Package ‘depth’

January 2, 2012

Type Package

Title Depth functions tools for multivariate analysis

Version 1.0-1

Date 2009-02-09

Author Jean-Claude Masse <jcmasse@mat.ulaval.ca>, Jean-Francois Plante <plante@utstat.toronto.edu>.

Maintainer Jean-Francois Plante <plante@utstat.toronto.edu>

Description Depth functions methodology applied 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.

Depends R (>= 2.4.0), grDevices, rgl

Suggests robustbase, MASS

License GPL-2

Repository CRAN

Date/Publication 2009-02-10 08:14:26

R topics documented:

depth-package 2
ctrmean 3
depth 4
isodepth 6
med 8
perspdepth 11
trmean 13

Index **16**

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:	1.0
Date:	2008-09-01
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.

Author(s)

Jean-Claude Masse <jcmasse@mat.ulaval.ca>, Jean-Francois Plante <plante@utstat.toronto.edu>.

Maintainer: Jean-Francois Plante <plante@utstat.toronto.edu>

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.

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 $0.5(1 + \binom{n}{p}^{-1} \sum (\mathrm{Volume}(S(u, x[i_1,], \ldots, x[i_p,])))^{-1}$, where $S(args)$ denotes the simplex generated by $args$, and sum and average are taken over all p-plets $x[i_1,], \ldots, x[i_p,]$ such that $1 <= i_1 < \ldots . < i_p <= n$. 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.

Rousseuw, 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(mean(starsCYG), 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 = floor(5*n^0.3*p), ntry = 10*(p+1), nalt = 4*(p+1),
   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. 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. 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. 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")
open3d()
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)
```

`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)

## example with a large data set
set.seed(345)
x <- matrix(rnorm(2100), nc = 3)
trmean(x, .1, approx = TRUE)

## 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

∗Topic **multivariate**
ctrmean, 3
depth, 4
depth-package, 2
isodepth, 6
med, 8
perspdepth, 11
trmean, 13
∗Topic **nonparametric**
ctrmean, 3
depth, 4
depth-package, 2
isodepth, 6
med, 8
perspdepth, 11
trmean, 13
∗Topic **package**
depth-package, 2
∗Topic **robust**
ctrmean, 3
depth, 4
depth-package, 2
isodepth, 6
med, 8
perspdepth, 11
trmean, 13

ctrmean, *2*, 3, *11*, *14*

depth, *2*, 4, *8*, *12, 13*
depth-package, 2

isodepth, *2*, 6, *6*, *12*

med, *2*, *4*, 8, *14*

par, *7*
perspdepth, *2*, *6*, *8*, 11

trmean, *2*, *4*, *11*, 13