Package ‘hdrcde’

December 21, 2018

Type Package

Title Highest Density Regions and Conditional Density Estimation

Version 3.3

BugReports https://github.com/robjhyndman/hdrcde/issues

Depends R (>= 2.15)

Imports locfit, ash, ks, KernSmooth, ggplot2, RColorBrewer

LazyData yes

LazyLoad yes

Description Computation of highest density regions in one and two dimensions, kernel estimation of univariate density functions conditional on one covariate,and multimodal regression.

License GPL-3

URL http://pkg.robjhyndman.com/hdrcde,
https://github.com/robjhyndman/hdrcde

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation yes

Author Rob Hyndman [aut, cre, cph],
Jochen Einbeck [ctb],
Matthew Wand [ctb]

Maintainer Rob Hyndman <Rob.Hyndman@monash.edu>

Repository CRAN

Date/Publication 2018-12-21 14:50:06 UTC

R topics documented:

`BoxCox` *Box Cox Transformation*

Description

BoxCox() returns a transformation of the input variable using a Box-Cox transformation. InvBox-Cox() reverses the transformation.

Usage

```
BoxCox(x, lambda)
```

Arguments

`x`	a numeric vector or time series
`lambda`	transformation parameter

Details

The Box-Cox transformation is given by

$$f_\lambda(x) = \frac{x^\lambda - 1}{\lambda}$$

if $\lambda \neq 0$. For $\lambda = 0$,

$$f_0(x) = \log(x).$$

Value

a numeric vector of the same length as x.

Author(s)

Rob J Hyndman

References

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. *JRSS B* **26** 211–246.

cde *Conditional Density Estimation*

Description

Calculates kernel conditional density estimate using local polynomial estimation.

Usage

```
cde(x, y, deg = 0, link = "identity", a, b, mean = NULL, x.margin,
  y.margin, x.name, y.name, use.locfit = FALSE, fw = TRUE,
  rescale = TRUE, nxmargin = 15, nymargin = 100, a.nndefault = 0.3,
  ...)
```

Arguments

`x`	Numerical vector or matrix: the conditioning variable(s).
`y`	Numerical vector: the response variable.
`deg`	Degree of local polynomial used in estimation.
`link`	Link function used in estimation. Default "identity". The other possibility is "log" which is recommended if degree > 0.
`a`	Optional bandwidth in x direction.
`b`	Optional bandwidth in y direction.
`mean`	Estimated mean of y|x. If present, it will adjust conditional density to have this mean.
`x.margin`	Values in x-space on which conditional density is calculated. If not specified, an equi-spaced grid of `nxmargin` values over the range of x is used. If x is a matrix, x.margin should be a list of two numerical vectors.
`y.margin`	Values in y-space on which conditional density is calculated. If not specified, an equi-spaced grid of `nymargin` values over the range of y is used.
`x.name`	Optional name of x variable used in plots.
`y.name`	Optional name of y variable used in plots.
`use.locfit`	If TRUE, will use `locfit` for estimation. Otherwise `ksmooth` is used. `locfit` is used if degree>0 or link not the identity or the dimension of x is greater than 1 even if `use.locfit=FALSE`.
`fw`	If TRUE (default), will use fixed window width estimation. Otherwise nearest neighbourhood estimation is used. If the dimension of x is greater than 1, nearest neighbourhood must be used.
`rescale`	If TRUE (default), will rescale the conditional densities to integrate to one.
`nxmargin`	Number of values used in `x.margin` by default.
`nymargin`	Number of values used in `y.margin` by default.
`a.nndefault`	Default nearest neighbour bandwidth (used only if `fw=FALSE` and `a` is missing.).
`...`	Additional arguments are passed to locfit.

Details

If bandwidths are omitted, they are computed using normal reference rules described in Bashtannyk and Hyndman (2001) and Hyndman and Yao (2002). Bias adjustment uses the method described in Hyndman, Bashtannyk and Grunwald (1996). If deg>1 then estimation is based on the local parametric estimator of Hyndman and Yao (2002).

Value

A list with the following components:

`x`	grid in x direction on which density evaluated. Equal to x.margin if specified.
`y`	grid in y direction on which density is evaluated. Equal to y.margin if specified.
`z`	value of conditional density estimate returned as a matrix.
`a`	window width in x direction.
`b`	window width in y direction.
`x.name`	Name of x variable to be used in plots.
`y.name`	Name of y variable to be used in plots.

Author(s)

Rob J Hyndman

References

Hyndman, R.J., Bashtannyk, D.M. and Grunwald, G.K. (1996) "Estimating and visualizing conditional densities". *Journal of Computational and Graphical Statistics*, **5**, 315-336.

Bashtannyk, D.M., and Hyndman, R.J. (2001) "Bandwidth selection for kernel conditional density estimation". *Computational statistics and data analysis*, **36**(3), 279-298.

Hyndman, R.J. and Yao, Q. (2002) "Nonparametric estimation and symmetry tests for conditional density functions". *Journal of Nonparametric Statistics*, **14**(3), 259-278.

See Also

`cde.bandwidths`

Examples

```
# Old faithful data
faithful.cde <- cde(faithful$waiting, faithful$eruptions,
x.name="Waiting time", y.name="Duration time")
plot(faithful.cde)
plot(faithful.cde, plot.fn="hdr")

# Melbourne maximum temperatures with bias adjustment
x <- maxtemp[1:3649]
y <- maxtemp[2:3650]
maxtemp.cde <- cde(x,y,
```

```
x.name="Today's max temperature",y.name="Tomorrow's max temperature")
# Assume linear mean
fit <- lm(y~x)
fit.mean <- list(x=6:45,y=fit$coef[1]+fit$coef[2]*(6:45))
maxtemp.cde2 <- cde(x,y,mean=fit.mean,
x.name="Today's max temperature",y.name="Tomorrow's max temperature")
plot(maxtemp.cde)
```

`cde.bandwidths` *Bandwidth calculation for conditional density estimation*

Description

Calculates bandwidths for kernel conditional density estimates. Methods described in Bashtannyk and Hyndman (2001) and Hyndman and Yao (2002).

Usage

```
cde.bandwidths(x, y, deg = 0, link = "identity", method = 1,
  y.margin, passes = 2, ngrid = 8, min.a = NULL, ny = 25,
  use.sample = FALSE, GCV = TRUE, b = NULL, ...)
```

Arguments

`x`	Numerical vector: the conditioning variable.
`y`	Numerical vector: the response variable.
`deg`	Degree of local polynomial used in estimation.
`link`	Link function used in estimation. Default "identity". The other possibility is "log" which is recommended if degree > 0.
`method`	**method = 1:** Hyndman-Yao algorithm if deg>0; Bashtannyk-Hyndman algorithm if deg=0; **method = 2:** Normal reference rules; **method = 3:** Bashtannyk-Hyndman regression method if deg=0; **method = 4:** Bashtannyk-Hyndman bootstrap method if deg=0.
`y.margin`	Values in y-space on which conditional density is calculated. If not specified, an equi-spaced grid of 50 values over the range of y is used.
`passes`	Number of passes through Bashtannyk-Hyndman algorithm.
`ngrid`	Number of values of smoothing parameter in grid.
`min.a`	Smallest value of a to consider if method=1.
`ny`	Number of values to use for y margin if `y.margin` is missing.
`use.sample`	Used when regression method (3) is chosen.
`GCV`	Generalized cross-validation. Used only if method=1 and deg>0. If GCV=FALSE, method=1 and deg=0, then the AIC is used instead. The argument is ignored if deg=0 or method>1.

b	Value of b can be specified only if method=1 and deg>0. For deg=0 or method>1, this argument is ignored.
...	Other arguments control details for individual methods.

Details

Details of the various algorithms are in Bashtannyk and Hyndman (2001) and Hyndman and Yao (2002).

Value

a	Window width in x direction.
b	Window width in y direction.

Author(s)

Rob J Hyndman

References

Hyndman, R.J., Bashtannyk, D.M. and Grunwald, G.K. (1996) "Estimating and visualizing conditional densities". *Journal of Computational and Graphical Statistics*, **5**, 315-336.

Bashtannyk, D.M., and Hyndman, R.J. (2001) "Bandwidth selection for kernel conditional density estimation". *Computational statistics and data analysis*, **36**(3), 279-298.

Hyndman, R.J. and Yao, Q. (2002) "Nonparametric estimation and symmetry tests for conditional density functions". *Journal of Nonparametric Statistics*, **14**(3), 259-278.

See Also

cde

Examples

```
bands <- cde.bandwidths(faithful$waiting,faithful$eruptions,method=2)
plot(cde(faithful$waiting,faithful$eruptions,a=bands$a,b=bands$b))
```

---

hdr *Highest Density Regions*

Description

Calculates and plots highest density regions in one dimension including the HDR boxplot.

Usage

```
hdr(x = NULL, prob = c(50, 95, 99), den = NULL, h = hdrbw(BoxCox(x,
  lambda), mean(prob)), lambda = 1, nn = 5000, all.modes = FALSE)

hdr.den(x, prob = c(50, 95, 99), den, h = hdrbw(BoxCox(x, lambda),
  mean(prob)), lambda = 1, xlab = NULL, ylab = "Density",
  ylim = NULL, plot.lines = TRUE, col = 2:8, bgcol = "gray",
  legend = FALSE, ...)

hdr.boxplot(x, prob = c(99, 50), h = hdrbw(BoxCox(x, lambda),
  mean(prob)), lambda = 1, boxlabels = "", col = gray((9:1)/10),
  main = "", xlab = "", ylab = "", pch = 1, border = 1,
  outline = TRUE, space = 0.25, ...)
```

Arguments

`x`	Numeric vector containing data. In `hdr` and `hdr.den`, if `x` is missing then `den` must be provided, and the HDR is computed from the given density. For `hdr.boxplot`, `x` can be a list containing several vectors.
`prob`	Probability coverage required for HDRs
`den`	Density of data as list with components `x` and `y`. If omitted, the density is estimated from `x` using `density`.
`h`	Optional bandwidth for calculation of density.
`lambda`	Box-Cox transformation parameter where `0 <= lambda <= 1`.
`nn`	Number of random numbers used in computing f-alpha quantiles.
`all.modes`	Return all local modes or just the global mode?
`xlab`	Label for x-axis.
`ylab`	Label for y-axis.
`ylim`	Limits for y-axis.
`plot.lines`	If `TRUE`, will show how the HDRs are determined using lines.
`col`	Colours for regions of each box.
`bgcol`	Colours for the background behind the boxes. Default `"gray"`, if `NULL` no box is drawn.
`legend`	If `TRUE` add a legend on the right of the boxes.
`...`	Other arguments passed to plot.
`boxlabels`	Label for each box plotted.
`main`	Overall title for the plot.
`pch`	Plotting character.
`border`	Width of border of box.
`outline`	If not <code>TRUE</code>, the outliers are not drawn.
`space`	The space between each box, between 0 and 0.5.

Details

Either x or den must be provided. When x is provided, the density is estimated using kernel density estimation. A Box-Cox transformation is used if lambda!=1, as described in Wand, Marron and Ruppert (1991). This allows the density estimate to be non-zero only on the positive real line. The default kernel bandwidth h is selected using the algorithm of Samworth and Wand (2010).

Hyndman's (1996) density quantile algorithm is used for calculation. hdr.den plots the density with the HDRs superimposed. hdr.boxplot displays a boxplot based on HDRs.

Value

hdr.boxplot retuns nothing. hdr and hdr.den return a list of three components:

hdr	The endpoints of each interval in each HDR
mode	The estimated mode of the density.
falpha	The value of the density at the boundaries of each HDR.

Author(s)

Rob J Hyndman

References

Hyndman, R.J. (1996) Computing and graphing highest density regions. *American Statistician*, **50**, 120-126.

Samworth, R.J. and Wand, M.P. (2010). Asymptotics and optimal bandwidth selection for highest density region estimation. *The Annals of Statistics*, **38**, 1767-1792.

Wand, M.P., Marron, J S., Ruppert, D. (1991) Transformations in density estimation. *Journal of the American Statistical Association*, **86**, 343-353.

See Also

hdr.boxplot.2d

Examples

```
# Old faithful eruption duration times
hdr(faithful$eruptions)
hdr.boxplot(faithful$eruptions)
hdr.den(faithful$eruptions)

# Simple bimodal example
x <- c(rnorm(100,0,1), rnorm(100,5,1))
par(mfrow=c(1,2))
boxplot(x)
hdr.boxplot(x)
par(mfrow=c(1,1))
hdr.den(x)
```

```
# Highly skewed example
x <- exp(rnorm(100,0,1))
par(mfrow=c(1,2))
boxplot(x)
hdr.boxplot(x,lambda=0)
```

`hdr.2d` *Bivariate Highest Density Regions*

Description

Calculates and plots highest density regions in two dimensions, including the bivariate HDR boxplot.

Usage

```
hdr.2d(x, y, prob = c(50, 95, 99), den = NULL, kde.package = c("ash",
  "ks"), h = NULL, xextend = 0.15, yextend = 0.15)

hdr.boxplot.2d(x, y, prob = c(50, 99), kde.package = c("ash", "ks"),
  h = NULL, xextend = 0.15, yextend = 0.15, xlab = "", ylab = "",
  shadecols = gray((length(prob):1)/(length(prob) + 1)), pointcol = 1,
  ...)

## S3 method for class 'hdr2d'
plot(x, shaded = TRUE, show.points = FALSE,
  outside.points = FALSE, pch = 20,
  shadecols = gray((length(x$alpha):1)/(length(x$alpha) + 1)),
  pointcol = 1, ...)
```

Arguments

`x`	Numeric vector
`y`	Numeric vector of same length as `x`.
`prob`	Probability coverage required for HDRs
`den`	Bivariate density estimate (a list with elements x, y and z where x and y are grid values and z is a matrix of density values). If `NULL`, the density is estimated.
`kde.package`	Package to be used in calculating the kernel density estimate when `den=NULL`.
`h`	Pair of bandwidths passed to either `ash2` or `kde`. If NULL, a reasonable default is used. Ignored if `den` is not `NULL`.
`xextend`	Proportion of range of `x`. The density is estimated on a grid extended by `xextend` beyond the range of `x`.
`yextend`	Proportion of range of `y`. The density is estimated on a grid extended by `yextend` beyond the range of `y`.

xlab	Label for x-axis.
ylab	Label for y-axis.
shadecols	Colors for shaded regions
pointcol	Color for outliers and mode
...	Other arguments to be passed to plot.
shaded	If TRUE, the HDR contours are shown as shaded regions.
show.points	If TRUE, the observations are plotted over the top of the HDR contours.
outside.points	If TRUE, the observations lying outside the largest HDR are shown.
pch	The plotting character used for observations.

Details

The density is estimated using kernel density estimation. Either ash2 or kde is used to do the calculations. Then Hyndman's (1996) density quantile algorithm is used to compute the HDRs.

hdr.2d returns an object of class hdr2d containing all the information needed to compute the HDR contours. This object can be plotted using plot.hdr2d.

hdr.boxplot.2d produces a bivariate HDR boxplot. This is a special case of applying plot.hdr2d to an object computed using hdr.2d.

Value

Some information about the HDRs is returned. See code for details.

Author(s)

Rob J Hyndman

References

Hyndman, R.J. (1996) Computing and graphing highest density regions *American Statistician*, **50**, 120-126.

See Also

hdr.boxplot

Examples

```
x <- c(rnorm(200,0,1),rnorm(200,4,1))
y <- c(rnorm(200,0,1),rnorm(200,4,1))
hdr.boxplot.2d(x,y)

hdrinfo <- hdr.2d(x,y)
plot(hdrinfo, pointcol="red", show.points=TRUE, pch=3)
```

hdr.cde	*Calculate highest density regions continously over some conditioned variable.*

Description

Calculates and plots highest density regions for a conditional density estimate. Uses output from `cde`.

Usage

```
hdr.cde(den, prob = c(50, 95, 99), plot = TRUE, plot.modes = TRUE,
  mden = rep(1, length(den$x)), threshold = 0.05, nn = 1000, xlim,
  ylim, xlab, ylab, border = TRUE, font = 1, cex = 1, ...)
```

Arguments

`den`	Conditional density in the same format as the output from `cde`.
`prob`	Probability coverage level for HDRs
`plot`	Should HDRs be plotted? If FALSE, results are returned.
`plot.modes`	Should modes be plotted as well as HDRs?
`mden`	Marginal density in the `x` direction. When small, the HDRs won't be plotted. Default is uniform so all HDRs are plotted.
`threshold`	Threshold for margin density. HDRs are not plotted if the margin density `mden` is lower than this value.
`nn`	Number of points to be sampled from each density when estimating the HDRs.
`xlim`	Limits for x-axis.
`ylim`	Limits for y-axis.
`xlab`	Label for x-axis.
`ylab`	Label for y-axis.
`border`	Show border of polygons
`font`	Font to be used in plot.
`cex`	Size of characters.
`...`	Other arguments passed to plotting functions.

Value

`hdr`	array (a,b,c) where where a specifies conditioning value, b gives the HDR end-points and c gives the probability coverage.
`modes`	estimated mode of each conditional density

Author(s)

Rob J Hyndman

References

Hyndman, R.J., Bashtannyk, D.M. and Grunwald, G.K. (1996) "Estimating and visualizing conditional densities". *Journal of Computational and Graphical Statistics*, **5**, 315-336.

See Also

cde, hdr

Examples

```
faithful.cde <- cde(faithful$waiting,faithful$eruptions)
plot(faithful.cde,xlab="Waiting time",ylab="Duration time",plot.fn="hdr")
```

---

hdrbw *Highest Density Region Bandwidth*

Description

Estimates the optimal bandwidth for 1-dimensional highest density regions

Usage

```
hdrbw(x, HDRlevel, gridsize = 801, nMChdr = 1e+06,
  graphProgress = FALSE)
```

Arguments

x	Numerical vector containing data.
HDRlevel	HDR-level as defined in Hyndman (1996). Setting 'HDRlevel' equal to p (0<p<1) corresponds to a probability of 1-p of inclusion in the highest density region.
gridsize	the number of equally spaced points used for binned kernel density estimation.
nMChdr	the size of the Monte Carlo sample used for density quantile approximation of the highest density region, as described in Hyndman (1996).
graphProgress	logical flag: if 'TRUE' then plots showing the progress of the bandwidth selection algorithm are produced.

Details

This is a plug-in rule for bandwidth selection tailored to highest density region estimation

Value

A numerical vector of length 1.

Author(s)

Matt Wand

References

Hyndman, R.J. (1996). Computing and graphing highest density regions. *The American Statistician*, **50**, 120-126.

Samworth, R.J. and Wand, M.P. (2010). Asymptotics and optimal bandwidth selection for highest density region estimation. *The Annals of Statistics*, **38**, 1767-1792.

Examples

```
HDRlevelVal <- 0.55
x <- faithful$eruptions
hHDR <- hdrbw(x,HDRlevelVal)
HDRhat <- hdr.den(x,prob=100*(1-HDRlevelVal),h=hHDR)
```

`hdrconf` *HDRs with confidence intervals*

Description

Calculates Highest Density Regions with confidence intervals.

Usage

```
hdrconf(x, den, prob = 95, conf = 95)
```

Arguments

`x`	Numeric vector containing data.
`den`	Density of data as list with components `x` and `y`.
`prob`	Probability coverage for for HDRs.
`conf`	Confidence for limits on HDR.

Value

`hdrconf` returns list containing the following components:

`hdr`	Highest density regions
`hdr.lo`	Highest density regions corresponding to lower confidence limit.
`hdr.hi`	Highest density regions corresponding to upper confidence limit.
`falpha`	Values of f_α corresponding to HDRs.
`falpha.ci`	Values of f_α corresponding to lower and upper limits.

Author(s)

Rob J Hyndman

References

Hyndman, R.J. (1996) Computing and graphing highest density regions *American Statistician*, **50**, 120-126.

See Also

hdr, plot.hdrconf

Examples

```
x <- c(rnorm(100,0,1),rnorm(100,4,1))
den <- density(x,bw=hdrbw(x,50))
trueden <- den
trueden$y <- 0.5*(exp(-0.5*(den$x*den$x)) + exp(-0.5*(den$x-4)^2))/sqrt(2*pi)
sortx <- sort(x)

par(mfcol=c(2,2))
for(conf in c(50,95))
{
    m <- hdrconf(sortx,trueden,conf=conf)
    plot(m,trueden,main=paste(conf,"% HDR from true density"))
    m <- hdrconf(sortx,den,conf=conf)
    plot(m,den,main=paste(conf,"% HDR from empirical density\n(n=200)"))
}
```

`hdrscatterplot` *Scatterplot showing bivariate highest density regions*

Description

Produces a scatterplot where the points are coloured according to the bivariate HDRs in which they fall.

Usage

```
hdrscatterplot(x, y, levels = c(1, 50, 99), kde.package = c("ash",
  "ks"), noutliers = NULL)
```

Arguments

`x`	Numeric vector or matrix with 2 columns.
`y`	Numeric vector of same length as `x`.
`levels`	Percentage coverage for HDRs
`kde.package`	Package to be used in calculating the kernel density estimate when `den=NULL`.
`noutliers`	Number of outliers to be labelled. By default, all points outside the largest HDR are labelled.

Details

The bivariate density is estimated using kernel density estimation. Either `ash2` or `kde` is used to do the calculations. Then Hyndman's (1996) density quantile algorithm is used to compute the HDRs. The scatterplot of (x,y) is created where the points are coloured according to which HDR they fall. A ggplot object is returned.

Author(s)

Rob J Hyndman

See Also

`hdr.boxplot.2d`

Examples

```
x <- c(rnorm(200, 0, 1), rnorm(200, 4, 1))
y <- c(rnorm(200, 0, 1), rnorm(200, 4, 1))
hdrscatterplot(x, y)
```

`lane2` *Speed-Flow data for Californian Freeway*

Description

These are two data sets collected in 1993 on two individual lanes (lane 2 and lane 3) of the 4-lane Californian freeway I-880. The data were collected by loop detectors, and the time units are 30 seconds per observation (see Petty et al., 1996, for details).

Usage

```
lane2; lane3
```

Format

Two data frames (`lane2` and `lane3`) each with 1318 observations on the following two variables:

flow a numeric vector giving the traffic flow in vehicles per lane per hour.

speed a numeric vector giving the speed in miles per hour.

Details

The data is examined in Einbeck and Tutz (2006), using a nonparametric approach to multi-valued regression based on conditional mean shift.

Source

Petty, K.F., Noeimi, H., Sanwal, K., Rydzewski, D., Skabardonis, A., Varaiya, P., and Al-Deek, H. (1996). “The Freeway Service Patrol Evaluation Project: Database Support Programs, and Accessibility”. *Transportation Research Part C: Emerging Technologies*, **4**, 71-85.

The data is provided by courtesy of CALIFORNIA PATH, Institute of Transportation Studies, University of California, Berkeley.

References

Einbeck, J., and Tutz, G. (2006). “Modelling beyond regression functions: an application of multimodal regression to speed-flow data”. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **55**, 461-475.

Examples

```
plot(lane2)
plot(lane3)
```

`maxtemp` *Daily maximum temperatures in Melbourne, Australia*

Description

Daily maximum temperatures in Melbourne, Australia, from 1981-1990. Leap days have been omitted.

Usage

```
maxtemp
```

Format

Time series of frequency 365.

Source

Hyndman, R.J., Bashtannyk, D.M. and Grunwald, G.K. (1996) "Estimating and visualizing conditional densities". *Journal of Computational and Graphical Statistics*, **5**, 315-336.

Examples

```
plot(maxtemp)
```

---

modalreg — *Nonparametric Multimodal Regression*

Description

Nonparametric multi-valued regression based on the modes of conditional density estimates.

Usage

```
modalreg(x, y, xfix = seq(min(x), max(x), l = 50), a, b, deg = 0,
  iter = 30, P = 2, start = "e", prun = TRUE, prun.const = 10,
  plot.type = c("p", 1), labels = c("", "x", "y"), pch = 20, ...)
```

Arguments

`x`	Numerical vector: the conditioning variable.
`y`	Numerical vector: the response variable.
`xfix`	Numerical vector corresponding to the input values of which the fitted values shall be calculated.
`a`	Optional bandwidth in x-direction.
`b`	Optional bandwidth in y-direction.
`deg`	Degree of local polynomial used in estimation (0 or 1).
`iter`	Positive integer giving the number of mean shift iterations per point and branch.
`P`	Maximal number of branches.
`start`	Character determining how the starting points are selected. `"q"`: proportional to quantiles; `"e"`: equidistant; `"r"`: random. All, `"q"`, `"e"`, and `"r"`, give starting points which are constant over `x`. As an alternative, the choice `"v"` gives variable starting points, which are equal to `"q"` for the smallest `x`, and equal to the previously fitted values for all subsequent `x`.
`prun`	Boolean. If TRUE, parts of branches are dismissed (in the plotted output) where their associated kernel density value falls below the threshold `1/(prun.const*(max(x)-min(x))*(max(`
`prun.const`	Numerical value giving the constant used above (the higher, the less pruning)

plot.type	Vector with two elements. The first one is character-valued, with possible values "p", "l", and "n". If equal to "n", no plotted output is given at all. If equal to "p", fitted curves are symbolized as points in the graphical output, otherwise as lines. The second vector component is a numerical value either being 0 or 1. If 1, the position of the starting points is depicted in the plot, otherwise omitted.
labels	Vector of three character strings. The first one is the "main" title of the graphical output, the second one is the label of the x axis, and the third one the label of the y axis.
pch	Plotting character. The default corresponds to small bullets.
...	Other arguments passed to cde.bandwidths.

Details

Computes multi-modal nonparametric regression curves based on the maxima of conditional density estimates. The tool for the estimation is the conditional mean shift as outlined in Einbeck and Tutz (2006). Estimates of the conditional modes might fluctuate highly if deg=1. Hence, deg=0 is recommended. For bandwidth selection, the hybrid rule introduced by Bashtannyk and Hyndman (2001) is employed if deg=0. This corresponds to the setting method=1 in function cde.bandwidths. For deg=1 automatic bandwidth selection is not supported.

Value

A list with the following components:

xfix	Grid of predictor values at which the fitted values are calculated.
fitted.values	A [P x length(xfix)]- matrix with fitted j-th branch in the j-th row ($1 \leq j \leq P$)
bandwidths	A vector with bandwidths a and b.
density	A [P x length(xfix)]- matrix with estimated kernel densities. This will only be computed if prun=TRUE.
threshold	The pruning threshold.

Author(s)

Jochen Einbeck (2007)

References

Einbeck, J., and Tutz, G. (2006) "Modelling beyond regression functions: an application of multimodal regression to speed-flow data". *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **55**, 461-475.

Bashtannyk, D.M., and Hyndman, R.J. (2001) "Bandwidth selection for kernel conditional density estimation". *Computational Statistics and Data Analysis*, **36**(3), 279-298.

tests for conditional density functions". *Journal of Statistics*, **14**(3), 259-278.

See Also

cde.bandwidths

Examples

```
lane2.fit <- modalreg(lane2$flow, lane2$speed, xfix=(1:55)*40, a=100, b=4)
```

`plot.cde` *Plots conditional densities*

Description

Produces stacked density plots or highest density region plots for a univariate density conditional on one covariate.

Usage

```
## S3 method for class 'cde'
plot(x, firstvar = 1,
  mfrow = n2mfrow(dim(x$z)[firstvar]), plot.fn = "stacked", x.name,
  margin = NULL, ...)
```

Arguments

`x`	Output from `cde`.
`firstvar`	If there is more than one conditioning variable, `firstvar` specifies which variable to fix first.
`mfrow`	If there is more than one conditioning variable, `mfrow` is passed to `par` before plotting.
`plot.fn`	Specifies which plotting function to use: "stacked" results in stacked conditional densities and "hdr" results in highest density regions.
`x.name`	Name of x (conditioning) variable for use on x-axis.
`margin`	Marginal density of conditioning variable. If present, only conditional densities corresponding to non-negligible marginal densities will be plotted.
`...`	Additional arguments to plot.

Value

If `plot.fn=="stacked"` and there is only one conditioning variable, the function returns the output from `persp`. If `plot.fn=="hdr"` and there is only one conditioning variable, the function returns the output from `hdr.cde`. When there is more than one conditioning variable, nothing is returned.

Author(s)

Rob J Hyndman

References

Hyndman, R.J., Bashtannyk, D.M. and Grunwald, G.K. (1996) "Estimating and visualizing conditional densities". *Journal of Computational and Graphical Statistics*, **5**, 315-336.

See Also

hdr.cde, cde, hdr

Examples

```
faithful.cde <- cde(faithful$waiting,faithful$eruptions,
x.name="Waiting time", y.name="Duration time")
plot(faithful.cde)
plot(faithful.cde,plot.fn="hdr")
```

---

plot.hdrconf *Plot HDRs with confidence intervals*

Description

Plots Highest Density Regions with confidence intervals.

Usage

```
## S3 method for class 'hdrconf'
plot(x, den, ...)
```

Arguments

x	Output from hdrconf.
den	Density of data as list with components x and y.
...	Other arguments are passed to plot.

Value

None

Author(s)

Rob J Hyndman

References

Hyndman, R.J. (1996) Computing and graphing highest density regions *American Statistician*, **50**, 120-126.

See Also

hdrconf

Examples

```
x <- c(rnorm(100,0,1),rnorm(100,4,1))
den <- density(x,bw=bw.SJ(x))
trueden <- den
trueden$y <- 0.5*(exp(-0.5*(den$x*den$x)) + exp(-0.5*(den$x-4)^2))/sqrt(2*pi)
sortx <- sort(x)

par(mfcol=c(2,2))
for(conf in c(50,95))
{
m <- hdrconf(sortx,trueden,conf=conf)
plot(m,trueden,main=paste(conf,"% HDR from true density"))
m <- hdrconf(sortx,den,conf=conf)
plot(m,den,main=paste(conf,"% HDR from empirical density\n(n=200)"))
}
```

Index