remaining contours derived from an algorithm in Rousseeuw *et al.* (1999).

nstp
: Positive integer. Maximum number of steps in the iteration process leading to an approximate value of the Tukey median. If NULL, the default value is taken to be the largest integer not greater than $5n^{0.3}p$, where n is the number of observations and p the dimension.

ntry
: Positive integer. Maximum number of steps without an increase of the Tukey depth in the iteration process leading to an approximate value of the Tukey median. If NULL, the default value is taken to be $10(p+1)$, where p is the dimension.

nalt
: Positive integer. Maximum number of consecutive steps without an increase of the Tukey depth at any time in the iteration process leading to an approximate value of the Tukey median. If NULL, the default value is taken to be $4(p+1)$, where p is the dimension.

ndir
: Positive integer. Number of random directions used in the iteration process leading to an approximate value of the Tukey median.

Details

method "Tukey" computes the Tukey median. Calculation is exact in dimensions 1 and 2, and approximate in higher dimensions. The bivariate case utilises algorithm HALFMED by Rousseeuw and Ruts (1998) as well as an algorithm from Rousseeuw *et al.* (1999). Argument factor determines which algorithm to use. If n is the number of observations, contours of depth $\leq$ factor $n/2$ are derived from algorithm HALFMED, while the remaining contours are obtained from the second algorithm. The higher dimensional case is covered by Fortran code from Struyf and Rousseeuw (2000).

When method is "Tukey", data must be in general position. If not, in dimension 2 dithering can be used in the sense that random noise is added to each component of each observation. Random noise takes the form eps times dithfactor times U for the horizontal component and eps times dithfactor times V for the vertical component, where U, V are independent uniform on [-.5, 5.]. This is done in a number of consecutive steps applying independent U's and V's.

method "Liu" computes the Liu median. It is based on Fortran code from Rousseeuw and Ruts (1996) and restricted to two-dimensional data.

method "Oja" computes the Oja median. It is based on Fortran code by Niinimaa et al. (1992) and restricted to two-dimensional data.

method "Spatial" computes the spatial median or mediancentre. It is based on Fortran code by Gower (1974), and Bedall and Zimmermann (1979).

method "CWmed" computes the coordinatewise median.

Value

A list with components

median
: the median

depth
: the depth of the median (omitted when method is "Spatial" or "CWmed")

Author(s)

Jean-Claude Masse and Jean-Francois Plante, based on Fortran code by authors listed in the references.

References

Gower, J.C. (1974), AS 78: The Mediancentre, *Appl. Stat.*, **23**, 466–470.

Bedall, F.K. and Zimmermann, H. (1979), AS 143: The Mediancentre, *Appl. Stat.*, **28**, 325–328.

Niinimaa, A, Oja, H., Nyblom, J. (1992), AS 277 : The Oja Bivariate Median, *Appl. Stat.*, **41**, 611–617.

Rousseeuw, P.J. and Ruts, I. (1996), Algorithm AS 307: Bivariate location depth, *Appl. Stat.-J. Roy. St. C*, **45**, 516–526.

Rousseeuw, P.J. and Ruts, I. (1998), Constructing the bivariate Tukey median, *Stat. Sinica*, **8**, 828–839.

Rousseeuw, P.J., Ruts, I., and Tukey, J.W. (1999), The Bagplot: A Bivariate Boxplot, *The Am. Stat.*, **53**, 382–387.

Small, C.G. (1990), A survey of multidimensional medians, *Int. Statist. Rev.*, **58**, 263–277.

Struyf, A. and Rousseeuw, P.J. (2000), High-dimensional computation of the deepest location, *Comput. Statist. Data Anal.*, **34**, 415–436.

Masse, J.C and Plante, J.F. (2003), A Monte Carlo study of the accuracy and robustness of ten bivariate location estimators, *Comput. Statist. Data Anal.*, **42**, 1–26.

See Also

`trmean` and `ctrmean` for trimmed means

Examples

```
## exact Tukey median for a mixture of bivariate normals
set.seed(159); library(MASS)
mu1 <- c(0,0); mu2 <- c(6,0); sigma <- matrix(c(1,0,0,1), nc = 2)
mixbivnorm <- rbind(mvrnorm(80, mu1, sigma), mvrnorm(20, mu2, sigma))
med(mixbivnorm)

##  approximate Tukey median of a four-dimensional data set
set.seed(601)
zz <- matrix(rnorm(96), nc = 4)
med(zz)

## data set not in general position
data(starsCYG, package = "robustbase")
med(starsCYG, method = "Liu")

## use of dithering for the Tukey median
med(starsCYG, mustdith = TRUE)
```

`perspdepth` *Perspective plots for depth functions*

Description

Draws a perspective plot of the surface of a depth function over the x-y plane.

Usage

```
perspdepth(x, method = "Tukey", output = FALSE, tt = 50,
   xlab = "X", ylab = "Y", zlab = NULL, col = NULL, ...)
```

Arguments

`x`	Bivariate data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, both components must be numerical vectors of equal length (coordinates of observations).
`method`	Character string which determines the depth function used. `method` can be "Tukey" (the default), "Liu" or "Oja".
`output`	Logical. Default FALSE produces a perspective plot; otherwise, returns a list containing the grid points and depth values over these points.
`tt`	Gridsize. Number of equally spaced grid points in each coordinate direction to be used in perspective plot.
`xlab`	Title for x-axis. Must be a character string.
`ylab`	Title for y-axis. Must be a character string.
`zlab`	Title for z-axis. Must be a character string. Default NULL identifies the depth function.
`col`	Color of the surface plot. Default NULL is "lightblue".
`...`	Any additional graphical parameters.

Details

Requires the `rgl` package. The perspective plot takes advantage of some of the user interaction facilities of that package.

Value

Default `output = FALSE` yields a perspective plot; otherwise the function returns a list with components

`x`	x-coordinates of the grid where the depth function is evaluated.
`y`	y-coordinates of the grid where the depth function is evaluated.
`z`	Matrix whose entry z[i ,j] is the value of the depth function at (x[i], y[j]).

Author(s)

Jean-Claude Masse and Jean-Francois Plante, based on Fortran code by Rousseeuw, Ruts and Struyf from University of Antwerp.

References

Rousseuw, P.J. and Ruts, I. (1996), AS 307 : Bivariate location depth, *Appl. Stat.-J. Roy. S. C*, **45**, 516–526.

See Also

isodepth, depth

Examples

```
## 2 perspective plots
data(geyser, package = "MASS")
perspdepth(geyser, col = "magenta")
set.seed(159); library(MASS)
mu1 <- c(0,0); mu2 <- c(6,0); sigma <- matrix(c(1,0,0,1), nc = 2)
mixbivnorm <- rbind(mvrnorm(80, mu1, sigma),mvrnorm(20, mu2, sigma))
perspdepth(mixbivnorm, col = "chartreuse")

## grid coordinates and corresponding depth values
set.seed(601)
x <- matrix(rnorm(48), nc = 2)
perspdepth(x, output = TRUE, tt = 10)
```

---

pkg-internal — *Internal functions of the depth package*

Description

These functions are not meant to be used at the user-level.

---

scontour — *Plotting spherical depth contours*

Description

Traces spherical depth contours of a multivariate data set. Supports data on the circle or on the sphere.

Usage

```
scontour(P, tracepoints=FALSE, colpoints="black", tracemed=TRUE,
   maxdepth=FALSE, xlim=c(0,2*pi), displaymed=FALSE,
   title="Circular Tukey contours", ylab="Tukey's circular depth",
   xlab=expression(theta), colmed=2, colarc="red", sizepoints=3)
```

Arguments

`P`	The data as a vector, a matrix, a data frame or a list.
`tracepoints`	Logical; if TRUE, data points are added to the plot.
`colpoints`	A specification for the color of the data points.
`tracemed`	Logical; if TRUE, the Tukey median is added to the plot.
`maxdepth`	Logical; On the circle only; if TRUE, the maximum depth is printed on the plot.
`xlim`	Numeric vectors of length 2, giving the x coordinate range.
`displaymed`	Logical; On the circle only; if TRUE, the median value is printed on the plot.
`title`	On the circle only, a specification for the plot title.
`ylab`	On the circle only, a specification for the y axis title.
`xlab`	On the circle only, a specification for the x axis title.
`colmed`	Color of the Tukey median on the plot.
`colarc`	On the sphere only, color of the spherical depth contours on the plot.
`sizepoints`	Size of plotted points.

Details

Supports data on the circle or the sphere. For data on the circle, data must be expressed in polar coordinates as a angle in radians with values between 0 and 2π. Data on the sphere can be expressed in Euclidean coordinates (n by 3 matrix) or in spherical coordinates (n by 2 matrix) where the first column contains θ and the second column ϕ. The type of coordinates is determined automatically based on the dimensions of the input.

Value

`plot`	A plot of Tukey spherical depth if the input data are on the circle, or the Tukey spherical depth contours if the input data are on the sphere.

If data are on the sphere only, a list of 3 elements is also outputted.

`1`	A sorted vector giving the depths of the plotted contours.
`2`	A list of matrices with the vertices of every contour.
`3`	The Euclidean coordinates of the Tukey median

Author(s)

Maxime Genest.

References

Liu, R.Y., Parelius, J.M. and Singh, K. (1999), Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion), *Ann. Statist.*, **27**, 783–858.

Mardia, K.V. and Jupp, E.J. (1999). *Directional Statistics*, Wiley.

See Also

sdepth for calculation of the depth of a point, smed for Tukey's spherical median.

Examples

```
## Plot of Tukey spherical depth for data on the circle.
set.seed(2011)
scontour(runif(30,min=0,max=2*pi))

## Tukey spherical depth contours for data
## on the shpere expressed in spherical coordinates.
scontour(cbind(runif(20,min=0,max=2*pi),runif(20,min=0,max=pi)))

## Tukey spherical depth contours for data
## on the sphere expressed in Euclidean coordinates.
x=matrix(rnorm(60),ncol=3)
x=t(apply(x,1,function(y){y/sqrt(sum(y^2))}))
scontour(x)
```

---

sdepth — *Calculation of spherical depth*

Description

Computes the spherical depth of a point with respect to a multivariate data set. Supports data on the circle or on the sphere.

Usage

```
sdepth(theta, P)
```

Arguments

theta	Numerical vector whose depth is to be calculated. The coordinate system must match that of the observations.
P	The data as a vector, a matrix, a data frame or a list.

Details

Computes the Tukey depth of `theta` with respect to the dataset `P`. For data on the circle, data must be expressed in polar coordinates as a angle in radians with values between 0 and 2π. Data on the sphere can be expressed in Euclidean coordinates (n by 3 matrix) or in spherical coordinates (n by 2 matrix) where the first column contains θ and the second column ϕ. The type of coordinates is determined automatically based on the dimensions of the input.

Value

Returns the spherical depth of multivariate point `theta` with respect to the data set `P`.

Author(s)

Maxime Genest.

References

Liu, R.Y., Parelius, J.M. and Singh, K. (1999), Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion), *Ann. Statist.*, **27**, 783–858.

Mardia, K.V. and Jupp, E.J. (1999). *Directional Statistics*, Wiley.

See Also

`scontour` for depth graphics, `smed` for Tukey's spherical median.

Examples

```
## Tukey spherical depth for a dataset on the circle
set.seed(2011)
sdepth(pi,runif(50,min=0,max=2*pi))

## Tukey spherical depth for data in spherical coordinates.
sdepth(c(pi,pi/2),cbind(runif(50,min=0,max=2*pi),runif(50,min=0,max=pi)))

## Tukey spherical depth for data in Eudlidean coordinates.
x=matrix(rnorm(150),ncol=3)
x=t(apply(x,1,function(y){y/sqrt(sum(y^2))}))
sdepth(x[1,],x)
```

---

smed — *Calculating spherical medians*

Description

Computes the spherical median of a data set on the circle.

Usage

```
smed(P, sort=FALSE, depths=NULL, alpha=NULL,
   method="Tukey", tracecontour=FALSE, tracepoints=FALSE)
```

Arguments

`P`	The data as a vector, a matrix, a data frame or a list.
`sort`	Logical; TRUE indicates that the data in `P` is already sorted.
`depths`	For Tukey's method only; An optionnal vector of the same length as P that contains the Tukey depth of each data. The calculation of the depth is then skipped and the provided values are used instead.
`alpha`	For Tukey's method only; `alpha` is an optionnal numeric value between 0 and 1 to compute the median on a trimmed region rather than on the whole dataset. The trimming keeps only those points with a depth greater than or equal to `alpha`. The default value of `NULL` computes the median from the maximum depth trimmed region (i.e. no trimming).
`method`	Character string which determines the depth function used. `method` can be "Tukey" (the default) or "Circular".
`tracecontour`	Only if `method="Circular"`. Traces the plot of depth with respect to angular positions on the circle.
`tracepoints`	Only if `method="Circular"`. Draws the points and their median on the circle.

Details

Calculates shperical medians for data on the circle only. The input must be a list of angles in radians between 0 and 2π (polar coordinates). If `method="Tukey"`, the Tukey median is returned. If `method="Circular"`, the circular median (the point minimizing the average distance based on arccosine) is returned.

Value

A numeric value between 0 and 2π giving the median in polar coordinate.

Author(s)

Maxime Genest.

References

Liu, R.Y., Parelius, J.M. and Singh, K. (1999), Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion), *Ann. Statist.*, **27**, 783–858.

Mardia, K.V. and Jupp, E.J. (1999). *Directional Statistics*, Wiley.

See Also

`sdepth` for calculation of the depth of a point, `scontour` for Tukey's spherical median.

Examples

```
## calculation of the Tukey spherical median for data on the circle
set.seed(2011)
smed(runif(30,min=0,max=2*pi))
```

---

strmeasure *Computing trimmed measures of sherical location*

Description

Computes a sample trimmed measure of location based on the spherical Tukey's depth. Supports data on the circle or on the sphere (for Circular median only).

Usage

```
strmeasure(P,sorted=FALSE,depths=NULL,alpha=0,method="Mean")
```

Arguments

`P`	The data as a vector, a matrix, a data frame or a list.
`sorted`	Logical; if TRUE, it indicates that the data given in first argument is sorted.
`depths`	An optionnal vector of the same length of P that contains the Tukey's depth of each data. The calculation of the depth is then skipped and the provided values are used instead.
`alpha`	An optionnal numeric value between 0 and 1 to compute the median on a trimmed region rather than on the whole dataset. The trimming keeps only those points with a depth greater than or equal to `alpha`. The default value of 0 computes the median from the maximum depth trimmed region (i.e. no trimming).
`method`	Character string which determines the measure used. `method` can be "Mean" (the default) to compute trimmed mean direction or "Tukey" (for circular sample only) to compute trimmed Tukey's median.

Details

This function returns a location estimate (Tukey's median or mean direction) of a sample truncated by Tukey's depth. For data on the circle, data must be expressed in polar coordinates as a angle in radians with values between 0 and 2π. Data on the sphere can be expressed in Euclidean coordinates (n by 3 matrix) or in spherical coordinates (n by 2 matrix) where the first column contains θ and the second column ϕ. The type of coordinates is determined automatically based on the dimensions of the input.

While the option `method="Tukey"` supports only data on the circle, `method="Mean"` can also handle data on the sphere.

Value

If the input sample is on the circle, a numeric value between 0 and 2π giving the trimmed measure. If the input sample is on the sphere, the trimmed measure in Euclidean coordinates.

Author(s)

Maxime Genest.

References

Liu, R.Y., Parelius, J.M. and Singh, K. (1999), Multivariate analysis by data depth: Descriptive statistics, graphics and inference (with discussion), *Ann. Statist.*, **27**, 783–858.

Mardia, K.V. and Jupp, E.J. (1999). *Directional Statistics*, Wiley.

See Also

sdepth for the calculation of the depth of a point, scontour for Tukey's spherical median.

Examples

```
## calculation of trimmed mean direction
set.seed(2011)
strmeasure(runif(30,min=0,max=2*pi),alpha=1/3,method="Mean")

## calculating of trimmed Tukey median
set.seed(2011)
strmeasure(runif(30,min=0,max=2*pi),alpha=1/3,method="Tukey")
```

---

trmean *Classical-like depth-based trimmed mean*

Description

Computes a sample trimmed mean based on the Tukey depth, the Liu depth or the Oja depth.

Usage

```
trmean(x, alpha, W = function(dep, alpha){return(1)},
   method = "Tukey", ndir = 1000, approx = FALSE,
   eps = 1e-8, ...)
```

Arguments

x	The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
alpha	Outer trimming fraction (0 to 0.5). Observations whose depth is less than alpha to be trimmed.
W	Nonnegative weight function defined on [0, 1] through its argument dep. Number of arguments can be greater than 2 but the trimming fraction has to be one argument. See examples.

method	Character string which determines the depth function used. method can be "Tukey" (the default), "Liu" or "Oja".
ndir	Positive integer. Number of random directions used when approximate Tukey depth is utilised. Used jointly with approx = TRUE.
approx	Logical. If dimension is 3, should approximate Tukey depth be used? Useful when sample size is large.
eps	Error tolerance to control the calculation.
...	Any additional arguments to the weight function.

Details

Dimension 2 or higher when method is "Tukey" or "Oja"; dimension 2 only when method is "Liu". Exactness of calculation depends on method. See depth.

Value

Multivariate depth-based trimmed mean

Author(s)

Jean-Claude Masse and Jean-Francois Plante, based on Fortran code by Ruts and Rousseeuw from University of Antwerp.

References

Masse, J.C and Plante, J.F. (2003), A Monte Carlo study of the accuracy and robustness of ten bivariate location estimators, *Comput. Statist. Data Anal.*, **42**, 1–26.

Masse, J.C. (2008), Multivariate Trimmed means based on the Tukey depth, *J. Statist. Plann. Inference*, in press.

Rousseeuw, P.J. and Ruts, I. (1996), Algorithm AS 307: Bivariate location depth, *Appl. Stat.-J. Roy. St. C*, **45**, 516–526.

See Also

med for medians and ctrmean for a centroid trimmed mean.

Examples

```
## exact trimmed mean with default constant weight function
data(starsCYG, package = "robustbase")
trmean(starsCYG, .1)

## another example with default constant weight function
set.seed(159); library(MASS)
mu1 <- c(0,0); mu2 <- c(6,0); sigma <- matrix(c(1,0,0,1), nc = 2)
mixbivnorm <- rbind(mvrnorm(80, mu1, sigma), mvrnorm(20, mu2, sigma))
trmean(mixbivnorm, 0.3)

## trimmed mean with a non constant weight function
```

```
W1 <-function(x,alpha,epsilon) {
  (2*(x-alpha)^2/epsilon^2)*(alpha<=x)*(x<alpha+epsilon/2)+
  (-2*(x-alpha)^2/epsilon^2+4*(x-alpha)/epsilon-1)*
  (alpha+epsilon/2<=x)*(x<alpha+epsilon)+(alpha+epsilon<=x)
}
set.seed(345)
x <- matrix(rnorm(210), nc = 3)
trmean(x, .1, W = W1, epsilon = .05)

## two other examples of weighted trimmed mean
set.seed(345)
x <- matrix(rnorm(210), nc = 3)
W2 <- function(x, alpha) {x^(.25)}
trmean(x, .1, W = W2)
W3 <- function(x, alpha, beta){1-sqrt(x)+x^2/beta}
trmean(x, .1, W = W3, beta = 1)
```

Index