Package ‘DepthProc’

February 19, 2020

Version 2.1.3

Title Statistical Depth Functions for Multivariate Analysis

Description Data depth concept offers a variety of powerful and user friendly tools for robust exploration and inference for multivariate data. The offered techniques may be successfully used in cases of lack of our knowledge on parametric models generating data due to their nature. The package consist of among others implementations of several data depth techniques involving multivariate quantile-quantile plots, multivariate scatter estimators, multivariate Wilcoxon tests and robust regressions.

License GPL-2

Depends R (>= 3.0.0), ggplot2, Rcpp (>= 0.11.2), rrcov, methods, MASS, np

Imports lattice, sm, geometry, colorspace, zoo, grDevices

Suggests mvtnorm, rgl, sn, robustbase, dplyr, RcppArmadillo, xts, covr, testthat, fda, lintr

LinkingTo Rcpp, RcppArmadillo

SystemRequirements C++11

RoxygenNote 7.0.2

Encoding UTF-8

URL `https://www.depthproc.zstat.pl/`, `https://github.com/zzawadz/DepthProc`

BugReports `https://github.com/zzawadz/DepthProc/issues`

NeedsCompilation yes

Author Zygmunt Zawadzki [aut, cre], Daniel Kosiorowski [aut], Krzysztof Slomczynski [ctb], Mateusz Bocian [ctb], Anna Wegrzynkiewicz [ctb]

Maintainer Zygmunt Zawadzki `<zygmunt@zstat.pl>`

Repository CRAN

Date/Publication 2020-02-19 21:00:03 UTC

R topics documented:

`abline,RobReg-method` *Add line to plot*

Description

Add fitted line to a plot. This is overloaded function for robust regression methods from package depthproc.

Usage

```
## S4 method for signature 'RobReg'
abline(
  a = NULL,
  b = NULL,
  h = NULL,
  v = NULL,
  reg = NULL,
  coef = NULL,
  untf = FALSE,
  ...
)
```

Arguments

a	an object of class RobReg
b	not used.
h	not supported.
v	not supported.

reg	not supported.
coef	not supported.
untf	not supported.
...	Arguments to be passed to methods, such as graphical parameters (see par).

---

as.matrix *as.matrix method for DepthCurveList.*

Description

Create a matrix from DepthCurve and DepthCurveList.

Usage

```
as.matrix(x, ...)

## S4 method for signature 'DepthCurveList'
as.matrix(x)
```

Arguments

x	an object of class that inherits from DepthCurveList (ScaleCurveList or AsymmetryCurveList).
...	other arguments passed to standard as.matrix function.

---

asymmetryCurve *Asymmetry curve based on depths*

Description

Produces an asymmetry curve estimated from given data.

Usage

```
asymmetryCurve(
  x,
  y = NULL,
  alpha = seq(0, 1, 0.01),
  movingmedian = FALSE,
  name = "X",
  name_y = "Y",
  depth_params = list(method = "Projection")
)
```

Arguments

`x`	The data as a matrix or data frame. If it is a matrix or data frame, then each row is viewed as one multivariate observation.
`y`	Additional matrix of multivariate data.
`alpha`	An ordered vector containing indices of central regins used for asymmetry curve calculation.
`movingmedian`	Logical. For default FALSE only one depth median is used to compute asymmetry norm. If TRUE — for every central area, a new depth median will be used — this approach needs much more time.
`name`	Name of set X — used in plot legend
`name_y`	Name of set Y — used in plot legend
`depth_params`	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).
`method`	Character string which determines the depth function used. The method can be "Projection" (the default), "Mahalanobis", "Euclidean", "Tukey" or "LP". For details see `depth`.

Details

For sample depth function $D(x, Z^n)$, $x \in R^d$, $d \geq 2$, $Z^n = \{z_1, ..., z_n\} \subset R^d$, $D_\alpha(Z^n)$ denoting α — central region, we can define the asymmetry curve $AC(\alpha) = \left(\alpha, \left\|c^{-1}(\{\bar{z} - med|D_\alpha(Z^n)\})\right\|\right) \subset R^2$, for $\alpha \in [0, 1]$ being nonparametric scale and asymmetry functional correspondingly, where c — denotes constant, $\bar{z}$ — denotes mean vector, denotes multivariate median induced by depth function and vol — denotes a volume.

Asymmetry curve takes uses function convhulln from package geometry for computing a volume of convex hull containing central region.

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

References

Serfling R. J. Multivariate Symmetry and Asymmetry, *Encyclopedia of Statistical Science*, S Kotz, C.B. Read, N. Balakrishnan, B. Vidakovic (eds), 2nd, ed., John Wiley.

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.

Chaudhuri, P. (1996), On a Geometric Notion of Quantiles for Multivariate Data, *Journal of the American Statistical Association*, 862–872.

Dyckerhoff, R. (2004), Data Depths Satisfying the Projection Property, *Allgemeines Statistisches Archiv.*, **88**, 163–190.

See Also

`scaleCurve`, `depth`

Examples

```
# EXAMPLE 1
library(sn)
xi <- c(0, 0)
alpha <- c(2, -5)
Omega <- diag(2) * 5

n <- 500
X <- mvrnorm(n, xi, Omega) # normal distribution
Y <- rmst(n, xi, Omega, alpha, nu = 1)
asymmetryCurve(X, Y, name = "NORM", name_y = "S_T(2, -5, 10)")

# EXAMPLE 2
data(under5.mort)
data(inf.mort)
data(maesles.imm)
data1990 <- cbind(under5.mort[, 1], inf.mort[, 1], maesles.imm[, 1])
data2011 <- cbind(under5.mort[, 22], inf.mort[, 22], maesles.imm[, 22])
as1990 <- asymmetryCurve(data1990, name = "scale curve 1990")
as2011 <- asymmetryCurve(data2011, name = "scale curve 2011")
figure <- getPlot(combineDepthCurves(as1990, as2011)) +
  ggtitle("Scale curves")
figure
```

AsymmetryCurve-class   *AsymmetryCurve and AsymmetryCurveList*

Description

AsymmetryCurve is a class that stores results of asymmetryCurve function.

Details

The mechanism of creating plots with multiple curves is shown in DepthCurve-class (same mechanism is applied for ScaleCurve).

BinnDepth2d-class   *BinnDepth2d*

Description

Class that stores result of function binningDepth2D(...)

Slots

- `freq` Matrix with number of elements in certain bin.
- `mid_x` Middle values on x-axis.
- `mid_y` Middle values on y-axis.
- `breaks_x` Boundaries of bins.
- `breaks_y` Boundaries of bins.
- `input_data` Binned data.
- `max_depth_x` Point with maximum depth on x-axis.
- `max_depth_y` Point with maximum depth on y-axis.

`binningDepth2D` *2d Binning*

Description

A robust method of decreasing a sample size and therefore a complexity of a statistical procedure. The method may be used within a kernel density or a predictive distribution estimation.

Usage

```
binningDepth2D(
  x,
  binmethod = "LocDepth",
  nbins = 8,
  k = 1,
  remove_borders = FALSE,
  depth_params = list(method = "LP")
)
```

Arguments

`x`	bivariate matrix containing data. Each row is viewed as one two-dimensional observation.
`binmethod`	A method for calculation center and dispersion measures. "LocDepth" uses location-scale depth, MAD uses median and MAD in each dimension.
`nbins`	number of bins in each dimension
`k`	responsible for tightness of bins.
`remove_borders`	Logical, include or not marginal bins
`depth_params`	other arguments passed to depthMedian

Details

Let us recall, that binning is a popular method of decreasing a sample size. To bin a window of n points $W_{i,n} = \{X_{i-n+1}, ..., X_i\}$ to a grid $X'_1, ..., X'_m$ we simply assign each sample point X_i to the nearest grid point X'_j. When binning is completed, each grid point X'_j has an associated number c_i, which is the sum of all the points that have been assigned to X'_j. This procedure replaces the data $W_{i,n} = \{X_{i-n+1}, ..., X_i\}$ with the smaller set $W'_{j,m} = \{X'_{j-m+1}, ..., X'_j\}$. Although simple binning can speed up the computation, it is criticized for a lack of precise approximate control over the accuracy of the approximation. Robust binning however stresses properties of the majority of the data and decreases the computational complexity of the DSA at the same time.

For a 1D window $W_{i,n}$, let $Z_{i,n-k}$ denote a 2D window created basing on $W_{i,n}$ and consisted of $n-k$ pairs of observations and the k lagged observations $Z_{i,n-k} = \{(X_{i-n-k}, X_{i-n+1})\}, 1 \leq i \leq n-k$. Robust 2D binning of the $Z_{i,n-p}$ is a very useful technique in a context of robust estimation of the predictive distribution of a time series (see *Kosiorowski:2013b*).

Assume we analyze a data stream $\{X_t\}$ using a moving window of a fixed length n, i.e., $W_{i,n}$ and the derivative window $Z_{i,n-1}$. In a first step we calculate the weighted sample L^p depth for $W_{i,n}$. Next we choose equally spaced grid of points $l_1, ..., l_m$ in this way that $[l_1, l_m] \times [l_1, l_m]$ covers fraction of the β central points of $Z_{i,n-1}$ w.r.t. the calculated L^p depth, i.e., it covers $R^{\beta}(Z_{i,n-1})$ for certain prefixed threshold $\beta \in (0, 1)$. For both X_t and X_{t-1} we perform a simple binning using following bins: $(-\infty, l_1)$, (l_1, l_2), ..., (l_m, ∞). For robust binning we reject "border" classes and further use only midpoints and binned frequencies for classes (l_1, l_2), (l_2, l_3), ..., (l_{m-1}, l_m).

Value

freq: a matrix containing the binned frequencies

mid_x: mid points for x

mid_y: mid points for y

breaks_x: breaks for x

breaks_y: breaks for y

input_data: max_depth_x and max_depth_y:

Author(s)

Daniel Kosiorowski and Zygmunt Zawadzki from Cracow University of Economics.

References

Hall, P., Wand, M. P. (1996) On the Accuracy of Binned Kernel Density Estimators, Journal of Multivariate Analysis archive, Volume 56 Issue 2, 165–184

Holmstrom, L. (2000) The Accuracy and the Computational Complexity of a Multivariate Binned Kernel Density Estimator, Journal of Multivariate Analysis, Volume 72, Issue 2, 264–309, http://dx.doi.org/10.1006/jmva.1999.1863. (http://www.sciencedirect.com/science/article/pii/S0047259X99918638)

See Also

depth

Examples

```
# EXAMPLE 1
Sigma1 <- matrix(c(10, 3, 3, 2), 2, 2)
X1 <- mvrnorm(n = 8500, mu = c(0, 0), Sigma1)
Sigma2 <- matrix(c(10, 0, 0, 2), 2, 2)
X2 <- mvrnorm(n = 1500, mu = c(-10, 6), Sigma2)
BALLOT <- rbind(X1, X2)
train <- sample(1:10000, 500)
data <- BALLOT[train, ]
plot(data)

b1 <- binningDepth2D(data, remove_borders = FALSE, nbins = 12, k = 1)
b2 <- binningDepth2D(data, nbins = 12, k = 1, remove_borders = TRUE)
plot(b1)
plot(b2)

# EXAMPLE 2
data(under5.mort)
data(maesles.imm)
data2011 <- cbind(under5.mort[, 22], maesles.imm[, 22])
plot(binningDepth2D(data2011, nbins = 8, k = 0.5, remove_borders = TRUE))
```

---

combineDepthCurves *Adds plots*

Description

Adds plots

Usage

```
combineDepthCurves(x, y, .list = NULL)

## S4 method for signature 'ANY,ANY,list'
combineDepthCurves(x, y, .list = NULL)

## S4 method for signature 'DepthCurveList,DepthCurve,ANY'
combineDepthCurves(x, y, .list = NULL)

## S4 method for signature 'DepthCurve,DepthCurveList,ANY'
combineDepthCurves(x, y, .list = NULL)

## S4 method for signature 'DepthCurve,DepthCurve,ANY'
combineDepthCurves(x, y, .list = NULL)
```

Arguments

`x`	object
`y`	object
`.list`	list of plots to combine.

Details

See `DepthCurve-class` for description.

`CovDepthWeighted-class` *CovLP*

Description

This class, derived from the virtual class "CovRobust" accomodates weighted by L^p depth multivariate location and scatter estimator.

Details

See `CovLP` for the function used to calculate weighted by L^p depth covariance matrix.

`CovLP` *CovLp*

Description

Weighted by L^p depth (outlyingness) multivariate location and scatter estimators.

Usage

```
CovLP(x, pdim = 2, la = 1, lb = 1)
```

Arguments

`x`	The data as a matrix or data frame. If it is a matrix or data frame, then each row is viewed as one multivariate observation.
`pdim`	The parameter of the weighted $L^p dim$ depth
`la`	parameter of a simple weight function $w = ax + b$
`lb`	parameter of a simple weight function $w = ax + b$

Details

Using depth function one can define a depth-weighted location and scatter estimators. In case of location estimator we have

$$L(F) = \int x w_1(D(x, F)) dF(x) / w_1(D(x, F)) dF(x)$$

Subsequently, a depth-weighted scatter estimator is defined as

$$S(F) = \frac{\int (x - L(F))(x - L(F))^T w_2(D(x, F)) dF(x)}{\int w_2(D(x, F)) dF(x)},$$

where $w_2(\cdot)$ is a suitable weight function that can be different from $w_1(\cdot)$.

The **DepthProc** package offers these estimators for weighted L^p depth. Note that $L(\cdot)$ and $S(\cdot)$ include multivariate versions of trimmed means and covariance matrices. Their sample counterparts take the form

$$T_{WD}(X^n) = \sum_{i=1}^{n} d_i X_i / \sum_{i=1}^{n} d_i,$$

$$DIS(X^n) = \frac{\sum\limits_{i=1}^{n} d_i \left(X_i - T_{WD}(X^n)\right)\left(X_i - T_{WD}(X^n)\right)^T}{\sum\limits_{i=1}^{n} d_i},$$

where d_i are sample depth weights, $w_1(x) = w_2(x) = x$.

Value

loc: Robust Estimate of Location:

cov: Robust Estimate of Covariance:

Returns depth weighted covariance matrix.

Author(s)

Daniel Kosiorowski and Zygmunt Zawadzki from Cracow University of Economics.

See Also

`depthContour` and `depthPersp` for depth graphics.

Examples

```
# EXAMPLE 1
x <- mvrnorm(n = 100, mu = c(0, 0), Sigma = 3 * diag(2))
cov_x <- CovLP(x, 2, 1, 1)

# EXAMPLE 2
data(under5.mort, inf.mort, maesles.imm)
data1990 <- na.omit(cbind(under5.mort[, 1], inf.mort[, 1], maesles.imm[, 1]))
CovLP(data1990)
```

cracow.airpollution *Air pollution with PM10 in Cracow within day and night in December 2016*

Description

Air pollution with PM10 in Cracow within day and night in December 2016

Usage

```
data("cracow.airpollution")
```

Format

data frame containing 744 rows.

References

1. Kosiorowski D, Rydlewski J P, Zawadzki Z (2017). 'Functional Outliers Detection By The Example Of Air Quality Monitoring.' submitted.

2. Kosiorowski D, Szlachtowska E (2017). 'K- local Median Algorithm for Functional Data in Empirical Analysis of Air Pollution Data.' Proceedings from the 11th Professor A. Zelias International Conference, pp. 153-162. Cracow University of Economics.

ddmvnorm *Normal depth versus depth plot*

Description

Produces a normal DD plot of a multivariate dataset.

Usage

```
ddMvnorm(
  x,
  size = nrow(x),
  robust = FALSE,
  alpha = 0.05,
  title = "ddMvnorm",
  depth_params = list()
)
```

Arguments

`x`	The data sample for DD plot.
`size`	size of theoretical set
`robust`	Logical. Default `FALSE`. If `TRUE`, robust measures are used to specify the parameters of theoretical distribution.
`alpha`	cutoff point for robust measure of covariance.
`title`	title of a plot.
`depth_params`	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).

Details

In the first step the location and scale of x are estimated and theoretical sample from normal distribution with those parameters is generated. The plot presents the depth of empirical points with respect to dataset x and with respect to the theoretical sample.

Value

Returns the normal depth versus depth plot of multivariate dataset `x`.

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

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., Singh K. (1993), A Quality Index Based on Data Depth and Multivariate Rank Test, *Journal of the American Statistical Association* vol. 88.

See Also

`ddPlot` to generate ddPlot to compare to datasets or to compare a dataset with other distributions.

Examples

```
# EXAMPLE 1
norm <- mvrnorm(1000, c(0, 0, 0), diag(3))
con <- mvrnorm(100, c(1, 2, 5), 3 * diag(3))
sample <- rbind(norm, con)
ddMvnorm(sample, robust = TRUE)

# EXAMPLE 2
data(under5.mort, inf.mort, maesles.imm)
data1990 <- na.omit(cbind(under5.mort[, 1], inf.mort[, 1], maesles.imm[, 1]))
ddMvnorm(data1990, robust = FALSE)
```

ddPlot *Depth versus depth plot*

Description

Produces a DD plot which allows to compare two multivariate datasets or to compare a subject dataset with theoretical distribution.

Usage

```
ddPlot(
  x,
  y,
  scale = FALSE,
  location = FALSE,
  name = "X",
  name_y = "Y",
  title = "Depth vs. depth plot",
  depth_params = list()
)
```

Arguments

`x`	The first or only data sample for ddPlot.
`y`	The second data sample. `x` and `y` must be of the same space.
`scale`	logical. determines whether the dispersion is to be aligned.
`location`	determines whether the location is to be aligned to 0 vector with depth median.
`name`	name for data set x. It will be passed to drawing function.
`name_y`	as above for y
`title`	title of the plot.
`depth_params`	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).

Details

For two probability distributions F and G, both in R^d, we can define `depth vs. depth` plot being very useful generalization of the one dimensional quantile-quantile plot:

$$DD(F,G) = \left\{ \left(D(z,F), D(z,G) \right), z \in R^d \right\}$$

Its sample counterpart calculated for two samples $X^n = \{X_1, ..., X_n\}$ from F, and $Y^m = \{Y_1, ..., Y_m\}$ from G is defined as

$$DD(F_n, G_m) = \left\{ \left(D(z, F_n), D(z, G_m) \right), z \in \{X^n \cup Y^m\} \right\}$$

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

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**, 822–831.

Liu, R.Y., Singh K. (1993), A Quality Index Based on Data Depth and Multivariate Rank Test, *Journal of the American Statistical Association* vol. 88.

Examples

```
library(sn)
library(mvtnorm)

# EXAMPLE 1: Location difference
standard <- mvrnorm(1000, c(0, 0), diag(2))
shift <- mvrnorm(1000, c(0.5, 0), diag(2))
ddPlot(x = standard, y = shift, title = "Difference in position")
ddPlot(x = standard, y = shift, location = TRUE, title = "Location aligned")

# EXAMPLE 2: Scale difference
standard <- mvrnorm(1000, c(0, 0), diag(2))
scale <- mvrnorm(1000, c(0, 0), 4 * diag(2))
ddPlot(x = standard, y = scale)
ddPlot(x = standard, y = scale, scale = TRUE)
```

DDPlot-class *DDPlot*

Description

Class fro DDPlot

Slots

`X` Object of class Depth-class.

`Y` Object of class Depth-class.

`title` title of a plot.

`deepReg2d` *Simple deepest regression method.*

Description

This function calculates deepest regression estimator for simple regression.

Usage

```
deepReg2d(x, y)
```

Arguments

x	Independent variable.
y	Dependent variable.

Details

Function originates from an original algorithm proposed by Rousseeuw and Hubert. Let $Z^n = (x_1, y_1), ..., (x_n, y_n) \subset R^d$ denotes a sample considered from a following semiparametric model: $y_l = a_0 + a_1 x_{1l} + ... + a_{(d-1)l} x_{(d-1)l} + \varepsilon_l, l = 1, ..., n$, we calculate a depth of a fit $\alpha = (a_0, ..., a_{d-1})$ as $RD(\alpha, Z^n) = u \neq 0 \min \sharp l : \frac{r_l(\alpha)}{u^T x_l} < 0, l = 1, ..., n$, where $r(\cdot)$ denotes the regression residual, $\alpha = (a_0, ..., a_{d-1}), u^T x_l \neq 0$. The `deepest regression estimator` $DR(\alpha, Z^n)$ is defined as $DR(\alpha, Z^n) = \alpha \neq 0 \text{ arg max } RD(\alpha, Z^n)$

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

References

Rousseeuw J.P., Hubert M. (1998), Regression Depth, *Journal of The American Statistical Association*, vol.94.

Examples

```
# EXAMPLE 1
data(pension)
plot(pension)
abline(
  lm(Reserves ~ Income, data = pension),
  lty = 3,
  lwd = 2) # lm
abline(
  deepReg2d(pension[, 1], pension[, 2]),
  lwd = 2) # deepreg2d
```

```
# EXAMPLE 2
data(under5.mort)
data(inf.mort)
data(maesles.imm)
data2011 <- na.omit(
    cbind(under5.mort[, 22], inf.mort[, 22],
    maesles.imm[, 22]))

x <- data2011[, 3]
y <- data2011[, 2]
plot(
  x, y,
  cex = 1.2,
  ylab = "infant mortality rate per 1000 live birth",
  xlab = "against masles immunized percentage",
  main = "Projection Depth Trimmed vs. LS regressions"
)
abline(lm(x ~ y), lwd = 2, col = "black") # lm
abline(
  deepReg2d (x, y),
  lwd = 2, col = "red"
) # trimmed reg

legend(
  "bottomleft",
  c("LS", "DeepReg"),
  fill = c("black", "red"),
  cex = 1.4,
  bty = "n"
)
```

`DeepReg2d-class` *DeepReg2d*

Description

Class for robust regression methods from depthproc package

Slots

`coef` coefficients of fitted model

`depth` regression depth of the fitted values depth *Depth calculation*

Description

Calculate depth functions.

Usage

```
depth(u, X, method = "Projection", threads = -1, ...)
```

Arguments

u	Numerical vector or matrix 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. `method` can be "Projection" (the default), "Mahalanobis", "Euclidean" or "Tukey". For details see `depth`.
threads	number of threads used in parallel computations. Default value -1 means that all possible cores will be used.
...	parameters specific to method — see `depthEuclid`

Details

The Mahalanobis depth

$$D_{MAH}(y, X^n) = \frac{1}{1 + (y - \bar{x})^T S^{-1} (y - \bar{x})},$$

where S denotes the sample covariance matrix X^n.

A symmetric projection depth $D\left(x, X\right)$ of a point $x \in R^d$, $d \geq 1$ is defined as

$$D\left(x, X\right)_{PRO} = \left[1 + sup_{\|u\|=1} \frac{\left|u^T x - Med\left(u^T X\right)\right|}{MAD\left(u^T X\right)}\right]^{-1},$$

where Med denotes the univariate median, $MAD\left(Z\right)$ = $Med\left(\left|Z - Med\left(Z\right)\right|\right)$. Its sample version denoted by $D\left(x, X^n\right)$ or $D\left(x, X^n\right)$ is obtained by replacing F by its empirical counterpart F_n calculated from the sample X^n .

Next interesting depth is the weighted L^p depth. The weighted L^p depth $D(x, F)$ of a point $x \in R^d$, $d \geq 1$ generated by d dimensional random vector X with distribution F, is defined as $D(x, F) = \frac{1}{1+Ew(\|x-X\|_p)}$, where w is a suitable weight function on $[0, \infty)$, and $\|\cdot\|_p$ stands for the L^p norm (when p = 2 we have usual Euclidean norm). We assume that w is non-decreasing and continuous on $[0, \infty)$ with $w(\infty-) = \infty$, and for $a, b \in R^d$ satisfying $w(\|a + b\|) \leq w(\|a\|) + w(\|b\|)$. Examples of the weight functions are: $w(x) = a + bx$, $a, b > 0$ or $w(x) = x^{\alpha}$. The empirical version of the weighted L^p depth is obtained by replacing distribution F of X in $Ew(\|x - X\|_p) = \int w(\|x - t\|_p)dF(t)$ by its empirical counterpart calculated from the sample X^n...

The Projection and Tukey's depths are calculated using an approximate algorithm. Calculations of Mahalanobis, Euclidean and L^p depths are exact. Returns the depth of multivariate point u with respect to data set X.

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

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.

Mosler K (2013). Depth statistics. In C Becker, R Fried, K S (eds.), Robustness and Complex Data Structures, Festschrift in Honour of Ursula Gather, pp. 17–34. Springer.

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

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

See Also

depthContour and depthPersp for depth graphics.

Examples

```
library(robustbase)

# Calculation of Projection depth
data(starsCYG, package = "robustbase")
depth(t(colMeans(starsCYG)), starsCYG)

# Also for matrices
depth(starsCYG, starsCYG)

# Projection depth applied to a large bivariate data set
x <- matrix(rnorm(9999), nc = 3)
depth(x, x)
```

`Depth-class` *Depth*

Description

Virtual class with structure for every depth class from depthproc package.

Slots

`u` data set.

`X` reference set.

`method` depth type.

`depthContour` *Approximate depth contours*

Description

Draws an approximate contours of depth for bivariate data.

Usage

```
depthContour(
  x,
  xlim = extendrange(x[, 1], f = 0.1),
  ylim = extendrange(x[, 2], f = 0.1),
  n = 50,
  pmean = TRUE,
  mcol = "blue",
  pdmedian = TRUE,
  mecol = "brown",
  legend = TRUE,
  points = FALSE,
  colors = heat_hcl,
  levels = 10,
  depth_params = list(),
  graph_params = list(),
  contour_method = c("auto", "convexhull", "contour")
)
```

Arguments

`x`	Bivariate data
`xlim`	Determines the width of x-axis.
`ylim`	Determines the width of y-axis.
`n`	Number of points in each coordinate direction to be used in contour plot.
`pmean`	Logical. If TRUE mean will be marked.
`mcol`	Determines the color of lines describing the mean.
`pdmedian`	Logical. If TRUE depth median will be marked.
`mecol`	Determines the color of lines describing the depth median.
`legend`	Logical. If TRUE legend for mean and depth median will be drawn.
`points`	Logical. If TRUE points from matrix x will be drawn.
`colors`	function for colors pallete (e.g. gray.colors).
`levels`	number of levels for color scale.
`depth_params`	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).
`graph_params`	list of graphical parameters for functions filled.contour and contour (e.g. lwd, lty, main).
`contour_method`	determines the method used to draw the contour lines. The default value ("auto") tries to determine the best method for given depth function. "convexhull" uses a convex hull algorithm to determine boundaries. "contour" uses the algorithm from `filled.contour`.

Details

The set of all points that have depth at least α is called α-trimmed region. The α-trimmed region w.r.t. F is denoted by $D_\alpha(F)$, i.e.,

$$D_\alpha(F) = \left\{ z \in R^d : D(z, F) \geq \alpha \right\}.$$

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

See Also

`depthPersp`

Examples

```
# EXAMPLE 1
set.seed(123)
x <- mvrnorm(1000, c(0, 0), diag(2))
depthContour(x, colors = gray.colors)
# with points
```

```
depthContour(x, points = TRUE)
depthContour(x, points = FALSE, levels = 10)

# EXAMPLE 2
data(inf.mort, maesles.imm)
data1990 <- na.omit(cbind(inf.mort[, 1], maesles.imm[, 1]))
depthContour(data1990, n = 50, pmean = TRUE, mcol = "blue",
             pdmedian = TRUE, mecol = "brown", legend = TRUE, points = TRUE,
             depth_params = list(method = "LP"),
             graph_params = list(
               xlab = "infant mortality rate per 1000 live birth",
               ylab = "against masles immunized percentage",
               main = "L2 depth, UN Fourth Goal 2011 year"))


#EXAMPLE 3
data("france")
depthContour(france,
    depth_params = list(method = "Tukey"),
    points = TRUE
)
```

`DepthCurve-class` *DepthCurve*

Description

This page describes mechanism behavior of ScaleCurve and AsymmetryCurve

Details

DepthCurve is a virtual class that contains methods (getPlot(...) and plot(...)) for rendering single curve such as ScaleCurve or AsymmetryCurve. Such object can be combined by overloaded operator '

Slots

`depth` object of Depth-class

`name` name of dataset used on plot

`title` title of a plot

`alpha` central area values

Examples

```
library(mvtnorm)
x <- mvrnorm(n = 100, mu = c(0, 0), Sigma = 2 * diag(2))
y <- rmvt(n = 100, sigma = diag(2), df = 4)
s1 <- scaleCurve(x, depth_params = list(method = "Projection"))
s2 <- scaleCurve(y, depth_params = list(method = "Projection"), name = "Set2")

sc_list <- combineDepthCurves(s1, s2) # Add one curve to another

plot(sc_list) # Draw plot with two curves

z <- mvrnorm(n = 100, mu = c(0, 0), Sigma = 1 * diag(2))
s3 <- scaleCurve(z, depth_params = list(method = "Projection"))
plot(combineDepthCurves(sc_list, s3)) # Add third curve and draw a plot
```

`DepthCurveList-class` *DepthCurveList*

Description

DepthCurveList is a special container for DepthCurve objects. See DepthCurve-class

---

`depthDensity` *Depth weighted density estimator*

Description

Experimental function used to fit depth weighted density estimator.

Usage

```
depthDensity(x, y, nx = 5, ny = 32, xg = NULL, yg = NULL, ...)
```

Arguments

`x`	numeric vector
`y`	numeric vector
`nx`	the number of equally spaced points at which the density is to be estimated in x-dimension.
`ny`	the number of equally spaced points at which the density is to be estimated in x-dimension.
`xg`	vector of point at which the density is to be estimated.
`yg`	vector of point at which the density is to be estimated.
`...`	arguments passed to depthLocal.

References

Kosiorowski D. and Zawadzki Z. (2014) Notes on optimality of predictive distribution pseudo-estimators in the CHARME models and automatic trading strategies, FindEcon2014, submitted

Examples

```
## Not run:
# .sampleData is special function for creating
# data for testing conditional denisty estimators
data <- DepthProc:::.sampleData(1:5, 100)
x <- data[, 1]
y <- data[, 2]
plot(x, y)
dep <- depthDensity(x, y)
plot(dep, type = "raw")
plot(dep, type = "depth")

## End(Not run)
```

`DepthDensity-class` *DepthDensity*

Description

Class for depth based density estimator.

Details

`depthDensity`

`depthEuclid` *Euclidean Depth*

Description

Computes the euclidean depth of a point or vectors of points with respect to a multivariate data set.

Usage

```
depthEuclid(u, X)
```

Arguments

u	Numerical vector or matrix 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).

Details

Calculation of Euclidean depth is exact.

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

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
x <- matrix(rnorm(9999), nc = 3)
depthEuclid(x, x)
```

`depthLocal` *Local depth*

Description

Computes local version of depth according to proposals of Paindaveine and Van Bever — see referencess.

Usage

```
depthLocal(
  u,
  X,
  beta = 0.5,
  depth_params1 = list(method = "Projection"),
  depth_params2 = depth_params1
)
```

Arguments

`u`	Numerical vector or matrix 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. If it is a matrix or data frame, then each row is viewed as one multivariate observation.
`beta`	cutoff value for neighbourhood
`depth_params1`	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).
`depth_params2`	as above — default is depth_params1.

Details

A successful concept of local depth was proposed by Paindaveine and Van Bever (2012). For defining a neighbourhood of a point authors proposed using idea of symmetrisation of a distribution (a sample) with respect to a point in which depth is calculated. In their approach instead of a distribution P^X, a distribution $P_x = \frac{1}{2}P^X + \frac{1}{2}P^{2x-X}$ is used. For any $\beta \in [0,1]$, let us introduce the smallest depth region bigger or equal to β,

$$R^{\beta}(F) = \bigcap_{\alpha \in A(\beta)} D_{\alpha}(F),$$

where $A(\beta) = \{\alpha \geq 0 : P\left[D_{\alpha}(F)\right] \geq \beta\}$. Then for a locality parameter β we can take a neighbourhood of a point x as $R^{\beta}_{x}(P)$.

Formally, let $D(\cdot, P)$ be a depth function. Then the local depth with the locality parameter β and w.r.t. a point x is defined as

$$LD^{\beta}(z, P) : z \rightarrow D(z, P^{\beta}_{x}),$$

where $P^{\beta}_{x}(\cdot) = P\left(\cdot | R^{\beta}_{x}(P)\right)$ is cond. distr. of P conditioned on $R^{\beta}_{x}(P)$.

References

Paindaveine, D., Van Bever, G. (2013) From depth to local depth : a focus on centrality. Journal of the American Statistical Association 105, 1105–1119.

Examples

```
## Not run:
# EXAMPLE 1
data <- mvrnorm(100, c(0, 5), diag(2) * 5)
# By default depth_params2 = depth_params1
depthLocal(data, data, depth_params1 = list(method = "LP"))
depthLocal(data, data, depth_params1 = list(method = "LP"),
           depth_params2 = list(method = "Projection"))
# Depth contour
depthContour(data, depth_params = list(method = "Local", depth_params1 = list(method = "LP")))

# EXAMPLE 2
data(inf.mort, maesles.imm)
```

```
data1990 <- na.omit(cbind(inf.mort[, 1], maesles.imm[, 1]))
depthContour(data1990,
             depth_params = list(
               method = "Local",
               depth_params1 = list(method = "LP"),
               beta = 0.3
             ))

# EXAMPLE 3
Sigma1 <- matrix(c(10, 3, 3, 2), 2, 2)
X1 <- mvrnorm(n = 8500, mu = c(0, 0), Sigma1)
Sigma2 <- matrix(c(10, 0, 0, 2), 2, 2)
X2 <- mvrnorm(n = 1500, mu = c(-10, 6), Sigma2)
BALLOT <- rbind(X1, X2)

train <- sample(1:10000, 100)
data <- BALLOT[train, ]
depthContour(data,
            depth_params = list(
              method = "Local",
              beta = 0.3,
              depth_params1 = list(method = "Projection")
            ))

## End(Not run)
```

---

depthLP *LP Depth*

Description

Computes the LP depth of a point or vectors of points with respect to a multivariate data set.

Usage

```
depthLP(u, X, pdim = 2, la = 1, lb = 1, threads = -1, func = NULL)
```

Arguments

u	Numerical vector or matrix 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).
pdim	dimension used in calculating depth function.
la	slope the weighing function.
lb	intercept in the weighing function.

`threads` number of threads used in parallel computations. Default value -1 means that all possible cores will be used.

`func` the weighing function. Currently it is not supported.

Details

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

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
x <- matrix(rnorm(3000), ncol = 3)

# Same results
depthLP(x, x, pdim = 2)
```

---

depthMah *Mahalanobis Depth*

Description

Computes the mahalanobis depth of a point or vectors of points with respect to a multivariate data set.

Usage

```
depthMah(u, X, cov = NULL, mean = NULL, threads = -1)
```

Arguments

`u` Numerical vector or matrix 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).

`cov` custom covariance matrix passed. If NULL standard calculations will be based on standard covariance estimator.

`mean` custom mean vector. If null — mean average will be used.

`threads` number of threads used in parallel computations. Default value -1 means that all possible cores will be used.

Details

Calculation of Mahalanobis depth is exact.

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

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
x <- matrix(rnorm(9999), nc = 3)
depthMah(x, x)
```

`depthMedian` *Depth median*

Description

Return point with maximum depth function value. If multiple points have the same value, mean average of them will be returned.

Usage

```
depthMedian(x, depth_params = list(), convex = FALSE)

## S4 method for signature 'matrix'
depthMedian(x, depth_params = list(), convex = FALSE)

## S4 method for signature 'data.frame'
depthMedian(x, depth_params = list(), convex = FALSE)

## S4 method for signature 'Depth'
depthMedian(x, convex = FALSE)
```

Arguments

x	object of class Depth or matrix.
depth_params	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).
convex	logical. If true, than centroid of the convex hull created from deepest points is returned.

Examples

```
# depthMedian for matrix
x <- matrix(rnorm(600), nc = 3)
depthMedian(x)

# depthMedian works with object of class Depth
dp <- depth(x)
depthMedian(dp)
```

---

depthPersp *Perspective plot for depth functions*

Description

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

Usage

```
depthPersp(
  x,
  plot_method = "lattice",
  xlim = extendrange(x[, 1], f = 0.1),
  ylim = extendrange(x[, 2], f = 0.1),
  n = 50,
  xlab = "x",
  ylab = "y",
  plot_title = NULL,
  colors = heat_hcl,
  depth_params = list(),
  graph_params = list()
)
```

Arguments

x	bivariate data
plot_method	there are two options "lattice", and "rgl" — see details
xlim	limits for x-axis
ylim	limits for y-axis
n	number of points that will be used to create plot (n^2)
xlab	description of x-axis
ylab	description of y-axis
plot_title	plot title (default NULL means paste(depth_params$method, "depth"))
colors	function for colors pallete (e.g. gray.colors).

`depth_params` list of parameters for function depth ("method", "threads", "ndir", "la", "lb", "pdim", "mean", "cov", "exact").

`graph_params` list of graphical parameters for functions rgl::persp3d and lattice::wireframe.

Details

plot_method — rgl package is not in depends list beacuse it may cause problems when OpenGL is not supported. To use plot_method = "rgl" you must load this package on your own.

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
# EXAMPLE 1
x <- mvrnorm(100, c(0, 0), diag(2))
depthPersp(x, depth_params = list(method = "Euclidean"))

# EXAMPLE 2
data(inf.mort, maesles.imm)
data1990 <- na.omit(cbind(inf.mort[, 1], maesles.imm[, 1]))

## Not run:
library(rgl)
depthPersp(data1990, plot_method = "rgl",
           depth_params = list(method = "Projection"))

## End(Not run)
```

`depthProjection` *Projection Depth*

Description

Computes the Projection depth of a point or vectors of points with respect to a multivariate data set.

Usage

```
depthProjection(u, X, ndir = 1000, threads = -1)
```

Arguments

`u`	Numerical vector or matrix 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).
`ndir`	number of directions used in computations
`threads`	number of threads used in parallel computations. Default value -1 means that all possible cores will be used.

Details

Irrespective of dimension, Projection and Tukey's depth is obtained by approximate calculation.

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

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
x <- matrix(rnorm(3000), nc = 3)
a <- depthProjection(x, x, ndir = 2000)
```

---

depthTukey — *Tukey Depth*

Description

Computes the Tukey depth of a point or vectors of points with respect to a multivariate data set.

Usage

```
depthTukey(u, X, ndir = 1000, threads = -1, exact = FALSE)
```

Arguments

`u`	Numerical vector or matrix 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).
`ndir`	number of directions used in computations

threads	number of threads used in parallel computations. Default value -1 means that all possible cores will be used.
exact	if TRUE exact alhorithm will be used . Currently it works only for 2 dimensional data set.

Details

Irrespective of dimension, Projection and Tukey's depth is obtained by approximate calculation.

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

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
## Not run:
x <- matrix(rnorm(3000), nc = 3)
depthTukey(x, ndir = 2000)

## End(Not run)

# Exact algorithm in 2d
x <- matrix(rnorm(2000), nc = 2)
depthTukey(x, exact = TRUE)
```

---

fncBoxPlot *Functional boxplot based on Modified Band Depth*

Description

Functional boxplot based on Modified Band Depth

Usage

```
fncBoxPlot(u, X = NULL, bands = c(0, 0.5), method = "MBD", byrow = NULL, ...)
```

Arguments

u	data matrix
X	reference set. If null u will be used as reference.
bands	limits for bands
method	depth method
byrow	byrow
...	other arguments passed to fncDepth

Examples

```
# some data:
x <- matrix(rnorm(200), ncol = 10)

fncBoxPlot(x, bands = c(0, 0.5, 1), method = "FM")
fncBoxPlot(x, bands = c(0, 0.5, 1), method = "FM", byrow = FALSE)

colnames(x) <- paste0("f", 1:ncol(x))
fncBoxPlot(x, bands = c(0, 0.5, 1), method = "FM")

# fncBoxPlot handles zoo and xts objects
library(xts)
x <- matrix(rnorm(200), ncol = 10)
time <- as.POSIXct(1:ncol(x) * 86400, origin = "1970-01-01")
x_xts <- xts(t(x), order.by = time)
fncBoxPlot(x_xts, bands = c(0, 0.5, 1), method = "FM")

data("katowice.airpollution")
pl <- fncBoxPlot(katowice.airpollution, bands = c(0, 0.5, 1), method = "MBD")
pl + ggtitle("Air pollution in Katowice") + labs(y= "pollination ", x = "hour ")
```

---

fncDepth *Basic function for functional depths*

Description

Calculates depth functions.

Usage

```
fncDepth(u, X = NULL, method = "MBD", byrow = NULL, ...)

## S3 method for class 'matrix'
fncDepth(u, X = NULL, method = "MBD", byrow = NULL, ...)

## S3 method for class 'zoo'
fncDepth(u, X = NULL, method = "MBD", byrow = NULL, ...)
```

Arguments

`u`	data
`X`	reference set. If null u will be used as reference.
`method`	depth method - "MBD" (default), or "FM" (Frainman-Muniz depth)
`byrow`	logical or character.
`...`	additional arguments passed to fncDepthFM.

Examples

```
x <- matrix(rnorm(60), ncol = 20)
fncDepth(x, method = "FM", dep1d = "Mahalanobis")
fncDepth(x, byrow = FALSE)

# zoo and xts
library(xts)
data(sample_matrix)
sample.xts <- as.xts(sample_matrix, descr = "my new xts object")
fncDepth(sample.xts)
```

`fncDepthFM` *FM Depth*

Description

Computes Frainman-Muniz depth for functional data.

Usage

```
fncDepthFM(u, X, dep1d_params = list(method = "Projection"))
```

Arguments

`u`	Numerical vector or matrix whose depth is to be calculated. Dimension has to be the same as that of the observations.
`X`	The data as a matrix. If it is a matrix or data frame, then each row is viewed as one multivariate observation.
`dep1d_params`	parameters passed to depth function used in one dimension.

Examples

```
x <- matrix(rnorm(60), nc = 20)
fncDepthFM(x)
```

`fncDepthMBD` *Modified band depth*

Description

Computes the modified band depth.

Usage

```
fncDepthMBD(u, X)
```

Arguments

`u`	Numerical vector or matrix whose depth is to be calculated. Dimension has to be the same as that of the observations.
`X`	The data as a matrix. If it is a matrix or data frame, then each row is viewed as one multivariate observation.

Examples

```
x <- matrix(rnorm(60), nc = 20)
fncDepthMBD(x)
fncDepthMBD(x, x)
```

`fncDepthMedian` *Functional median*

Description

Calculate functional median based on data depth.

Usage

```
fncDepthMedian(u, X = NULL, method = "MBD", byrow = NULL, unique = TRUE, ...)
```

Arguments

`u`	data matrix
`X`	reference set. If null u will be used as reference.
`method`	depth method
`byrow`	byrow
`unique`	if true
`...`	other arguments passed to fncDepth

Examples

```
x <- matrix(rnorm(600), nc = 20)
md <- fncDepthMedian(x, method = "FM", dep1d = "Mahalanobis")
```

---

fncGetBand — *Functional bands*

Description

Extract bands from functional depth object.

Usage

```
fncGetBand(obj, band = 0.5)
```

Arguments

obj	object that inherits from FunctionalDepth.
band	single numeric value.

Examples

```
x <- matrix(rnorm(600), nc = 20)
obj <- fncDepth(x, method = "FM", dep1d = "Mahalanobis")
fncGetBand(obj)
```

---

france — *Relation between minimum wage (MW) and unemployment rate (UR) in France.*

Description

Relation between minimum wage (MW) and unemployment rate (UR) in France.

Usage

```
data(france)
```

Format

data frame containing 17 rows and two column. MW is a minimum wage, and UR is an unemployment rate.

FunctionalDepth-class *Functional Depth*

Description

Virtual class with structure for every functional depth class from depthproc package. Inherits from `Depth-class`.

Slots

`index` numeric, or time-based object.

getPlot *Create ggplot object from DepthCurve, DepthCurveList and DDPlot classes.*

Description

Create an object of class ggplot from DepthCurve and DepthCurveList.

Usage

```
getPlot(object)

## S4 method for signature 'AsymmetryCurveList'
getPlot(object)

## S4 method for signature 'DDPlot'
getPlot(object)

## S4 method for signature 'ScaleCurveList'
getPlot(object)
```

Arguments

`object` a DDPlot ScaleCurve or AsymmetryCurve object class.

`inf.mort` *Infant mortality rate (0–1 year) per 1,000 live births*

Description

Infant mortality rate (0–1 year) per 1,000 live births

Usage

```
data(inf.mort)
```

Format

A data frame with 654 rows and 4 variables

Source

http://mdgs.un.org/unsd/mdg/Data.aspx

`internet.users` *Internet view data*

Description

Internet view data

Usage

```
data(internet.users)
```

Format

data frame containing 17518 rows and 6 columns — 17518 working days of the Internet service considered with respect to variables: service, month, day, hour, unique users and page views.

References

Kosiorowski, Rydlewski, Snarska (2016), Detecting a Structural Change in Functional Time Series Using Local Wilcoxon Statistic, arXiv: 1604.03776v2

`katowice.airpollution` *Air pollution in Katowice city by hour.*

Description

Air pollution in Katowice city by hour.

Usage

```
data("katowice.airpollution")
```

Format

data frame containing 181 rows (days) and 24 columns. Each column is an air pollination for given hour.

`lsdAddContour` *Adds location scale depth contour to the existing plot.*

Description

This function add one location-scale contour to the existing plot.

Usage

```
lsdAddContour(x, cont = NULL, ...)

## S4 method for signature 'LSDepthContour'
lsdAddContour(x, cont = NULL, ...)
```

Arguments

`x`	object of class LSDepthContour
`cont`	depth of contour to plot
`...`	other arguments passed to polygon function

Examples

```
smp <- rf(100, 5, 10)
x <- lsdSampleDepthContours(smp)
plot(x)
lsdAddContour(x, 0.1, col = "grey50")
lsdAddContour(x, 0.3, col = "grey10", border = "red", lwd = 4)
```

`LSDepth-class` *Location-Scale depth class*

Description

Class used to store maximum location-scale depth results.

Slots

`max_depth` maximum Student depth value.

`mu` location estimate in the deepest point.

`sigma` scale estimate in the deepest point.

`LSDepthContour-class` *Location-Scale depth contour class*

Description

Class used to store result of location-scale depth contours.

Slots

`cont_depth` depth values used to calculate contours.

`sample` original sample used to calculate depth contours.

`.Data` list with estimated values of scale-depth contours.

`lsdGetContour` *Get location-scale contour from LSDepthContour object.*

Description

Get numeric values of the location-scale depth contour from existing object of LSDepthContour class.

Usage

```
lsdGetContour(x, cont)

## S4 method for signature 'LSDepthContour'
lsdGetContour(x, cont)
```

Arguments

x	object of class LSDepthContour
cont	single numeric — depth of contour to return

Details

Calculations are based on lsdepth algorithm written by Ch. Muller.

Examples

```
dcont <- lsdSampleDepthContours(rf(200, 4, 7), depth = c(0.1, 0.2))

# get contour that is present in dcont object
lsdGetContour(dcont, 0.1)

# get contour that is not present in dcont
# it will be automatically calculated
lsdGetContour(dcont, 0.3)
```

---

lsdSampleDepthContours

Calculate sample Mizera and Muller Student depth contours

Description

Calculate sample one-dimensional Mizera and Muller Student depth contours.

Usage

```
lsdSampleDepthContours(x, depth = c(0.1, 0.2, 0.3, 0.4), lengthmu = 1000)
```

Arguments

x	one dimensional vector with sample
depth	depth level for contours
lengthmu	number of points to evalute depth

Details

Calculations are based on lsdepth algorithm written by Ch. Muller.

References

Mizera, I., Muller, C. H., 2004. Location-scale depth (with discussion). Journal of the American Statistical Association 99, 949–966.

Examples

```
# EXAMPLE 1
# F-distribution
dcont <- lsdSampleDepthContours(rf(200, 4, 7))
plot(dcont)

# EXAMPLE 2
# normal distribution - more contours calculated
dcont_norm <- lsdSampleDepthContours(rnorm(100), seq(0.05, 0.4, 0.05))
plot(dcont_norm)
```

---

lsdSampleMaxDepth *Calculates the maximum sample location-scale depth*

Description

Calculates the maximum Student depth estimator of location and scale for one dimensional data (an alternative for MED and MAD or for the mean and standard deviation).

Usage

```
lsdSampleMaxDepth(x, iter = 100, eps = 1e-04, p_length = 10)
```

Arguments

x	one dimensional vector with sample
iter	maximum number of iterations in algorith for calculation Location-Scale Depth
eps	tolerance level
p_length	is the maximum length of the precision step at the end

Details

Calculations are based on lsdepth algorithm written by Ch. Muller.

References

Mizera, I., Muller, C. H., 2004. Location-scale depth (with discussion). Journal of the American Statistical Association 99, 949–966.

Examples

```
x <- rnorm(100)
lsdSampleMaxDepth(x)
y <- rf(100, 4, 10)
lsdSampleMaxDepth(y)
```

maesles.imm *Children 1 year old immunized against measles, percentage*

Description

Children 1 year old immunized against measles, percentage

Usage

```
data(maesles.imm)
```

Format

A data frame with 654 rows and 4 variables

Source

http://mdgs.un.org/unsd/mdg/Data.aspx

mWilcoxonTest *Multivariate Wilcoxon test for equality of dispersion.*

Description

Depth based multivariate Wilcoxon test for a scale difference.

Usage

```
mWilcoxonTest(x, y, alternative = "two.sided", depth_params = list())
```

Arguments

x	data matrix
y	data matrix
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
depth_params	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).

Details

Having two samples X^n and Y^m using any depth function, we can compute depth values in a combined sample $Z^{n+m} = X^n \cup Y^m$, assuming the empirical distribution calculated basing on all observations, or only on observations belonging to one of the samples X^n or Y^m.

For example if we observe $X_l's$ depths are more likely to cluster tightly around the center of the combined sample, while $Y_l's$ depths are more likely to scatter outlying positions, then we conclude Y^m was drawn from a distribution with larger scale.

Properties of the DD plot based statistics in the i.i.d setting were studied by Li & Liu (2004). Authors proposed several DD-plot based statistics and presented bootstrap arguments for their consistency and good effectiveness in comparison to Hotelling T^2 and multivariate analogues of Ansari-Bradley and Tukey-Siegel statistics. Asymptotic distributions of depth based multivariate Wilcoxon rank-sum test statistic under the null and general alternative hypotheses were obtained by Zuo & He (2006). Several properties of the depth based rang test involving its unbiasedness was critically discussed by Jureckova & Kalina (2012). Basing on DD-plot object, which is available within the **DepthProc** it is possible to define several multivariate generalizations of one-dimensional rank and order statistics in an easy way. These generalizations cover well known Wilcoxon rang-sum statistic.

The depth based multivariate Wilcoxon rang sum test is especially useful for the multivariate scale changes detection and was introduced among other by Liu & Singh (2003) and intensively studied by Jureckowa & Kalina (2012) and Zuo & He (2006) in the i.i.d. setting.

For the samples $X^m = \{X_1, ..., X_m\}$, $Y^n = \{Y_1, ..., Y_n\}$, their $d_1^X, ..., d_m^X$, $d_1^Y, ..., d_n^Y$, depths w.r.t. a combined sample $Z = X^n \cup Y^m$ the Wilcoxon statistic is defined as $S = \sum_{i=1}^{m} R_i$, where R_i denotes the rang of the i-th observation, $i = 1, ..., m$ in the combined sample $R(y_l) = \sharp\left\{z_j \in Z : D(z_j, Z) \leq D(y_l, Z)\right\}, l = 1, ..., m$.

The distribution of S is symmetric about $E(S) = \frac{1}{2}m(m + n + 1)$, its variance is $D^2(S) = \frac{1}{12}mn(m + n + 1)$.

References

Jureckova J, Kalina J (2012). Nonparametric multivariate rank tests and their unbiasedness. Bernoulli, 18(1), 229–251. Li J, Liu RY (2004). New nonparametric tests of multivariate locations and scales using data depth. Statistical Science, 19(4), 686–696. Liu RY, Singh K (1995). A quality index based on data depth and multivariate rank tests. Journal of American Statistical Association, 88, 252–260. Zuo Y, He X (2006). On the limiting distributions of multivariate depth-based rank sum statistics and related tests. The Annals of Statistics, 34, 2879–2896.

Examples

```
# EXAMPLE 1
x <- mvrnorm(100, c(0, 0), diag(2))
y <- mvrnorm(100, c(0, 0), diag(2) * 1.4)
mWilcoxonTest(x, y)
mWilcoxonTest(x, y, depth_params = list(method = "LP"))

# EXAMPLE 2
data(under5.mort)
```

```
data(inf.mort)
data(maesles.imm)
data2011 <- na.omit(cbind(under5.mort[, 22], inf.mort[, 22],
                          maesles.imm[, 22]))
data1990 <- na.omit(cbind(under5.mort[, 1], inf.mort[, 1], maesles.imm[, 1]))
mWilcoxonTest(data2011, data1990)
```

---

plot *Method for plotting DepthCurve and DDPlot object.*

Description

Plot Depth curve

Usage

```
plot(x, y, ...)

## S4 method for signature 'DDPlot,ANY'
plot(x)

## S4 method for signature 'DepthCurve,ANY'
plot(x)

## S4 method for signature 'DepthCurveList,ANY'
plot(x)
```

Arguments

x	object that inherits from DepthCurve class (ScaleCurve or AsymmetryCurve), or DDPlot class.
y	not supported.
...	not supported.

Examples

```
x <- mvrnorm(n = 100, mu = c(0, 0), Sigma = 3 * diag(2))
sc <- scaleCurve(x)
plot(sc)
```

`plot,BinnDepth2d,ANY-method`
2d Binning plot

Description

Binning 2d

Usage

```
## S4 method for signature 'BinnDepth2d,ANY'
plot(x, ..., alpha = 0.1, bg_col = "red", add_mid = TRUE)
```

Arguments

`x`	object of class BinnDepth2d
`...`	graphical parameters passed to plot
`alpha`	alpha value for rgb function
`bg_col`	backgroud color
`add_mid`	logical. If TRUE centers of binns will be marked.

See Also

`depth`

Examples

```
tmp <- binningDepth2D(x = mvrnorm(100, rep(0, 2), diag(2)))
plot(tmp)
```

`plot,DepthDensity,ANY-method`
Plot function for DepthDensity.

Description

Create plot for DepthDensity. See `depthDensity` for more information.

Usage

```
## S4 method for signature 'DepthDensity,ANY'
plot(x, type = "depth", ...)
```

Arguments

x	object of class DepthDensity
type	type of density that will be plotted. "depth" is a depth scaled density, and "raw" is denisty without scaling.
...	graphical arguments.

---

plot,LSDepthContour,ANY-method

Plot Location-Scale depth contours.

Description

Create location-scale depth plot. See lsdSampleDepthContours for more information.

Usage

```
## S4 method for signature 'LSDepthContour,ANY'
plot(
  x,
  cont = NULL,
  ratio = 1,
  mu_min = NULL,
  mu_max = NULL,
  col = NULL,
  border = NULL,
  ...
)
```

Arguments

x	object of class LSDepthContour
cont	plotted contours. Default NULL means that all contours stored in x will be plotted.
ratio	ratio
mu_min	mu_min
mu_max	mu_max
col	vectors with area colors passed to polygon function
border	vector with colors for borders
...	other parameters passed to polygon

Examples

```
smp <- rf(100, 5, 10)
x <- lsdSampleDepthContours(smp)
plot(x, col = paste0("grey", col = rev(seq(10, 40, 10))))
```

RobReg-class *RobReg*

Description

Virtual class for robust regression methods from depthproc package

Slots

`coef` coefficients of fitted model

runifsphere *Random number generation from unit sphere.*

Description

This function generates random numbers from p-dimensional unit sphere.

Usage

```
runifsphere(n, p = 2)
```

Arguments

`n`	number of random samples.
`p`	dimension of the unit sphere.

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
x <- runifsphere(n = 100)
plot(x)
```

`scaleCurve` *Scale curve*

Description

Draws a scale curve: measure of dispersion.

Usage

```
scaleCurve(
  x,
  y = NULL,
  alpha = seq(0, 1, 0.01),
  name = "X",
  name_y = "Y",
  title = "Scale Curve",
  depth_params = list(method = "Projection")
)
```

Arguments

x	Multivariate data as a matrix.
y	Additional matrix with multivariate data.
alpha	Vector with values of central area to be used in computation.
name	Name of matrix X used in legend.
name_y	Name of matrix Y used in legend.
title	title of the plot.
depth_params	list of parameters for function depth (method, threads, ndir, la, lb, pdim, mean, cov, exact).

Details

For sample depth function $D(x, Z^n)$, $x \in R^d$, $d \geq 2$, $Z^n = \{z_1, ..., z_n\} \subset R^d$, $D_\alpha(Z^n)$ denoting α — central region, we can define the scale curve $SC(\alpha) = (\alpha, vol(D_\alpha(Z^n)) \subset R^2$, for $\alpha \in [0, 1]$

The scale curve is a two-dimensional method of describing the dispersion of random vector around the depth induced median.

Function scalecurve for determining the volumes of the convex hull containing points from alpha central regions, uses function convhulln from geometry package.

The minimal dimension of data in X or Y is 2.

ggplot2 package is used to draw a plot.

Value

Returns the volume of the convex hull containing subsequent central points of `X`.

Author(s)

Daniel Kosiorowski, Mateusz Bocian, Anna Wegrzynkiewicz and Zygmunt Zawadzki from Cracow University of Economics.

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.

Chaudhuri, P. (1996), On a Geometric Notion of Quantiles for Multivariate Data, *Journal of the American Statistical Association*, 862–872.

Dyckerhoff, R. (2004), Data Depths Satisfying the Projection Property, *Allgemeines Statistisches Archiv.*, **88**, 163–190.

See Also

`depthContour` and `depthPersp` for depth graphics.

Examples

```
library(mvtnorm)
x <- mvrnorm(n = 100, mu = c(0, 0), Sigma = 3 * diag(2))
y <- rmvt(n = 100, sigma = diag(2), df = 2)
scaleCurve(x, y, depth_params = list(method = "Projection"))
# Comparing two scale curves
# normal distribution and mixture of normal distributions
x <- mvrnorm(100, c(0, 0), diag(2))
y <- mvrnorm(80, c(0, 0), diag(2))
z <- mvrnorm(20, c(5, 5), diag(2))
scaleCurve(x, rbind(y, z), name = "N", name_y = "Mixture of N",
           depth_params = list(method = "Projection"))
```

`ScaleCurve-class` *ScaleCurve and ScaleCurveList*

Description

ScaleCurve is a class that stores results of scaleCurve function.

Details

ScaleCurve intherits behviour from numeric vector, so raw values of ScaleCurve can be accessed via as.numeric(...).

The mechanism of creating plots with multiple curves is shown in DepthCurve-class (same mechanism is applied for AsymmetryCurve).

Examples

```
library(mvtnorm)
x <- mvrnorm(n = 100, mu = c(0, 0), Sigma = 2 * diag(2))
y <- rmvt(n = 100, sigma = diag(2), df = 4)
s1 <- scaleCurve(x, depth_params = list(method = "Projection"))
s2 <- scaleCurve(y, depth_params = list(method = "Projection"), name = "Set2")

sc_list <- combineDepthCurves(s1, s2) # Add one curve to another

plot(sc_list) # Draw plot with two curves

z <- mvrnorm(n = 100, mu = c(0, 0), Sigma = 1 * diag(2))
s3 <- scaleCurve(z, depth_params = list(method = "Projection"))
plot(combineDepthCurves(sc_list, s3)) # Add third curve and draw a plot
```

---

trimProjReg2d *trimProjReg2d*

Description

Computes projection trimmed regression in 2 dimensions.

Usage

```
trimProjReg2d(x, y, alpha = 0.1)
```

Arguments

x	Independent variable
y	Dependent variable
alpha	Percentage of trimmed observations

Author(s)

Zygmunt Zawadzki from Cracow University of Economics.

Examples

```
# EXAMPLE 1
data(pension)
plot(pension)
abline(lm(Reserves ~ Income, data = pension), lty = 3, lwd = 2) # lm
abline(trimProjReg2d(pension[, 1], pension[, 2]), lwd = 2) # trimprojreg2d
legend("bottomright", c("OLS", "TrimLS"), lty = 1:2)

# EXAMPLE 2
```

```
data(under5.mort)
data(inf.mort)
data(maesles.imm)

data2011 <- na.omit(cbind(under5.mort[, 22], inf.mort[, 22],
                          maesles.imm[, 22]))
x <- data2011[, 3]
y <- data2011[, 2]
plot(x, y, cex = 1.2, ylab = "infant mortality rate per 1000 live birth",
     xlab = "against masles immunized percentage",
     main = "Projection Depth Trimmed vs. LS regressions")
abline(lm(x ~ y), lwd = 2, col = "black") # lm
abline(trimProjReg2d(x, y), lwd = 2, col = "red") # trimmed reg
legend("bottomleft", c("LS", "TrimReg"), fill = c("black", "red"), cex = 1.4,
       bty = "n")

##### Comparsion of a few regression methods #####
library(DepthProc)
library(MASS)

data("france")
plot(UR ~ MW, pch = 19, data = france)

# linear regression
lm.fit <- lm(UR ~ MW, data = france)
abline(lm.fit, lwd=2, cex=3, col='red')

# M-estimator
rlm.fit <- rlm(UR ~ MW, data = france)
abline(rlm.fit, lwd = 2,col = "blue")

# LMS
lqs.lms <- lqs(UR ~ MW, method = "lms", data = france) #least median of squares#
lqs.lts <- lqs(UR ~ MW, method = "lts", data = france) #least trimmed squares#
abline(lqs.lms, lwd = 2, col="green")
abline(lqs.lts, lwd = 2, col="pink")

# Lowess
lines(lowess(france$MW, france$UR, f = 0.5, iter = 0), lwd = 2) # loess

# Depth trimmed regression
trim.reg <- trimProjReg2d(france$MW, france$UR) #trimprojreg2d
abline(trim.reg, lwd = 4, col = 'orange')
```

`TrimReg2d-class` *TrimReg2d*

Description

Class for robust regression methods from depthproc package `under5.mort` *Children under 5 months mortality rate per 1,000 live births*

Description

Children under 5 months mortality rate per 1,000 live births

Usage

```
data(under5.mort)
```

Format

A data frame with 654 rows and 4 variables

Source

http://mdgs.un.org/unsd/mdg/Data.aspx

`USLABOUR` *US Labour dataset*

Description

US Labour dataset

Usage

```
data(USLABOUR)
```

Format

A data frame with 654 rows and 4 variables

Source

U.S.Department of Labor — Bureau of Labour Statistics FRED

Index