# Package ‘hdrcde’

February 20, 2015

**Type** Package  
**Title** Highest density regions and conditional density estimation  
**Version** 3.1  
**Date** 2013-10-19  
**Author** Rob J Hyndman <Rob.Hyndman@monash.edu> with contributions from Jochen Einbeck and Matt Wand  
**Maintainer** Rob J Hyndman <Rob.Hyndman@monash.edu>  
**Depends** R (>= 2.15), mvtnorm  
**Imports** locfit, ash, ks, KernSmooth  
**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 (>= 2)  
**URL** [http://www.robjhyndman.com/software/hdrcde](http://www.robjhyndman.com/software/hdrcde)  
**NeedsCompilation** yes  
**Repository** CRAN  
**Date/Publication** 2013-10-19 15:53:16

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxCox</td>
<td>2</td>
</tr>
<tr>
<td>cde</td>
<td>3</td>
</tr>
<tr>
<td>cde.bandwidths</td>
<td>5</td>
</tr>
<tr>
<td>hdr</td>
<td>6</td>
</tr>
<tr>
<td>hdr.boxplot.2d</td>
<td>8</td>
</tr>
<tr>
<td>hdr.cde</td>
<td>10</td>
</tr>
<tr>
<td>hdrbw</td>
<td>11</td>
</tr>
<tr>
<td>hdrconf</td>
<td>12</td>
</tr>
</tbody>
</table>
Description

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

Usage

BoxCox(x, lambda)
InvBoxCox(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

Description
Calculates kernel conditional density estimate using local polynomial estimation.

Usage
```
cde(x, y, deg = 0, link = "identity", a, b, mean,
   x.margin, y.margin, x.name, y.name,
   use.locfit = FALSE, fw = TRUE, rescale = TRUE, nxmargin=15, nymargin=100,
   a.nnndefault=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.nnndefault**: 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


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)

## 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


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, prob = c(50, 95, 99), den, 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", ...)
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, ...)
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 $0 \leq \lambda \leq 1$.
- **nn**: Number of random numbers used in computing f-alpha quantiles.
- **all.modes**: Return all local modes or just the global mode?
- **boxlabels**: Label for each box plotted.
- **col**: Colours for regions of each box.
- **main**: Overall title for the plot.
- **xlab**: Label for x-axis.
- **ylab**: Label for y-axis.
- **pch**: Plotting character.
- **...**: Other arguments passed to `plot`.

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` returns 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


See Also

*hdr.boxplot.2d*

Examples

```r
# 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.boxplot.2d**  
*Bivariate Highest Density Regions*

Description

Calculates and plots highest density regions in two dimensions, including the bivariate HDR box-plot.

Usage

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

## S3 method for class 'hdr2d'
plot(x, shaded=TRUE, show.points=FALSE, outside.points=FALSE, pch=20,
```
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** is **NULL**.
- **h**: Pair of bandwidths passed to either **ashR** 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
- **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.
- **...**: Other arguments to be passed to plot.

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

See Also
hdr.boxplot

Examples
```r
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)
```

dr.cde

*Calculate highest density regions continuously over some conditioned variable.*

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

Usage
```r
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.
 hdrbw

**Description**

Estimates the optimal bandwidth for 1-dimensional highest density regions

**Usage**

```r
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**


**Examples**

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

---

**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 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 $f_\alpha$ corresponding to HDRs.
- **falpha.ci**: Values of $f_\alpha$ corresponding to lower and upper limits.

Author(s)

Rob J Hyndman

References


See Also

- `hdr`
- `plotNhdrconf`

Examples

```r
x <- c(rnorm(100,0,1),rnorm(100,4,1))
den <- density(x,bw=hdrbw(x,50))
trueden <- den
trueden$sy <- 0.5*(exp(-0.5*(den$sx*den$x)) + exp(-0.5*(den$sx-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,truexden,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)"))
}
```
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

data(lane2)
data(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


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

References


Examples

plot(lane2$flow, lane2$speed)
Description

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

Format

Time series of frequency 365.

Source


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 $1/(\text{prun.const}*(\text{max}(x)-\text{min}(x))*(\text{max}(y)-\text{min}(y)))$.

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 $\text{deg}=1$. Hence, $\text{deg}=0$ is recommended. For bandwidth selection, the hybrid rule introduced by Bashtannyk and Hyndman (2001) is employed if $\text{deg}=0$. This corresponds to the setting $\text{method}=1$ in function cde.bandwidths. For $\text{deg}=1$ 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 $[P \times \text{length(xfix)}]$-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 $[P \times \text{length(xfix)}]$-matrix with estimated kernel densities. This will only be computed if prun=TRUE.

threshold  The pruning threshold.

Author(s)
Jochen Einbeck (2007)
References


See Also
cde.bandwidths

Examples

```r
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**

```r
## 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.
plot.hdrconf

Plot HDRs with confidence intervals

Description

Plots Highest Density Regions with confidence intervals.

Usage

```r
## 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

Examples

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

Author(s)

Rob J Hyndman

References


See Also

`hdr.cde`, `cde`, `hdr`
**Author(s)**
Rob J Hyndman

**References**

**See Also**
hdrconf

**Examples**
```r
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

*Topic datasets
  lane, 14
  maxtemp, 15
*Topic distribution
cde, 3
cde.bandwidths, 5
hdr, 6
hdr.boxplot.2d, 8
hdr.cde, 10
hdrbw, 11
hdrconf, 12
plot.cde, 17
plot.hdrconf, 18
*Topic hplot
cde, 3
hdr, 6
hdr.boxplot.2d, 8
hdr.cde, 10
plot.cde, 17
plot.hdrconf, 18
*Topic math
BoxCox, 2
*Topic nonparametric
modalreg, 15
*Topic regression
modalreg, 15
*Topic smooth
cde, 3
cde.bandwidths, 5
hdr, 6
hdr.boxplot.2d, 8
hdr.cde, 10
hdrbw, 11
hdrconf, 12
plot.cde, 17
plot.hdrconf, 18
ash2, 9
BoxCox, 2
cde, 3, 6, 10, 11, 17, 18
cde.bandwidths, 4, 5, 16, 17
density, 7
hdr, 6, 11, 13, 18
hdr.2d (hdr.boxplot.2d), 8
hdr.boxplot, 10
hdr.boxplot.2d, 8, 8
hdr.cde, 10, 18
hdrbw, 11
hdrconf, 12, 19
InvBoxCox (BoxCox), 2
kde, 9
ksmooth, 3
lane, 14
lane2 (lane), 14
lane3 (lane), 14
locfit, 3
maxtemp, 15
modalreg, 15
par, 17
persp, 18
plot.cde, 17
plot.hdr2d (hdr.boxplot.2d), 8
plot.hdrconf, 13, 18