Package 'denpro'

May 22, 2015

Version 0.9.2

Date 2015-04-24

Title Visualization of Multivariate Functions, Sets, and Data

Author Jussi Klemela <jussi.klemela@gmail.com>

Maintainer Jussi Klemela <jussi.klemela@gmail.com>

Depends

Description We provide tools to
(1) visualize multivariate density functions and density estimates
with level set trees,
(2) visualize level sets with shape trees,
(3) visualize multivariate data with tail trees,
(4) visualize scales of multivariate density estimates with
mode graphs and branching maps, and
(5) visualize anisotropic spread with 2D volume functions and
2D probability content functions.
Level set trees visualize mode structure,
shape trees visualize shapes of level sets of unimodal densities,
and tail trees visualize connected data sets.
The kernel estimator is implemented
but the package may also be applied for visualizing other density estimates.

License GPL (>= 2)

URL http://jussiklemela.com/denpro/

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-05-22 19:38:40

R topics documented:

`denpro-package` *Visualization of Multivariate Functions, Sets, and Data*

Description

We provide tools to (1) visualize multivariate functions and density estimates with level set trees, (2) visualize level sets with shape trees, (3) visualize multivariate data with tail trees, (4) visualize scales of multivariate density estimates with mode graphs and branching maps, and (5) visualize anisotropic spread with 2D volume functions and 2D probability content functions. Level set trees visualize mode structure, shape trees visualize shapes of level sets of unimodal densities, and tail

trees visualize connected data sets. The kernel estimator is implemented but the package may also be applied for visualizing other density estimates.

Details

Package:	denpro
Version:	0.9.2
Date:	2015-05-12
Depends:	
License:	GPL(>=2)
URL:	http://www.jussiklemela.com/denpro/
Packaged:	Mon Jun 15 16:34:17 2015; jsk
Built:	R 3.2.0; i486-pc-linux-gnu; 2013-06-12 16:34:35; unix

The main function:

*leafsfirst

Functions to plot a level set tree, shape tree, or tail tree:

*plotvolu *plotbary *plottree *treedisc *prunemodes

Visualization of scales of estimates:

* lstseq.kern * modegraph * plotmodet * branchmap * plotbranchmap * scaletable * exmap * plotexmap * hgrid :

Visualization of anisotropic spread (2D volume function and 2D probability content function):

* stseq * shape2d * plotvolu2d * plotdelineator

Other utilities:

* pcf.kern * tree.segme * lst2xy * graph.matrix * plotbary.slide * Calculation of level set trees with "DynaDecompose" algorithm and with naive algorithm o profkern o profhist o profgene o profeval * locofmax * modecent * excmas

Other visualization tools:

* draw.pcf * slicing * draw.levset * paracoor * pp.plot * qq.plot * plot.kernscale * dist.func * paraclus

Miscallenous:

* pcf.func * sim.data * dend2parent * explo.compa * nn.radit * nn.likeset * rotation * liketree * leafsfirst.visu * plottwin

Index (alphaphetical):

```
branchmap               Calculates a branching map from a sequence of
                        level set trees
draw.levset             Plots a level set of a 2D function
draw.pcf                Prepares the plotting of a 2D or 1D piecewise
                        constant function
excmas                  Calculates the excess masses associated with
                        the nodes of a level set tree
```

exmap	Calculates a scale of excess mass profiles
hgrid	Returns a grid of smoothing parameter values
leafsfirst	Calculates a level set tree, shape tree, or tail tree
locofmax	Calculates the location of the maximum of a function
lstseq.kern	Calculates a scale of kernel estimates
modecent	Returns locations of modes of a density estimate
modegraph	Calculates a mode graph from a scale of estimates
paracoor	Makes a parallel coordinate plot
pcf.func	Calculates a piecewise constant function for some illustrative purposes
pcf.kern	Calculates a multivariate kernel estimate
plotbary	Makes a barycenter plot of a level set tree, a location plot of a shape tree, or a tail tree plot of a tail tree
plotbranchmap	Plots a branching map
plotexmap	Plots a scale of excess mass profiles
plotmodet	Plots a mode graph
plottree	Makes a tree plot of a level set tree, of a shape tree, or of a tail tree
plotvolu	Makes a volume plot of a level set tree, a shape plot of a shape tree, or a tail frequency plot of a tail tree
plotvolu2d	Makes a perspective plot of a 2D volume function or a 2D probability content function
profgene	Calculates the level set tree of a rectangularwise constant function
profhist	Calculates the level set tree of a histogram
profkern	Calculates a level set tree of a kernel estimate
prunemodes	Prunes modes away from a level set tree or a shape tree
scaletable	Implements the scale and shape table
shape2d	Returns a 2D volume function or 2D probability content function
sim.data	Generates data for illustrative purposes
slicing	Returns a one- or two-dimensional slice of a multivariate function
stseq	Calculates a sequence of radius functions from a sequence of level sets
treedisc	Prunes a level set tree or a tail tree
tree.segme	Returns the segmentation of the nodes of a visualization tree

Author(s)

Jussi Klemela <jussi.klemela@gmail.com>

Maintainer: Jussi Klemela <jussi.klemela@gmail.com>

References

http://www.rni.helsinki.fi/~jsk/denpro

Examples

```
# level set tree

dendat<-sim.data(n=200,type="mulmod")   # data
pcf<-pcf.kern(dendat,h=1,N=c(32,32))    # kernel estimate

lst<-leafsfirst(pcf)             # level set tree
td<-treedisc(lst,pcf,ngrid=60)   # pruned level set tree

plotvolu(td)                     # volume plot

plotbary(td)                     # barycenter plot

# shape tree

dendat<-sim.data(n=200,type="cross")    # data
pcf<-pcf.kern(dendat,h=1,N=c(32,32))    # kernel estimate

st<-leafsfirst(pcf,propo=0.01)   # shape tree, 1 per cent level set
tdst<-treedisc(st,pcf,ngrid=60)  # pruned shape tree

plotvolu(tdst)          # radius plot

plotbary(tdst)          # location plot

# tail tree

dendat<-sim.data(n=200,type="cross")    # data

tt<-leafsfirst(dendat=dendat,rho=0.65) # tail tree

plotbary(tt)     # tail tree plot

plotvolu(tt)     # tail frequency plot
```

---

branchmap *Computes a branching map from a sequence of level set trees*

Description

Branching map visualizes the levels of branching of level set trees of estimates belonging to a scale of estimates. It visualizes also the excess masses of the roots of the branches.

Usage

```
branchmap(estiseq, hseq = NULL, levnum = 80, paletti = NULL,
rootpaletti = NULL, type = "jump")
```

Arguments

estiseq	A sequence of estimates and level set trees of the estimates. Output of function lstseq.kern or function lstseq.carthisto.
hseq	The sequence of smoothing parameters of the scale of estimates.
levnum	The number of level sets used to approximate the level set trees.
paletti	A sequence of color names; colors for each branch, other than the root branches.
rootpaletti	A sequence of color names; colors for the root branches.
type	internal

Value

A representation as a list of a 2D function

level	x-coordinate is the level of the level sets
h	y-coordinate is the smoothing parameter
z	z-coordinate is the excess mass
col	colors for the graph of the 2D function

Author(s)

Jussi Klemela

See Also

lstseq.kern, plotbranchmap

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N,lstree=TRUE)
```

```
bm<-branchmap(estiseq)

plotbranchmap(bm)
```

`colors2data` *Assigns colors to data points*

Description

Assigns colors to data points according to cluster membership, when the clusters are defined as high density regions

Usage

```
colors2data(dendat, pcf, lst, paletti = NULL, clusterlevel = NULL, nodes = NULL,
type = "regular")
```

Arguments

`dendat`	n*d matrix of real numbers; the data matrix.
`pcf`	piecewise constant function; output of "pcf.kern", or "pcf.func", for example.
`lst`	level set tree; a list of vectors. The list contains at least vectors "level", "volume", and "parent". For example, function "leafsfirst" computes a level set tree.
`paletti`	a vector of names of colors; for example paletti=c("black","red","blue").
`clusterlevel`	a positive real number; gives the level which is such that the disconnected components of the level set with this level define the clusters.
`nodes`	a vector of positive integers; contains pointers to the nodes of level set tree "lst"; the nodes indicate which disconnected components of level sets define the clusters.
`type`	either "regular" or "adaptive"; if type="regular", then the piecewise constant function has a regular partition; if type="adaptive", then the piecewise constant function has an adaptive partition.

Value

a list with components "datacolo" and "ord". "datacolo" is a vector of color names. This vector assigns a color to each row of matrix "dendat". "ord" is a vector of positive integers. This vector contains pointers to rows of the matrix "dendat". The pointers indicate the order so that the data points with the lowest estimated density are first.

Author(s)

Jussi Klemela

See Also

pcf.kern leafsfirst

Examples

```
# generate data
seed<-1
n<-50
d<-2
l<-3; D<-4; c<-D/sqrt(2)
M<-matrix(0,l,d); M[2,]<-c; M[3,]<--c
sig<-matrix(1,l,d)
p<-rep(1/l,l)
dendat<-sim.data(type="mixt",n=n,M=M,sig=sig,p=p,seed=seed)

# colored volume function
h<-(4/(d+2))^(1/(d+4))*n^(-1/(d+4))*apply(dendat,2,sd)
N<-rep(20,d)
pcf<-pcf.kern(dendat,h,N)
lst<-leafsfirst(pcf)

# colored scatter plot
cd<-colors2data(dendat,pcf,lst)
plot(dendat,col=cd$datacolo)

# colored scatter plot: plot the observations so that the observations with
# the highest density value are plotted last
cd<-colors2data(dendat,pcf,lst)
plot(dendat[cd$ord,1],dendat[cd$ord,2],col=cd$datacolo[cd$ord])
```

---

draw.levset *Plots a level set of a 2D function*

Description

Plots a level set of a piecewise constant 2D function.

Usage

```
draw.levset(pcf, lev=NULL, bary = NULL, propor = 0.1, col=NULL,
bound = NULL, dendat = NULL, xaxt = "s", yaxt = "s", cex.axis = 1)
```

Arguments

pcf	piecewise constant function; output of "pcf.kern" or "pcf.func"
lev	real number; gives the level of the level set
bary	vector of 2 reals; the barycenter; if given will be plotted

propor	0<propor<1; the level set whose level is "propor" times the maximum value of the function will be drawn
col	the color of the lines of the plot; for example "red"
bound	vector of 4 real numbers; c(xmin,xmax,ymin,ymax); gives the values for the parameters "xlim" and "ylim" in the "plot" function; if "bound" is not specified, then the support of function "pcf" will be used
dendat	n*2-matrix or real values; when the "pcf" is a density estimate based on data "dendat", then specifying a value for the parameter "dendat", we can plot the data together with the estimate of a level set
xaxt	a character which specifies the x axis type; either "s" or "n"; see "par"
yaxt	a character which specifies the y axis type; either "s" or "n"; see "par"
cex.axis	the magnification to be used for axis annotation

Value

Makes a plot to the graphics window

Author(s)

Jussi Klemela

See Also

pcf.kern, pcf.func

Examples

```
dendat<-sim.data(n=100,type="mulmod")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))

draw.levset(pcf,lev=1.331979e-02)

draw.levset(pcf,propor=0.4)
```

---

draw.pcf *Prepares the plotting of a 2D or 1D piecewise constant function*

Description

Computes x, y, and z arguments for a piecewise constant function, to be used in "persp" or "contour" functions, or the "x" and "y" arguments to be used in "plot".

Usage

```
draw.pcf(pcf, pnum = rep(32,length(pcf$N)), corona = 5, dens = FALSE,
minval = 0, drawkern=TRUE)
```

Arguments

pcf	piecewise constant function; output of "pcf.kern" or "pcf.func"
pnum	vector of 2 positive integers; dimension of the grid where the function will be plotted; not needed in the 1D case
corona	positive integer; gives the number of zeros around the support of the function; the plots look better when there are a corona of zeros around the function
dens	TRUE for densities, used in the 1D case instead of the corona
minval	real number, gives the level of the basis plane of the picture, one takes "minval" typically to be the minimum value of the function (for densities it would be zero)
drawkern	TRUE or FALSE; internal parameter; when one wants to use function "drawkern" for calculation, then one should specify drawkern=TRUE, this can be used in the case that "pcf" is a kernel density estimate

Value

in the 2D case list of vectors "x" and "y" and matrix "z", in the 1D case list of vectors "x" and "y"

Author(s)

Jussi Klemela

See Also

pcf.kern, pcf.func, persp, contour,

Examples

```
# 2D case
dendat<-sim.data(n=100,type="mulmod")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))

dp<-draw.pcf(pcf,pnum=c(30,30))

contour(dp$x,dp$y,dp$z,drawlabels=FALSE)

persp(dp$x,dp$y,dp$z)

# 1D case
set.seed(1)
dendat<-matrix(rnorm(20),20)   #10*1 data-matrix
pcf<-pcf.kern(dendat,h=1,N=c(25))

dp<-draw.pcf(pcf)

plot(dp$x,dp$y,type="l")
```

etais *Undocumented functions*

Description

Undocumented functions of package denpro which are used by package delt

excmas *Computes the excess masses associated with the nodes of a level set tree*

Description

Given a level set tree, calculates the excess mass for every node of the tree. The excess mass is the probability mass under the curve, over a given level, in a certain branch of the level set tree. For example, function "leafsfirst" returns a level set tree.

Usage

```
excmas(lst)
```

Arguments

lst	level set tree object; list of vectors which is the output of for example "leafsfirst", "profkern", "profhist", or "profgene"

Value

Vector whose length is equal to the number of nodes of the level set tree.

Author(s)

Jussi Klemela

See Also

leafsfirst, plottree

Examples

```
set.seed(1)
dendat<-matrix(rnorm(20),10)   #10*2 data-matrix
pcf<-pcf.kern(dendat,h=1,N=c(16,16))
lst<-leafsfirst(pcf)
excmas(lst)
```

exmap *Computes a scale of excess mass profiles*

Description

Calculates a scale of excess mass profiles from a scale of estimates given as level set trees

Usage

```
exmap(estiseq, mt, hind = NULL, hseq = NULL, n = NULL, moteslist = NULL,
ylist = NULL)
```

Arguments

estiseq	a scale of estimates, output for example of "lstseq.kern"
mt	mode graph; output of "modegraph"
hind	the indexes of the estimates for which the excess mass profiles will be calculated
hseq	the sequence of smoothing parameters corresponding to estimates
n	sample size
moteslist	internal
ylist	internal, alternative to hind

Value

A list of excess mass profiles

Author(s)

Jussi Klemela

See Also

plotexmap, modegraph

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N,lstree=TRUE)

mt<-modegraph(estiseq)
```

```
horilines<-c(1,2,3,4)

sp<-exmap(estiseq,mt,horilines)

plotexmap(sp,mt,ylim=c(1,4.5))
```

---

findbnodes *Finds pointers to the nodes of a level set tree*

Description

Finds pointers to the nodes of a level set tree. The nodes are such that their excess mass is among "modenum" largest excess masses, and they start a new branch.

Usage

```
findbnodes(lst, modenum=1, num=NULL)
```

Arguments

lst	level set tree; a list of vectors. The list contains at least vectors "level", "volume", and "parent". For example, function "leafsfirst" computes a level set tree.
modenum	positive integer; the number of nodes whose pointers are computed
num	a technical parameter

Value

a vector of integers; pointers to the nodes of the level set tree

Author(s)

Jussi Klemela

See Also

leafsfirst

Examples

```
# generate data
seed<-1
n<-50
d<-2
l<-3; D<-4; c<-D/sqrt(2)
M<-matrix(0,l,d); M[2,]<-c; M[3,]<--c
sig<-matrix(1,l,d)
p<-rep(1/l,l)
```

```
dendat<-sim.data(type="mixt",n=n,M=M,sig=sig,p=p,seed=seed)

h<-(4/(d+2))^(1/(d+4))*n^(-1/(d+4))*apply(dendat,2,sd)
N<-rep(20,d)
pcf<-pcf.kern(dendat,h,N)
lst<-leafsfirst(pcf)

nodes<-findbnodes(lst,modenum=3)
```

`hgrid` *Returns a grid of smoothing parameter values*

Description

Returns a grid of smoothing parameter values for kernel estimates, either with a logarithmic spacing, or with equal spacing, in decreasing order.

Usage

```
hgrid(h1, h2, lkm, base = 10)
```

Arguments

`h1`	the lowest smoothing parameter value
`h2`	the largest smoothing parameter value
`lkm`	the number of smoothing parameters in the grid
`base`	the base of the logarithm, used in the logarithmic spacing, if NULL, then the equal spacing is applied

Value

a vector of smoothing parameters

Author(s)

Jussi Klemela

See Also

`lstseq.kern`, `branchmap`, `modegraph`, `exmap`

Examples

```
h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

hseq<-hgrid(h1,h2,lkm,base=NULL)
```

`leafsfirst` *Computes a level set tree, shape tree, or tail tree*

Description

Computes a level set tree or shape tree from a piecewise constant function, or calculates a tail tree from a data set.

Usage

```
leafsfirst(pcf = NULL, lev = NULL, refe = NULL, type = "lst",
levmet = "radius", ordmet = "etaisrec", ngrid = NULL, dendat = NULL, rho = 0,
propor= NULL, lowest = "dens", f=NULL)
```

Arguments

`pcf`	piecewise constant function; output of "pcf.kern", or "pcf.func", for example
`lev`	positive real number; a level of the level set whose shape tree will be calculated
`refe`	vector of d real numbers; the reference point of the shape tree
`type`	"lst" or "shape"; not needed
`levmet`	"radius" or "proba"; radius plot or probability content plot
`ordmet`	"etaisrec" or "baryrec"; a distance to a rectangle is defined to be the distance to the boundary or the distance to the barycenter; concerns shape trees
`ngrid`	a positive integer; the tree will be pruned to contain "ngrid" levels
`dendat`	n*d data matrix; gives the data set when a tail tree will be calculated
`rho`	positive real number; the resolution threshold for the tail tree
`propor`	0<propor<1; a shape tree of the level set whose level is "propor" times the maximum value of the function will be calculated
`lowest`	a character; if lowest="dens", then it is assumed that the function whose level set tree we calculate is nonnegative
`f`	a vector of estimated values of the density at the data points; this is used in the case of tail trees to estimate the volumes of the regions

Value

a level set tree, shape tree, or tail tree

Author(s)

Jussi Klemela

See Also

pcf.kern, treedisc, plotvolu, plotbary,

Examples

```
dendat<-sim.data(n=100,type="mulmod")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))

# level set tree

lst<-leafsfirst(pcf)

td<-treedisc(lst,pcf,ngrid=30)
plotvolu(td)

# shape tree

st<-leafsfirst(pcf,propo=0.1)   # 10% level set

tdst<-treedisc(st,pcf,ngrid=30)
plotvolu(tdst)

# tail tree

rho<-0.65
tt<-leafsfirst(dendat=dendat,rho=rho)

plotbary(tt)
```

---

leafsfirst.adagrid *Computes a level set tree of a discretized kernel estimator with an adaptive grid*

Description

Computes a level set tree of a discretized kernel estimator with an adaptive grid .

Usage

```
leafsfirst.adagrid(pcf)
```

Arguments

pcf	piecewise constant function; output of "pcf.greedy.kernel"

Value

a level set tree

Note

Function pcf.greedy.kernel is in package "delt"

Author(s)

Jussi Klemela

References

http://jussiklemela.com/Art/volume

See Also

leafsfirst

Examples

```
# generate data
seed<-1
n<-50
d<-2
l<-3; D<-4; c<-D/sqrt(2)
M<-matrix(0,l,d); M[2,]<-c; M[3,]<--c
sig<-matrix(1,l,d)
p<-rep(1/l,l)
dendat<-sim.data(type="mixt",n=n,M=M,sig=sig,p=p,seed=seed)

h<-(4/(d+2))^(1/(d+4))*n^(-1/(d+4))*apply(dendat,2,sd)
minobs<-3
#pcf<-pcf.greedy.kernel(dendat,h,minobs=minobs,type="greedy")
#lst<-leafsfirst.adagrid(pcf)
```

---

locofmax *Calculates the location of the maximum of a function*

Description

Calculates the location of the maximum of a piecewise constant function.

Usage

```
locofmax(pcf)
```

Arguments

pcf	piecewise constant function; output of "pcf.kern" or "pcf.func"

Value

a d-vector; the location of the maximum; this is defined as the barycenter of the set where the function has the maximum

Author(s)

Jussi Klemela

See Also

pcf.kern, pcf.func

Examples

```
dendat<-sim.data(n=100,type="mulmod")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))

locofmax(pcf)
```

---

lstseq.kern *Calculates a scale of kernel estimates*

Description

Calculates a scale of kernel estimates corresponding to a scale of smoothing parmeters.

Usage

```
lstseq.kern(dendat, hseq, N, lstree = NULL, level = NULL,
Q = NULL, kernel = "gauss", hw = NULL, algo = "leafsfirst", support = NULL)
```

Arguments

dendat	n*d matrix of real numbers; the data matrix
hseq	a vector of positive real numbers; the sequence should be monotonic
N	vector of d positive integers; the dimension of the grid where the kernel estimate will be evaluated; we evaluate the estimate on a regular grid which contains the support of the kernel estimate
lstree	if NULL, then level set trees are not calculated
level	NULL or a real number between 0 and 1; if NULL, then shape trees are not calculated; if number, then it is the level in percents of the maximum of the level sets for which the shape trees are calculated

Q	positive integer; needed only in the DynaDecompose algorithm, see parameter "algo"; the number of levels in the level set trees
kernel	"epane" or "gauss"; the kernel is either the Bartlett-Epanechnikov product kernel or the standard Gaussian
hw	positive integer; parameter for time localized kernel estimation; gives the smoothing parameter for the temporal smoothing
algo	"leafsfirst" or "dynadecompose"
support	2*d vector of reals gives the d intervals of a rectangular support; c(low1,upp1,...,lowd,uppd)

Value

A list with components

lstseq	a list of level set trees
pcfseq	a list of piecewise constant functions
stseq	a list of shape trees
hseq	a vector of smoothing parameters corresponding to the members in the sequences

Author(s)

Jussi Klemela

See Also

scaletable

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N)
```

modecent *Returns locations of modes of a density estimate*

Description

With "modecent" we may find the location of the modes of a function. Argument to function "modecent" is the level set tree of the function. A mode is defined here to be the barycenter of a set at which the estimate has a local maximum.

Usage

```
modecent(lst)
```

Arguments

lst	level set tree object; list containing lst$parent and lst$center; output of "leafsfirst", "profgene", "profhist", or "profkern"

Value

modenum*d-matrix; for each mode its location. The rows of output are d-vectors specifying the locations of the modes.

Author(s)

Jussi Klemela

See Also

leafsfirst, profkern, plotbary

Examples

```
set.seed(1)
dendat<-matrix(rnorm(20),10)   #10*2 data-matrix
pcf<-pcf.kern(dendat,h=2,N=c(16,16))
lst<-leafsfirst(pcf)
modecent(lst)
```

modegraph *Calculates a mode graph from a scale of estimates*

Description

Calculates a mode graph from a scale of estimates. The estimates in the scale are given as level set trees.

Usage

```
modegraph(estiseq, hseq = NULL, paletti = NULL)
```

Arguments

estiseq	a scale of estimates, output for example of "lstseq.kern"
hseq	the corresponding scale of smoothing parameters
paletti	vector of color names

Value

mode graph

Author(s)

Jussi Klemela

See Also

lstseq.kern, plotmodet

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N,lstree=TRUE)

mt<-modegraph(estiseq)

plotmodet(mt)
```

paracoor *Makes a parallel coordinate plot*

Description

Makes a parallel coordinate plot of a data matrix: the coordinates of the observations are plotted so that the coordinate axes are drawn parallel to each other and the coordinates of a single observation are joined with lines.

Usage

```
paracoor(X, Y=NULL, xmargin = 0.1,
paletti = matrix("black",dim(X)[1],1),
noadd = TRUE, verti = NULL, cex.axis = 1,
points=TRUE, col.verti="black", col.verti.y="red", digits=3,
arg=NULL, colarg="red", lwd=1, cex=1, yaxt="s")
```

Arguments

X	n*d data matrix of real numbers
Y	n*1 data matrix of real numbers; response variable
xmargin	positive real number; empty space in left and right
paletti	n vector of colors; the lines will be colored with these colors
noadd	TRUE or FALSE; if FALSE, then the lines will be added to an existing plot
verti	real number or NULL; gives the x coordinate of a vertical line; can be used to divide the plot vertically to two parts
cex.axis	the magnification to be used for axis annotation
points	TRUE or FALSE; whether points are plotted and not only lines
col.verti	a caharacter; the name of a color
col.verti.y	a caharacter; the name of a color
digits	a natural number; the number of digits in the annotation
arg	d*1 vector; the current value of the explanatory variable
colarg	a character; the name of a color
lwd	a positive real number; the width of the lines
cex	a positive real number; the largness of the circles
yaxt	a character which specifies the y axis type

Value

Makes a plot on the graphics window

Author(s)

Jussi Klemela

References

Inselberg (1985), Wegman (1990)

Examples

```
X<-sim.data(n=100,type="mulmod")
paracoor(X)

set.seed(1)
X<-matrix(rnorm(300),100,3)
paracoor(X)

set.seed(1)
X<-matrix(rnorm(400),100,4)
paracoor(X)
```

`pcf.func` *Calculates a piecewise constant function for some illustrative purposes*

Description

Calculates a piecewise constant function of a given functional form: Gaussian or a mixture of Gaussians, or functions with a given copula, ...

Usage

```
pcf.func(func, N,
sig = rep(1, length(N)), support = NULL, theta = NULL,
g=1, M = NULL, p = NULL, mul = 3, t = NULL,
marginal = "normal", r = 0,
mu = NULL, xi = NULL, Omega = NULL, alpha = NULL, df = NULL,
a = 0.5, b = 0.5, distr = FALSE, std = 1, lowest = 0)
```

Arguments

`func`	a character string; the possibilities are "mixt", "normal","student","gumbel","frank","clayton", "skewgauss", "prod", "epan", "hat"
`N`	vector of d positive integers; the dimension of the grid where the function will be evaluated; we evaluate the function on a regular grid which contains the support of the function

sig	mixnum*d matrix of positive real numbers; the standard deviations of the marginals in a mixture of Gaussians, or the scaling factors of the student and uniform copulas
support	2*d vector of reals gives the d intervals of a rectangular support
theta	rotation angle
g	parameter of the Archimedean copulas
M	mixnum*d matrix of positive real numbers; the means of the components in a mixture of Gaussians
p	mixnum-vector of probabilities; mixture weights
mul	internal parameter
t	parameter of the Student marginals (degrees of freedom)
marginal	marginal distributions for the copulas; "normal", "student", or "unif"
r	0<r<1; correlation for the Gaussian and Student copulas
mu	parameter
xi	parameter
Omega	parameter
alpha	skeweness parameter for the skewed Gaussian density
df	degrees of freedom for the Student copula
a, b	parameters of the hat function
distr	TRUE if the distribution function is to be drawn instead of the density
std	standard deviation in the 1D case
lowest	real value; for densities equal to 0, otherwise is equal to the minimum of the function

Value

a piecewise constant function object, see the web page

Author(s)

Jussi Klemela

References

http://www.rni.helsinki.fi/~jsk/denpro/

See Also

draw.pcf

Examples

```
# Elliptical copulas

N<-c(32,32)  # choose the grid size
copula<-c("gauss","student")
margin<-c("normal","student","unif")
b<-4
support<-c(-b,b,-b,b)

ci<-1        # copula, ci = 1, 2
r<-0.5       # parameter of the copula
df<-2        # degrees of freedom for the Student copula

mi<-1        # margin, mi = 1, 2, 3
sig<-c(1,1)  # std:s for the margins
t<-c(2,2)    # degreeds of freedom for the student margin

ef<-pcf.func(copula[ci],N,marginal=margin[mi],support=support,
             r=r,df=df,sig=sig,t=t)

dp<-draw.pcf(ef)
contour(dp$x,dp$y,dp$z)

persp(dp$x,dp$y,dp$z,theta=30,phi=30)

# Archimedean copulas

N<-c(32,32)  # choose the grid size
copula<-c("gumbel","frank","clayton")
margin<-c("normal","student","unif")
b<-4
support<-c(-b,b,-b,b)

ci<-1        # copula, ci = 1, 2, 3
g<-2         # parameter of the copula

mi<-1        # margin, mi = 1, 2, 3
sig<-c(1,1)  # std:s for the margins
t<-c(2,2)    # degreeds of freedom for the student margin

ef<-pcf.func(copula[ci],N,marginal=margin[mi],support=support,
             g=g,sig=sig,t=t)

dp<-draw.pcf(ef)
contour(dp$x,dp$y,dp$z)

persp(dp$x,dp$y,dp$z,theta=30,phi=30)

# mixture of Gaussians

d<-2
mixnum<-3              #we simulate a mixture of three Gaussians
```

```
M<-matrix(0,mixnum,d)   #rows of M contain the means of members of the mixture
M[1,]<-c(0,0)
M[2,]<-c(4,0)
M[3,]<-c(0,4)
sig<-matrix(1,mixnum,d) #rows of sig contain the std:s ot the marginals
p0<-1/mixnum
p<-p0*rep(1,mixnum)     #p is a vector of weights for the mixture members
N<-c(50,50)

pcf<-pcf.func("mixt",N,sig=sig,M=M,p=p)

dp<-draw.pcf(pcf,pnum=c(30,30))
contour(dp$x,dp$y,dp$z,drawlabels=FALSE)
persp(dp$x,dp$y,dp$z)

# skewed Gaussian

func<-"skewgauss"
N<-c(50,50)
support<-c(-6,2,-6,2)
mu<-c(0,0)
sig<-c(3,1)
alpha<-c(6,0)
theta<--3*pi/4

pcf<-pcf.func(func,N,support=support,mu=mu,sig=sig,
               alpha=alpha,theta=theta)

dp<-draw.pcf(pcf,pnum=c(60,60))

contour(dp$x,dp$y,dp$z)

persp(dp$x,dp$y,dp$z,theta=30,phi=30)

# product of univariate Student densities

func<-"prod"
marginal<-"student"
b<-5; support<-c(-b,b,-b,b)
N<-c(32,32)                 # choose the grid size
t<-0.5                      # degrees of freedom
ef<-pcf.func(func,N,t=t,support=support,marginal=marginal)

dp<-draw.pcf(ef)
contour(dp$x,dp$y,dp$z)

persp(dp$x,dp$y,dp$z,theta=30,phi=30)

# Bartlett-Epanechnikov

func<-"epan"
N<-c(50,50)
sig<-c(1,1)
```

```
ef<-pcf.func(func,N,sig)

dp<-draw.pcf(ef)
contour(dp$x,dp$y,dp$z)

persp(dp$x,dp$y,dp$z,theta=30,phi=30)
```

---

pcf.kern *Computes a multivariate kernel estimate*

Description

Computes a multivariate kernel estimate and gives the output as a piecewise constant function object.

Usage

```
pcf.kern(dendat, h, N, kernel = "gauss", weights = NULL, support = NULL,
lowest = 0, radi = 0)
```

Arguments

dendat	n*d matrix of real numbers; the data matrix
h	d vector of positive real numbers; vector of smoothing parameters
N	vector of d positive dyadic integers; the dimension of the grid where the kernel estimate will be evaluated; we evaluate the estimate on a regular grid which contains the support of the kernel estimate
kernel	"gauss", "epane", "bart", or "uniform"; the kernel is either the standard Gaussian, Epanechnikov product kernel, Bartlett kernel, or uniform kernel
weights	n vector of nonnegative weights, where n is the sample size; sum of the elements of "weights" should be one; these are the weights of a time localized kernel estimator
support	2*d vector of reals gives the d intervals of a rectangular support in the form c(low1,upp1,...,lowd,uppd)
lowest	a real value; the density estimate will be truncated to take value zero, if the value of the estimate is less or equal to "lowest"
radi	a nonnegative real number; the support is estimated as the smallest rectangle containing the observations with an additional band whose width is equal to "radi"

Value

a piecewise constant function object, see the web site

Author(s)

Jussi Klemela

References

http://www.rni.helsinki.fi/~jsk/denpro/

See Also

draw.pcf lstseq.kern

Examples

```
n<-100
dendat<-sim.data(n=n,type="mulmod")

h<-1
pcf<-pcf.kern(dendat,h=h,N=c(32,32))
dp<-draw.pcf(pcf)
contour(dp$x,dp$y,dp$z,drawlabels=FALSE)

d<-2
h<-(4/(d+2))^(1/(d+4))*n^(-1/(d+4))*apply(dendat,2,sd)
pcf<-pcf.kern(dendat,h=h,N=c(32,32))
dp<-draw.pcf(pcf)
contour(dp$x,dp$y,dp$z,drawlabels=FALSE)

# we use now nonuniform weighting of kernels

weights<-matrix(0,n,1)
threshold<-4
for (i in 1:n){
    eta<-(n-i)
    if (eta/h>threshold) result<-0 else result<-exp(-eta^2/(2*h^2))
    weights[i]<-result
}
weights<-weights/sum(weights)

pcf<-pcf.kern(dendat,h=1,N=c(32,32),weights=weights)

dp<-draw.pcf(pcf)
contour(dp$x,dp$y,dp$z,drawlabels=FALSE)
```

---

plotbary *Makes a barycenter plot of a level set tree, a location plot of a shape tree, or a tail tree plot of a tail tree*

Description

Plots a barycenter plot of a level set tree, or a location plot of a shape tree (of the given direction), or a tail tree plot of tail tree (of the given direction). A barycenter plot visualizes the barycenters of the separated regions of the level sets of the function. The barycenter of a set is the "center of mass" of this set. A location plot visualizes the barycenters of the tail regions of a set. A tail tree plot makes a parallel level plot of the data, and shows the tree structure of the tail tree in the plot.

Usage

```
plotbary(lst, coordi=1,
plot=TRUE, data=FALSE, crit=NULL,orderrule="distcenter",
modelabel=FALSE, ptext=0, leimat=NULL, symbo=NULL,
info=NULL, infolift=0, infopos=0,
xmarginleft=0, xmarginright=0, ymargin=0,
xlim=NULL, ylim=NULL, xaxt="s", yaxt="s",
nodesymbo=20, col=NULL, col.axis="black", collines=NULL, paletti=NULL,
shift=0, shiftindex=NULL,
modlabret=FALSE, modecolo=NULL, modepointer=NULL, colometh = "lst",
colothre = min(lst$level), lines=TRUE, wedge=FALSE, lty.wedge=2, title=TRUE,
titletext="coordinate", cex=NULL, nodemag=NULL, cex.sub=1, cex.axis = 1,
newtitle = FALSE, cex.lab = 1, lowest="dens", subtitle=NULL)
```

Arguments

lst	level set tree or shape tree; list of vectors. The list contains at least vectors "level", "volume", and "parent". For example, function "leafsfirst" gives a level set tree or a shape tree as an output. Functions "profkern" and "profhist" give a level set tree as an output.
coordi	integer 1,...,d, when the function is d-dimensional; gives the coordinate direction with respect which the plot will be made.
plot	T or F; TRUE if we make a plot, otherwise FALSE.
data	T or F; TRUE if we want the output to contain some information, for example the ordering for siblings. This option is needed only by other plotting functions of the package, it is not needed by the end user.
crit	d-vector of real numbers; gives a way to control an ordering of siblings. The leftmost sibling is the one whose barycenter is furthest away from vector "crit", in the Euclidean metric.
orderrule	lower level parameter
modelabel	T or F; TRUE if the modes will be labelled. The default is to use labels M1, M2,...
ptext	non-negative real number; amount by which the mode labels will be lifted.
leimat	vector of characters; the length of the vector should be equal to the number of modes of the estimate. This option is for the case we do not want the ordering of the labels to be done automatically.

symbo	character; for example "L". The default value for the automatic labelling of the modes is to use M1, M2,... With "symbo" we may switch to L1, L2,..., for example.
info	vector of numbers or characters, whose length is equal to the number of nodes of the level set tree. The elements of "info" will be placed on the right side of the nodes. For example "info" may be generated by "excmas" or we may define "info" to contain the frequencies of the nodes. (Frequencies may be obtained directly from the function "profhist")
infolift	real number; controls the vertical positioning of the elements of "info".
infopos	real number; controls the horizontal positioning of the elements of "info". Negative "infopos" will move elements of "info" to the right hand side.
xmarginleft	non-negative real number; adds more margin on the left hand side. The box around the plot will be moved to the left with the amount "xmarginleft".
xmarginright	non-negative real number; adds more margin on the right hand side. The box around the plot will be moved to the right with the amount "xmarginright".
ymargin	nonnegative real number; adds more margin on the top of the plot. The box around plot will be moved up with the amount "ymargin".
xlim	vector of 2 real numbers; gives the limits for the scale of x-axis.
ylim	vector of 2 real numbers; gives the limits for the scale of y-axis.
xaxt	a character which specifies the x axis type; either "s" or "n"; see "par"
yaxt	a character which specifies the y axis type; either "s" or "n"; see "par"
nodesymbo	symbol for the nodes of the tree; integer 19-25; see help(points) for the definitions
col	colour for the nodes; for example "black" or "blue".
col.axis	colour for the x and y-axis; for example "black" or "blue".
collines	colour for lines joining nodes; for example "black" or "blue".
paletti	a character vector of color names, in the order they will be used to color the nodes and the lines connecting the nodes; coloring starts from the leafs nodes and the color is changed always when two branches are joining
shift	a real number; the amount to shift a label of a leaf node; used to enhance the plot, see the parameter "shiftindex"
shiftindex	the index to be shifted, see the parameter "shift"
modlabret	T or F; T if the locations of the modes will be returned
modecolo	lowel level parameter
modepointer	lowel level parameter
colometh	if colometh=="lst", then colors are first chosen for the leafs, otherwise the colors are first chosen for the roots; for the level set trees it is better to choose "lst", but for dendrograms it is better to color the roots first
colothre	positive real number; fixes the colors at the level given by "colothre", this option is used for dendrograms
lines	TRUE or FALSE; used in a tail tree plot; FALSE suppresses the parent-child lines and one can avoid over plotting

wedge	TRUE or FALSE; used in a tail tree plot if TRUE, then the outer limts for the points will be plotted; these outer limits have a wedge form
lty.wedge	"lty" parameter (line type) for the wedge lines
title	if TRUE, then the subtitle "coordinate i" is printed when the coordinate is i and the the titletext is "coordinate"
titletext	character string; the text in the title will be titletext i, if the barycenter plot is for the i:th coordinate
cex	magnification factor for the symbols; see "par"
nodemag	magnification factor for the the points
cex.sub	magnification factor for the subtitle; see "par"
cex.axis	the magnification to be used for axis annotation; see "par"
newtitle	FALSE or TRUE; if TITLE=FALSE, we can put newtitle=TRUE to print the title as an x-axis annotation, and not as a subtitle
cex.lab	the magnification to be used for x and y labels; see "par"
lowest	"dens" or "regr"; if lowest="dens", then it is assumed that the underlying function is a density function; if lowest="regr", then it is assumed that the underlying function is an arbitrary multivariate real function
subtitle	NULL or a character string; gives the subtitle of the plot

Value

By default a plot is made on the graphics window. If modlabret=T, then the locations of the modes are returned, with the labeling of the modes corresponding to the labeling in the plot.

Author(s)

Jussi Klemela

See Also

leafsfirst, treedisc, prunemodes, plotvolu, plottree

Examples

```
# level set tree

dendat<-sim.data(n=100,type="mulmod")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))
lst<-leafsfirst(pcf)
td<-treedisc(lst,pcf,ngrid=30)

plotbary(td)

plotbary(td,coordi=2,ptext=0.002,symbo="L",modlabret=TRUE)

# shape tree
```

```
dendat<-sim.data(n=100,type="cross")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))
st<-leafsfirst(pcf,propor=0.01)
td<-treedisc(st,pcf,ngrid=60)

plotbary(td)

# tail tree

tt<-leafsfirst(dendat=dendat,rho=0.65)
plotbary(tt)
```

`plotbranchmap` *Plots a branching map*

Description

Plots a branching map to the graphics window.

Usage

```
plotbranchmap(bm, phi = 55, theta = 30)
```

Arguments

`bm`	A branching map; output of function "branchmap".
`phi`	The phi-parameter of function "persp".
`theta`	The theta-parameter of function "persp".

Details

Function "plotbranchmap" just calls function "persp".

Value

Makes a plot to the current graphics window.

Author(s)

Jussi Klemela

References

http://denstruct.net

See Also

`branchmap`

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N,lstree=TRUE)

bm<-branchmap(estiseq)

plotbranchmap(bm)

## The function is currently defined as
function(bm,phi=55,theta=30)
{

persp(x=bm$level,y=bm$h,z=bm$z,
xlab="level",ylab="h",zlab="excess mass",
ticktype="detailed",
col=bm$col,phi=phi,theta=theta)

  }
```

`plotexmap` *Plots a scale of excess mass profiles*

Description

Plots a scale of excess mass profiles.

Usage

```
plotexmap(sp, mt, xaxt = "n", lift = 0.1, leaflift = 0.1, ylim = NULL,
leafcolors = NULL)
```

Arguments

`sp`	scale of excess mass profiles; output of "exmap"
`mt`	mode graph; output of "modegraph"
`xaxt`	"n" or "y"; whether x-axis will be plotted
`lift`	positive real number; space between the lines of the excess mass profiles
`leaflift`	positive real number; lifting for the bullets associated with the leaf nodes of the excess mass profiles
`ylim`	2-vector; gives the interval of the y-axis
`leafcolors`	a paletti of the colors for the labeling of the leaf nodes

Value

Plots the scale of excess mass profiles to the graphic device

Author(s)

Jussi Klemela

See Also

exmap, modegraph

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N,lstree=TRUE)

mt<-modegraph(estiseq)

horilines<-c(1,2,3,4)

sp<-exmap(estiseq,mt,horilines)

plotexmap(sp,mt,ylim=c(1,4.5))
```

---

plotmodet *Plots a mode graph*

Description

Plots one window of a mode graph.

Usage

```
plotmodet(mt, coordi = 1, colot = NULL, shift = 0, xlim = NULL,
xlab = "", ylab = "", horilines = NULL, symbo = 20, loga = NULL,
lty = "dashed", cex.axis = 1, title = TRUE, cex.sub = 1, cex.lab = 1,
xaxt = "s", yaxt = "s")
```

Arguments

mt	mode graph; output of "modegraph"
coordi	integer 1,...,d
colot	colors
shift	real number; to shift the parent-child lines in the case of overlays
xlim	2-vector of reals, giving the interval of x-axis
xlab	labeling for the x-axis
ylab	labeling for y-axis
horilines	a vector of indexes; horizonta lines will be plotted
symbo	labeling for modes; "L" or "N"
loga	"y" or "n"; whether logarithmic y-axis will be used
lty	line type
cex.axis	magnification factor for the axis annotation; see "par"
title	TRUE or FALSE; makes a subtitle which shows the value of "coordi"
cex.sub	magnification factor for the subtitle; see "par"
cex.lab	magnification factor for the axis labels (names of x and y); see "par"
xaxt	a character which specifies the x axis type; either "s" or "n"; see "par"
yaxt	a character which specifies the y axis type; either "s" or "n"; see "par"

Value

Plots to the graphics device

Author(s)

Jussi Klemela

See Also

modegraph, lstseq.kern,

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N,lstree=TRUE)

mt<-modegraph(estiseq)
```

```
plotmodet(mt)

plotmodet(mt,coordi=2)
```

---

plottree *Makes a tree plot of a level set tree, of a shape tree, or of a tail tree*

Description

Plots a tree plot of a level set tree, of a shape tree, or of a tail tree to the graphics window.

Usage

```
plottree(lst,
plot=T, data=F, crit=NULL, orderrule="distcenter",
modelabel=TRUE, ptext=0, leimat=NULL, symbo=NULL,
info=NULL, infolift=0, infopos=0, infochar=NULL,
xmarginleft=0, xmarginright=0, ymargin=0,
xlim=NULL, ylim=NULL, col="black", col.axis="black",
linecol=rep("black",length(lst$parent)),
pch=21, dimen=NULL, yaxt="s", axes=T,
cex=NULL, nodemag=NULL, linemag=1,
cex.axis = 1, ylab = "", cex.lab = 1, colo = FALSE, paletti = NULL,
lowest = "dens")
```

Arguments

lst	level set tree; list of vectors. The list contains at least vectors "level", "volume", and "parent". For example, functions "leafsfirst", "profkern" and "profhist" give a level set tree as an output.
plot	T or F; TRUE if we make a plot, otherwise FALSE.
data	T or F; TRUE if we want output to contain some information, for example an ordering for siblings. This option is needed only by other plotting functions of the package, it is not needed by the end user.
crit	d-vector of real numbers; gives a way to control ordering of siblings. The left-most sibling is the one whose barycenter is furthest away from vector "crit", in the Euclidean distance.
orderrule	lower level paremeter
modelabel	T or F; TRUE if the modes will be labelled. The default is to use labels M1, M2,...
ptext	non-negative real number; amount by which the mode labels will be lifted.
leimat	vector of characters; length of the vector should be equal to the number of modes of the estimate. This option is for the case we do not want the ordering of the labels to be done automatically.

symbo	character; for example "L". The default value for the automatic labelling of the modes is to use M1, M2,... With "symbo" we may switch to L1, L2,..., for example.
info	vector of numbers or characters, whose length is equal to the number of nodes of the level set tree. The elements of "info" will be placed on the right side of the nodes. For example "info" may be generated by "excmas" or we may define "info" to contain the frequencies of the nodes. (Frequencies may be obtained directly from the function "profhist".)
infolift	real number; controls the vertical positioning of the elements of "info".
infopos	real number; controls the horizontal positioning of the elements of "info". Negative "infopos" will move elements of "info" to the right hand side.
infochar	vector of characters whose length is equal to the number of nodes of the level set tree; we may annotate nodes with some infromation
xmarginleft	nonnegative real number; adds more margin on the left hand side. The box around the plot will be moved to the left with the amount "xmarginleft".
xmarginright	nonnegative real number; adds more margin on the right hand side. The box around the plot will be moved to the right with the amount "xmarginright".
ymargin	nonnegative real number; adds more margin on the top of the plot. The box around the plot will be moved up with the amount "ymargin".
xlim	vector of 2 real numbers; gives the limits for the scale of x-axis.
ylim	vector of 2 real numbers; gives the limits for the scale of y-axis.
col	colour for the nodes; for example "black" or "blue".
col.axis	colour for the x and y-axis; for example "black" or "blue".
linecol	colour for the lines joining nodes; for example "black" or "blue".
pch	symbol for the nodes of the tree; integer 19-25; see help(points) for the definitions
dimen	positive integer; number of dimensions of the estimate. If "dimen" is not NULL, then we plot without using the usual ordering of the siblings. This option is used when we do not have barycenters available.
yaxt	axis type, see "par"
axes	T or F
cex	magnification factor for the symbols; see "par"
nodemag	magnification factor for the points
linemag	magnification factor for the lines joining the points
cex.axis	magnification factor for the axis annotation; see "par"
ylab	character; y-axis label (name of the y-variable); for example in a tree plot of a level set tree this could be "level"
cex.lab	magnification factor for the axis labels (names of x and y); see "par"
colo	TRUE or FALSE; if TRUE, nodes and lines joining the nodes will be colored
paletti	a vector of names of colors; these colors will be used to color the nodes and the lines joining the nodes
lowest	character string; if lowest="dens", then y-axis starts at 0

Value

By default a plot is made on the graphics window.

Author(s)

Jussi Klemela

See Also

plotvolu, plotbary, leafsfirst

Examples

```
dendat<-sim.data(n=50,type="mulmod")

pcf<-pcf.kern(dendat,h=1,N=c(32,32))
lst<-leafsfirst(pcf)
td<-treedisc(lst,pcf,ngrid=30)
plottree(td)
```

---

plotvolu *Makes a volume plot of a level set tree, a shape plot of a shape tree, or a tail frequency plot of a tail tree*

Description

Plots a volume plot of a level set tree to the graphics window, or a shape plot of a shape tree, or a tail frequency plot of a tail tree.

Usage

```
plotvolu(lst, length=NULL,
toplot=TRUE, data=FALSE, crit=NULL, orderrule="distcenter",
modelabel=FALSE, ptext=0, leimat=NULL, symbo=NULL,
info=NULL, infolift=0, infopos=0,
xmarginleft=0, xmarginright=0, ymargin=0,
xlim=NULL, ylim=NULL,
col="black", col.axis="black",
cutlev=NULL, xaxt="s", yaxt="s",
exmavisu=NULL, bg="transparent", tyyppi="n", lty="solid",colo=FALSE,
lowest="dens", proba=FALSE,
paletti=NULL, cex=NULL, modecolo = NULL, modepointer = NULL, upper = TRUE,
cex.axis = 1, xlab="", ylab="", cex.lab=1, colothre=NULL, nodes=NULL)
```

Arguments

lst	level set tree or shape tree; list of vectors. The list contain at least vectors "level", "volume", and "parent". For example, the output of "leafsfirst", "profkern" or "profhist".
length	vector of positive real numbers; the length of "length" is equal to the number of nodes of the level set tree "lst"; the nodes will be drawn as lines whose length is proportional to the corresponding value in vector "length". If length=NULL, then the length of the lines will be proportional to the values given in vector "volume" in the list "lst"; these are the volumes of the sets associated with nodes of the level set tree. We may for example apply function "excmas" which calculates the excess masses of the nodes and give this vector of excess masses as an argument: length=excmas(lst).
toplot	T or F; TRUE if we make a plot, otherwise FALSE.
data	T or F; TRUE if we want the output to contain some information, for example an ordering for siblings. This option is needed only by other plotting functions of the package, it is not needed by the end user.
crit	d-vector of real numbers; gives a way to control the ordering of siblings. The leftmost sibling is the one whose barycenter is furthest away from vector "crit".
orderrule	lowel level parameter
modelabel	T or F; TRUE if the modes will be labelled. The default is to use labels M1, M2,...
ptext	non-negative real number; the amount by which the mode labels will be lifted.
leimat	vector of characters; the length of the vector should be equal to the number of modes of the estimate. This option is for the case we do not want the ordering of the labels to be done automatically.
symbo	character; for example "L". The default value for the automatic labelling of the modes is to use M1, M2,... With "symbo" we may switch to L1, L2,..., for example.
info	vector of numbers or characters; the length of the vector is equal to the number of nodes of the level set tree. The elements of "info" will be placed on the right side of the nodes. For example, "info" may be generated by "excmas", or we may define "info" to be frequencies of the nodes. (Frequencies may be obtained directly from the function "profhist".)
infolift	real number; controls the vertical positioning of the elements of "info".
infopos	real number; controls the horizontal positioning of the elements of "info". Negative "infopos" will move elements of "info" to the right hand side.
xmarginleft	nonnegative real number; adds more margin on the left hand side. The box around the plot will be moved to the left with the amount "xmarginleft".
xmarginright	nonnegative real number; adds more margin on the right hand side. The box around the plot will be moved to the right with the amount "xmarginright".
ymargin	nonnegative real number; adds more margin on the top of the plot. The box around the plot will be moved up with the amount "ymargin".
xlim	vector of 2 real numbers; gives the limits for the scale of x-axis.

ylim	vector of 2 real numbers; gives the limits for the scale of y-axis.
col	colour for the nodes; for example "black" or "blue".
col.axis	color for the x and y-axis; for example "black" or "blue".
cutlev	positive real number; we may cut the volume plot at "cutlev" and thus zoom to the upper levels.
xaxt	a character which specifies the x axis type; either "s" or "n"; see "par"
yaxt	a character which specifies the y axis type; either "s" or "n"; see "par"
exmavisu	lower level parameter
bg	lower level parameter
tyyppi	lower level parameter
lty	lower level paremeter
colo	F or T; if T, then the regions under the curve of the volume transform/radius transform will be colored with the same color as the corresponding node in the barycenter plot/location plot
lowest	if lowest="dens" then the lowest value of y-axis is equal to 0, else the lowest value of y-axis is equal to the minimum of the function; this option allows to plot functions with negative values
proba	TRUE, if one wants to plot a tail probability plot of a shape tree
paletti	the colors for the segments of the graph; the colors are used to indicate the correspondence with the barycenter plot and thus the paletti should be the same
cex	the magnification factor for the symbols; see "par"
modecolo	internal; used by "scaletable" to set the colors of the leaf nodes
modepointer	internal; used by "scaletable" to find the modes
upper	TRUE or FALSE; if upper=TRUE, then a volume plot of an upper level set tree is drawn, otherwise a volume plot of a lower level set tree is drawn
cex.axis	magnification factor for the axis annotation; see "par"
xlab	character string; label of the x-axis
ylab	character string; label of the y-axis
cex.lab	positive real number; indicates the largeness of the labels of the plot
colothre	NULL or a postitive real number; used to color the volume plot when it is used to visualize a clustering
nodes	a vector of positive integers; used to color the volume plot when it is used to visualize a clustering with locally changing levels

Value

A plot is made on the graphics window.

Author(s)

Jussi Klemela

See Also

leafsfirst, treedisc, prunemodes, plotbary plottree,

Examples

```
# level set tree

dendat<-sim.data(n=100,type="mulmod")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))
lst<-leafsfirst(pcf)
td<-treedisc(lst,pcf,ngrid=30)

plotvolu(td)

plotvolu(td,ptext=0.002,modelabel=TRUE,symbo="L",colo=TRUE)

# shape tree

dendat<-sim.data(n=100,type="cross")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))
st<-leafsfirst(pcf,propor=0.01)
td<-treedisc(st,pcf,ngrid=60)

plotvolu(td)

plotvolu(td,proba=TRUE)

# tail tree

tt<-leafsfirst(dendat=dendat,rho=0.65)
plotvolu(tt)
```

`plotvolu2d` *Makes a perspective plot of a 2D volume function or a 2D probability content function*

Description

Plots the output of function "shape2d"; this is a 2D volume function or a 2D probability content function.

Usage

```
plotvolu2d(vd, theta = NULL, phi = NULL, typer = "flat")
```

Arguments

vd	Output of function "shape2d"; a list containing fields x, y, z; the meaning of the fealds is the same as in the function "persp"
theta	Viewing angle (left-right); the same parameter as in "persp"
phi	Viewing angle (up-down), the same as in "persp"
typer	"flat" or "dens"; internal

Value

Plots a perspective plot in the graphics window

Author(s)

Jussi Klemela

References

Jussi Klemela (2005). Visualization of the spread of multivariate distributions.

See Also

shape2d

Examples

```
N<-c(30,30)
lnum<-20
func<-"gumbel"
marginal<-"normal"
g<-2
ver<-3
support<-c(-ver,ver,-ver,ver)
st<-stseq(N,lnum,func=func,marginal=marginal,g=g,support=support)

gnum<-50
ngrid=50
vd<-shape2d(st,gnum=gnum,ngrid=ngrid)

plotvolu2d(vd)

plotvolu2d(vd,theta=30)
```

profgene *Calculates the level set tree of a rectangularwise constant function*

Description

Returns the level set tree of a function which is rectangularwise constant.

Usage

```
profgene(values, recs, frekv=NULL, cvol=TRUE, ccen=TRUE, cfre=FALSE,
outlsets=TRUE, invalue=TRUE)
```

Arguments

values	recnum-vector of positive real values; we want to define a rectangularwise constant non-negative function and we denote with "recnum" the number of rectangles at which the function has values greater than 0.
recs	recnum*(2*d)-matrix; corresponding to each element of vector "value" we specify a rectangle. In each row of the "recs" we specify a rectangle at which the estimate has the value given at the corresponding element of "value". Rows of "recs" have the form c(b1,e1,...,bd,ed) when rectangle is the product of intervals [bi,ei], i=1,...,d.
frekv	recnum-vector; for each rectangle the number of observations in this rectangle. Supplying this argument is useful in clustering: one may see the number of observations in each node of the level set tree.
cvol	TRUE if one wants that the volumes of separated parts of the level sets are returned, note that one needs volumes for drawing volume plots
ccen	TRUE if one wants that the barycenters of the separated parts of the level sets are returned
cfre	TRUE if one wants the frequencies of separated parts of the level sets to be returned
outlsets	not needed
invalue	not needed

Value

An augmented level set tree. The level set tree is a list of vectors. The elements of the vectors supply information for each node of the tree. Below we denote with "nodenum" the number of nodes of the tree.

parent	"nodenum"-vector of integers in range 0,..., nodenum-1; links to the parent of each node. Root nodes are marked with 0.
level	"nodenum"-vector of positive real numbers; level of the level set from which the set corresponding to the node is a part of.

volume	"nodenum"-vector of positive real numbers; volume of sets corresponding to each node
center	d*nodenum-matrix; barycenters of sets corresponding to each node
invalue	"nodenum"-vector of positive integers; level of the level set in terms of original frequencies (these values are not normalized so that estimate would integrate to one)
nodefrek	"nodenum"-vector of positive integers; number of observations in the set corresponding to node. Useful in cluster analysis applications.
lsets	nodenum*binnum-matrix; describes the sets associated with nodes. We have 1 in column "c" if the bin described in c:th row of recs is part of the set associated with this node

Note

Applies the naive algorithm of pairwise comparison of the separated components of the level sets, to find which components touch each other.

Author(s)

Jussi Klemela

See Also

profhist,profkern,plotvolu

Examples

```
recnum<-3
d<-2
value<-seq(1:recnum)
recs<-matrix(0,recnum,2*d)
recs[1,]<-c(0,1,0,1)
recs[2,]<-c(0,1,1,2)
recs[3,]<-c(1,2,0,2)
pg<-profgene(value,recs)
```

---

profhist *Calculates the level set tree of a histogram*

Description

Given a data matrix, returns the level set tree of a histogram, which is constructed from bins of equal size.

Usage

```
profhist(dendat, binlkm, cvol=TRUE, ccen=TRUE, cfre=FALSE)
```

Arguments

dendat	n*d data matrix
binlkm	positive integer; number of bins of the histogram estimate in one direction. We use the same number of bins for every direction, thus the total number of bins is binlkm to the power of d
cvol	TRUE if one wants that the volumes of the separated parts of the level sets are returned, note that one needs volumes for drawing the volume plot
ccen	TRUE if one wants that the barycenters of the separated parts of the level sets are returned
cfre	TRUE if one wants that the frequencies of the separated parts of the level sets are returned

Value

An augmented level set tree. The level set tree is a list of vectors. The elements of the vectors supply information for each node of the tree. Below we denote with "nodenum" the number of nodes of the tree.

parent	"nodenum"-vector of integers in range 0,..., nodenum-1; links to the parent of each node. Root nodes are marked with 0.
level	"nodenum"-vector of positive real numbers; level of the level set from which the set corresponding to the node is a part of.
invalue	"nodenum"-vector of positive integers; level of the level set in terms of the original frequencies (these values are not normalized so that estimate would integrate to one
volume	"nodenum"-vector of positive real numbers; gives volumes for sets corresponding to each node
center	d*nodenum-matrix; gives the barycenter for sets corresponding to each node
nodefrek	"nodenum"-vector of positive integers; number of observations in the set corresponding to the node. This is useful in cluster analysis applications.
recs	atomnum*(2*d)-matrix, where atomnum is the number of non-empty bins. Matrix defines the non-empty bins, these are the "atoms" of level sets
hisfrek	atomnum-vector of positive integers; number of observations in non-empty bins
lsets	nodenum*atomnum-matrix: describes the sets associated with the nodes. We have 1 in column "c" if the atom described in c:th row of recs is a part of the set associated with this node

Note

Applies the naive algorithm of pairwise comparison of the separated components of the level sets, to find which components touch each other.

Author(s)

Jussi Klemela

See Also

plotvolu, plotbary, plottree, profgene

Examples

```
set.seed(1)
dendat<-matrix(rnorm(20),10)   #10*2 data-matrix
ph<-profhist(dendat,binlkm=3,cfre=TRUE)
```

---

profkern *Calculates a level set tree of a kernel estimate*

Description

Given a data matrix, returns a level set tree of a kernel estimate, when the dimension of the estimate is less or equal to 4.

Usage

```
profkern(dendat, h, N, Q, cvol=TRUE, ccen=TRUE, cfre=FALSE,
kernel="epane", compoinfo=FALSE, trunc=3, threshold=0.0000001,
sorsa="crc",hw=NULL)
```

Arguments

dendat	n*d data matrix; d<=4
h	positive real number or vector of positive real numbers; smoothing parameters from the largest to the smallest
N	d-vector of dyadic integers >=4; number of knots for every direction: kernel estimate will be evaluated on these knots
Q	positive integer; number of quantization levels
cvol	TRUE or FALSE; TRUE if one wants that the volumes of the separated parts of the level sets are returned, note that one needs the volumes for drawing a volume plot
ccen	T or F; TRUE if one wants that the barycenters of the separated parts of the level sets are returned, note that one needs the barycenters for drawing a barycenter plot
cfre	T or F; TRUE if one wants that the frequencies for separated parts of the level sets are returned (not implemented)
kernel	either "epane" or "gauss"; the kernel of the kernel estimate
compoinfo	TRUE or FALSE; whether definitions of separated regions of level sets will be returned
trunc	positive real number; truncation of the Gaussian kernel to the interval [-trunc,trunc]

threshold	positive real number; kernel will be truncated to 0 if the value is smaller than the threshold (not implemented)
sorsa	if not "crc" uses slower R code
hw	lower level parameter

Value

Returns a list of level set trees: for each value of h the level set tree of the corresponding kernel estimate. If h is not a vector but scalar, then only one level set tree will be returned. The level set tree is a list of vectors. The elements of the vectors supply information for each node of the tree. Below we denote with "nodenum" the number of nodes of the tree.

parent	"nodenum"-vector of integers in the range 0,..., nodenum-1; links to the parent of each node. Root nodes are marked with 0.
level	"nodenum"-vector of positive real numbers; level of the level set from which the set corresponding to the node is a part of.
volume	"nodenum"-vector of positive real numbers; volumes of sets corresponding to each node
center	d*nodenum-matrix; barycenters of sets corresponding to each node
invalue	"nodenum"-vector of positive integers; level of the level set in terms of original frequencies (these values are not normalized so that the estimate would integrate to one
component	Will be returned if "compoinfo"=TRUE; for each node of the level set a tree pointer to "AtomlistAtom"
AtomlistAtom	Will be returned if "compoinfo"=TRUE; pointers to "index"
AtomlistNext	Will be returned if "compoinfo"=TRUE; pointers which define the list of atoms for each separated component (for each node of the level set tree)
index	Will be returned if "compoinfo"=TRUE; matrix with d columns: determines a knot of the multivariate grid where estimate was evaluated
nodefrek	"nodenum"-vector of positive integers; number of observations in the set corresponding to the node. This is useful in cluster analysis applications. (Not implemented.)

Note

Applies the DynaDecompose algorithm, described in the article "Algorithms for manipulation of level sets of nonparametric density estimates", by Jussi Klemela

Author(s)

Jussi Klemela

See Also

plotvolu, plotbary, plottree,

Examples

```
set.seed(1)
dendat<-matrix(rnorm(20),10)      # 10*2 data-matrix
pk<-profkern(dendat,h=1,N=c(8,8),Q=4)
plotvolu(pk)

dendat<-sim.data(n=200,type="mulmod")
pk<-profkern(dendat,h=1,N=c(64,64),Q=30)
plotvolu(pk)
```

`prunemodes` *Prunes modes away from a level set tree or a shape tree*

Description

Prunes smallest modes in terms of the excess mass away, from a level set tree or froma a shape tree. The tree is interpreted as a 1D function (in the sense of a volume function, radius function, or a tail probability function). The pruning may be useful to get cleaner plots, especially a location plot of a shape tree may suffer from spurious modes.

Usage

```
prunemodes(lst, modenum = 1, num = NULL, exmalim = NULL, maxnum = NULL)
```

Arguments

`lst`	A level set tree or a shape tree
`modenum`	positive integer; the desired number of modes
`num`	internal
`exmalim`	internal
`maxnum`	internal

Value

A level set tree or a shape tree, with an additional field "exma.of.modes".

Author(s)

Jussi Klemela

See Also

`leafsfirst`, `treedisc`,

Examples

```
dendat<-sim.data(n=100,type="cross")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))
st<-leafsfirst(pcf,propor=0.01)

td<-treedisc(st,pcf,ngrid=60)
plotbary(td)

td2<-prunemodes(td,modenum=4)
plotbary(td2)
```

`scaletable` *Implements the scale and shape table*

Description

Implements the scale and shape table, which is a dynamic tool to visualize a scale of multivariate estimates and to inspect the shape of the estimates. Takes as an input a scale of estimates.

Usage

```
scaletable(estiseq, paletti = NULL, shift = 0, ptext = 0,
ptextst = 0, bm = NULL, levnum = 60, levnumst = 60,
redu=TRUE, volu.modelabel = TRUE, volu.colo = TRUE,
st.modelabel = FALSE, st.colo = TRUE)
```

Arguments

`estiseq`	a list of density estimates, contains 5 fields: pcfseq, lstseq, stseq, hseq, and type
`paletti`	a character vector of color names
`shift`	a real number; possible shifting of mode tree lines
`ptext`	a real number; lifting of the mode labels
`ptextst`	a real number; lifting of the tail labels
`bm`	branching map; just to fasten the plotting in the case the branching map has been already calculated
`levnum`	positive ineteger; the number of levels in level set trees
`levnumst`	positive ineteger; the number of levels in shape trees
`redu`	TRUE or FALSE; TRUE when the number of levels will be reduced, see argument "levnum"
`volu.modelabel`	TRUE or FALSE; TRUE when the modes are labeled in volume plots (otherwise the identification of the modes between volume plots and barycenter plots is made with colors)
`volu.colo`	TRUE or FALSE; whether volume plots are colored
`st.modelabel`	FALSE or TRUE; whether the modes of radius plots are labeled
`st.colo`	TRUE or FALSE; whether the radius plots are colored

Value

Plots 7 frames: control frame, map of branches and mode graph (the biggest frames), volume plot, and barycenter plot (2 medium frames). radius plot, and location plot (2 small frames).

One chooses from the control panel the frame which one wants to manipulate. To return to the control panel, click at the bottom of the frame which is currently active.

Interaction in the mode graph frame: At the beginning the mode tree shows the first coordinate. Click at the top of the mode tree frame to change the coordinate of the mode tree. Click with the mouse at the scale of smoothing parameter values; the frames of volume plot barycenter plot, radius plot, and location plot change to correspond to this smoothing parameter value. The vertical position of the click point determines the smoothing parameter value, and the horizontal position does not matter. (At the beginning the frames correspond to the estimate with the largest smoothing parameter, and radius plot and location plot show the 10% level set.)

Interaction in the volume plot frame: Click inside of the figure to zoom in (to change the range of x-coordinates). Click at the top of the frame to return to the original scale. Click at the left to the y-coordinate axis to choose the level of the radius plot and location plot.

Interaction in the frame of the barycenter plot: Click at the top of the frame or inside the frame to change the coordinate. (At the beginning the barycenter plot shows the first coordinate.)

Interaction in the frame of the radius plot: Choose with the mouse click at the top of the frame the reference point of the radius and location plot. The reference point is either th elocation of the maximum of the estimate, or the barycenter of the level set.

Interaction in the frame of the location plot: Click at the top of the frame or inside the frame to change the coordinate. (At the beginning the location plot shows the first coordinate.)

Interaction in the frame of the map of branches: Rotate the perspective plot with the mouse clicks. Each click rotates 10 degrees. The direction of the rotation is up or down, left or right, dpending on the position of the mouse click relative to the center of the frame.

To finish the program, click with the mouse at the label "STOP" at the control frame.

Author(s)

Jussi Klemela

See Also

lstseq.kern

Examples

```
dendat<-sim.data(n=200,type="mulmod")

h1<-0.9
h2<-2.2
lkm<-5
hseq<-hgrid(h1,h2,lkm)

N<-c(16,16)
estiseq<-lstseq.kern(dendat,hseq,N,lstree=TRUE,level=0.1)
```

```
# scaletable(estiseq)
```

---

shape2d *Returns a 2D volume function or 2D probability content function*

Description

Calculates a 2D volume function or 2D probability content function from a sequence of shape trees, corresponding to a sequence of level sets of a multivariate function to be visualized. A 2D volume function and 2D probability content function are visualizations of a multivariate density with a 2D function. One joins a series of radius functions or tail probability functions to get a 2D visualization. Radius functions and tail probability functions are 1D functions which visualize the shape of level sets of a multivariate density.

Usage

```
shape2d(shtseq, gnum = 500, type = "radius", type2 = "slice",
gnum2 = 1000, ngrid = 30, norma = FALSE, xmax = 10, modelim = 2,
exmalim = NULL, maxnum = NULL)
```

Arguments

shtseq	A list of shape trees, made by function "stseq"; the shape trees in the list correspond to a grid of level sets of the function to be visualized
gnum	Number of grid points in the radius function or in the tail probability function (used when one transforms a shape tree to a 1D function)
type	"radius" or "proba"; whether a 2D volume function or 2D probability content function will be calculated
type2	"slice" or "boundary"; whether the 1D functions are slices or level sets of the 2D function. The option "boundary" is not recommended at the moment
gnum2	when type2="boundary", then gnum*gnum is the size of the grid of the 2D function, and "gnum2" is the grid used for transforming shape trees to 1D functions
ngrid	positive integer; one prunes the 1D functions to have "ngrid" level sets
norma	TRUE if one uses dimension normalization for the volumes; then volumes are tranformed to (volume/V)pow(1/d), where V is the volume of the d-dimensional unit ball
xmax	internal
modelim	internal
exmalim	internal
maxnum	internal

Value

A list containing elements x, y, z, whose meaning is the same as in function "persp".

Author(s)

Jussi Klemela

References

Jussi Klemela (2005). Visualization of the spread of multivariate distributions.

See Also

stseq, plotvolu2d

Examples

```
N<-c(30,30)
lnum<-20
func<-"gumbel"
marginal<-"normal"
g<-2
ver<-3
support<-c(-ver,ver,-ver,ver)
st<-stseq(N,lnum,func=func,marginal=marginal,g=g,support=support)

gnum<-50
ngrid=50
vd<-shape2d(st,gnum=gnum,ngrid=ngrid)

plotvolu2d(vd)

type<-"proba"
vd3<-shape2d(st,gnum=gnum,type=type,ngrid=ngrid,norma=TRUE)

plotvolu2d(vd3)
```

---

sim.data *Generates data for illustrative purposes*

Description

Returns a random sample from some distributions, to illustrate some visulization tools. Returns also the density (as a piecewise constant function) for some examples, or the distribution function.

Usage

```
sim.data(n = NULL, seed = 1, N = NULL, type = "mulmod",
M = NULL, sig = NULL, p = NULL, d = NULL,
cova = NULL, marginal = NULL, t = NULL, df = NULL, distr = FALSE,
noisedim = 1, sig1 = 0.5, sig2 = 1.5, diff = 0.1, dist = 4)
```

Arguments

n	positive integer; size of the sample to be generated
seed	real number; seed for the random number generator.
N	2*1 vector of positive integers; the size of the grid where the piecewise constant function is evaluated
type	"mixt", "mulmod", "fox", "tetra3d", "penta4d", "cross", "gauss", "student", "gumbel", "1d2modal", or "claw".
M	mixnum*d-matrix; rows of M are means of the Gaussians in the mixture. We have a mixture of "mixnum" Gaussians, whose dimension is d.
sig	mixnum*d-matrix; rows of sig are the diagonals of the covariance matrices of the mixtures.
p	mixnum-vector; weights for the members of the mixture. The sum of elements of "p" is 1.
d	positive integer; dimension of the vectors of the sample to be generated, need to be given only when type="mixt" and d=1
cova	Covariance matrix for the Gauss or Student copulas
marginal	NULL, "gauss", or "student"; this parameter is used to give the marginal distribution for the Gauss or Student copulas; if marginal=NULL, then the uniform marginals are used
t	if marginal="student", gives the degrees of freedom
df	degrees of freedom for the Student copula
distr	internal (implemented for "1d2modal") TRUE, if one wants the distribution function instead of the density function
noisedim	the number of noise dimension in the projection pursuit example ("fssk")
sig1	standard deviation for "cross" and "diff1d"
sig2	second standard deviation for "cross"
diff	parameter for "diff1d"; the difference between the Gaussians in the 1D mixture
dist	a positive real number; gives the distance between the mixture centers in the 4D mixture of Gaussians "penta4d"

Details

When type="mixt", generates data from a mixture of Gaussians. When type="mulmod", the density is 3-modal. When type="fox", the density has multimodal level sets.

Value

If "n" is not NULL, then the function returns a n*d-data matrix or a n*2-data matrix, if "N" is not NULL, then the function returns a piecewise constant function on the grid of size N[1]*N[2], if the both are NULL, then the function returns the mean, covariance, and the weights of the mixture components

Author(s)

Jussi Klemela

Examples

```
d<-2
mixnum<-3
M<-matrix(0,mixnum,d)
M[1,]<-c(0,0)
M[2,]<-c(4,0)
M[3,]<-c(0,4)
sig<-matrix(1,mixnum,d)
p0<-1/mixnum
p<-p0*rep(1,mixnum)
n<-100
dendat<-sim.data(type="mixt",n=n,M=M,sig=sig,p=p,seed=1)
plot(dendat)

dendat<-sim.data(n=100)
plot(dendat)

N<-c(20,20)
pcf<-sim.data(N=N)
dp<-draw.pcf(pcf,pnum=c(30,30))
contour(dp$x,dp$y,dp$z,drawlabels=FALSE)

sim.data()

type="fox"
dendat<-sim.data(n=100,type=type)
plot(dendat)

pcf<-sim.data(N=N,type=type)
dp<-draw.pcf(pcf,pnum=c(30,30))
contour(dp$x,dp$y,dp$z,drawlabels=FALSE)
```

---

slicing *Returns a one- or two-dimensional slice of a multivariate function*

Description

Returns a one- or two-dimensional slice of a multivariate function. The functions are represented as piecewise constant functions.

Usage

```
slicing(pcf, vecci, d1 = 1, d2 = NULL)
```

Arguments

pcf	a piecewise cosntant function
vecci	vector inside the support of the function which fixes the variables; length of "vecci" is d-2 or d-1, depending on whether the slice is 2- or 1-dimensional
d1	integer 1,...,d; the first variable of the slice
d2	integer 1,...,d, not equal to d1; the second variable of the slice

Value

a piecewise constant function

Author(s)

Jussi Klemela

See Also

draw.pcf

Examples

```
# 2D slice of a 3D function

N<-c(26,26,26)  # choose the grid size
copula<-"gauss"
margin<-"student"
b<-4
support<-c(-b,b,-b,b,-b,b)
r<-0.5       # parameter of the copula
t<-c(2,2,3)  # degreeds of freedom for the student margin
pcf<-pcf.func(copula,N,marginal=margin,support=support,r=r,t=t)

sl<-slicing(pcf,d1=1,d2=2,vecci=0)
dp<-draw.pcf(sl)
persp(dp$x,dp$y,dp$z,theta=30,phi=30)

# 1D slice of a 2D function
N<-c(60,60)
pcf<-sim.data(N=N,type="mulmod")

sl<-slicing(pcf,vecci=0)
dp<-draw.pcf(sl)
plot(dp$x,dp$y,type="l")
```

stseq *Calculates a sequence of radius functions from a sequence of level sets*

Description

Calculates a sequence of radius functions from level sets of a function of a given parametric form.

Usage

```
stseq(N, lnum, refe = NULL, func = NULL, dendat = NULL, h = NULL, Q = NULL,
kernel = "epane", weights = NULL, sig = rep(1, length(N)), support = NULL,
theta = NULL, M = NULL, p = NULL, mul = 3, t = rep(1, length(N)),
marginal = "normal", r = 0, mu = NULL, xi = NULL, Omega = NULL, alpha = NULL,
df = NULL, g = 1, base = 10)
```

Arguments

N	d-vector of integers >1; the size of the grid where the function is evaluated
lnum	positive integer; the number of level sets from which the transforms are calculated
refe	d-vector of real numbers; the reference point for the shape trees; if refe=NULL, then the location of the maximum of the function will be used
func	character; the name of the function, see the "pcf.func" for the possibilities
dendat	n*d matrix of data; when the function is a kernel estimate one gives the data as argument
h	postive real number; the smoothing parameter of the kernel estimate
Q	internal
kernel	"gauss" or "epane"; the kernel of the kernel estimate
weights	n vector of nonnegative weights, where n is the sample size; sum of the elements of "weights" should be one; these are the weights of a time localized kernel estimator
sig	see "pcf.func"
support	see "pcf.func"
theta	see "pcf.func"
M	see "pcf.func"
p	see "pcf.func"
mul	see "pcf.func"
t	see "pcf.func"
marginal	see "pcf.func"
r	see "pcf.func"

mu	see "pcf.func"
xi	see "pcf.func"
Omega	see "pcf.func"
alpha	see "pcf.func"
df	see "pcf.func"
g	see "pcf.func"
base	positve integer or NULL; the base of the logarithm, when the logarithmic spacing is used for the level sets, if "base" is NULL, then the level sets are equispaced

Value

A list of radius transforms.

Author(s)

Jussi Klemela

See Also

shape2d, plotvolu2d

Examples

```
N<-c(32,32)
lnum<-30
func<-"prod"
marginal<-"student"
yla<-5
support<-c(-yla,yla,-yla,yla)
g<-c(1,3)
st<-stseq(N,lnum,func=func,marginal=marginal,support=support,g=g)

gnum<-70
vd<-shape2d(st,gnum=gnum)
plotvolu2d(vd)
```

---

tree.segme *Returns the segmentation of the nodes of a visualization tree*

Description

Returns the segmentation of the nodes of a visualization tree. When the visualization tree is a tail tree, finds a grouping for the observations. The grouping may be used to enhance scatter plots, graphical matrices, and parallel coordinate plots.

Usage

```
tree.segme(tt, paletti = NULL, pcf = NULL)
```

Arguments

tt	visualization tree; for example a tail tree
paletti	a sequence of numbers or colors (group labels)
pcf	internal; piecewise constant function

Value

Vector of length n, where n is the number of nodes in the visualization tree. The elements of the vector give the segmentation labels.

Author(s)

Jussi Klemela

See Also

paracoor

Examples

```
dendat<-sim.data(n=1000,type="cross",seed=1)
rho<-1.1
tt<-leafsfirst(dendat=dendat,rho=rho)

ts<-tree.segme(tt)

plot(dendat,col=ts)

paracoor(dendat,paletti=ts)
```

---

treedisc *Prunes a level set tree or a tail tree*

Description

Prunes a level set tree or a tail tree so that it contains fewer levels. The pruning makes plotting of the trees much faster. The quality does not typically decrease essentially when a tree is pruned.

Usage

```
treedisc(lst, pcf, ngrid = NULL, r = NULL, type = NULL, lowest = "dens")
```

Arguments

lst	level set tree or a shape tree; output of for example "leafsfirst"
pcf	piecewise constant function; output of for example "pcf.kern" or "pcf.func"; "lst" should be the level set tree or shape tree of "pcf"
ngrid	positive integer; the number of levels which the pruned tree will have
r	vector of positive reals in increasing order; we may give the set of levels explicitely with the argument "r"; then the levels need not be equispaced as when we apply argument "ngrid"
type	"lst" or "shape"; not needed
lowest	a character; if lowest="dens", then it is assumed that the function whose level set tree we calculate is nonnegative

Value

a level set tree or a shape tree which contains fewer levels than the input

Author(s)

Jussi Klemela

See Also

leafsfirst

Examples

```
dendat<-sim.data(n=100,type="mulmod")
pcf<-pcf.kern(dendat,h=1,N=c(32,32))
lst<-leafsfirst(pcf)          # level set tree
td<-treedisc(lst,pcf,ngrid=30)

plotvolu(td)

r<-c(0.01,0.03,0.04)
td<-treedisc(lst,pcf,r=r)

plotvolu(td)
```

Index