Package ‘lpcde’

November 16, 2024

Type Package

Title Boundary Adaptive Local Polynomial Conditional Density Estimator

Version 0.1.6

Maintainer Rajita Chandak <rajita.chandak@epfl.ch>

Description Tools for estimation and inference of conditional densities, derivatives and functions. This is the companion software for Cattaneo, Chandak, Jansson and Ma (2024) <doi:10.3150/23-BEJ1711>.

Depends R (>= 3.3.0)

License GPL (>= 3)

Encoding UTF-8

LazyData false

SystemRequirements GNU make

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports Rcpp (>= 0.12.8),
ggplot2,
purrr,
MASS,
mvtnorm,
combinat,
Matrix,
Rdpack,
stats

RdMacros Rdpack

LinkingTo Rcpp,
RcppArmadillo

Suggests testthat (>= 3.0.0),
covr

Config/testthat/edition 3

Contents

basis_vec *Unit basis vector*

Description

Function to generate unit basis vector according to polynomial order and derivative order. This function returns unit vector that is the same size as the vector returned by `poly_base(x, p)`.

Usage

```
basis_vec(x, p, mu)
```

Arguments

`x`	Sample input scalar or vector.
`p`	Polynomial order.
`mu`	Derivative order.

Value

Vector of appropriate length with ones corresponding to entries of order `mu`.

Examples

```
basis_vec(x = 2, p = 5, mu = 1)
```

`coef.lpbwcde` *Coef method for local polynomial density bandwidth selection*

Description

The coef method for local polynomial density bandwidth selection objects.

Usage

```
## S3 method for class 'lpbwcde'
coef(object, ...)
```

Arguments

`object`	Class "lpbwcde" object, obtained by calling `lpbwcde`.
`...`	Other arguments.

Value

`Matrix`	A matrix containing y_grid points and selected bandwidths.

Author(s)

Matias D. Cattaneo, Princeton University. <`cattaneo@princeton.edu`>.

Rajita Chandak (maintainer), Princeton University. <`rchandak@princeton.edu`>.

Michael Jansson, University of California Berkeley. <`mjansson@econ.berkeley.edu`>.

Xinwei Ma, University of California San Diego. <`x1ma@ucsd.edu`>.

See Also

`lpbwcde` for data-driven bandwidth selection.

Supported methods: `coef.lpbwcde`, `print.lpbwcde`, `summary.lpbwcde`.

n=100 x_data = as.matrix(rnorm(n, mean=0, sd=1)) y_data = as.matrix(rnorm(n, mean=0, sd=1)) y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))bandwidth selection

y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1)) model2 = lpcde::lpbwcde(y_data=y_data, x_data=x_data, x=0, y_grid = y_grid, bw_type = "mse-rot") coef(model2) coef.lpcde *Coef method for local polynomial density conditional estimation*

Description

The coef method for local polynomial conditional density objects.

Usage

```
## S3 method for class 'lpcde'
coef(object, ...)
```

Arguments

object	Class "lpcde" object, obtained by calling lpcde.
...	Additional options.

Value

outputs	A matrix containing the estimates

Author(s)

Matias D. Cattaneo, Princeton University. <cattaneo@princeton.edu>.

Rajita Chandak (maintainer), Princeton University. <rchandak@princeton.edu>

Michael Jansson, University of California Berkeley. <mjansson@econ.berkeley.edu>.

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>.

See Also

lpcde for local polynomial conditional density estimation.

Supported methods: coef.lpcde, confint.lpcde, plot.lpcde, print.lpcde, summary.lpcde, vcov.lpcde

Examples

```
n=100
x_data = as.matrix(rnorm(n, mean=0, sd=1))
y_data = as.matrix(rnorm(n, mean=0, sd=1))
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
# density estimation
model1 = lpcde::lpcde(x_data=x_data, y_data=y_data, y_grid=y_grid, x=0, bw=0.5)
coef(model1)
```

confint.lpcde *Confint method for local polynomial density conditional estimation*

Description

The confint method for local polynomial conditional density objects.

Usage

```
## S3 method for class 'lpcde'
confint(
  object,
  parm = NULL,
  level = 0.95,
  CIuniform = FALSE,
  CIsimul = 2000,
  ...
)
```

Arguments

`object`	Class "lpdensity" object, obtained by calling `lpcde`.
`parm`	Integer, indicating which parameters are to be given confidence intervals.
`level`	Numeric scalar between 0 and 1, the confidence level for computing confidence intervals/bands. Equivalent to (1-significance level).
`CIuniform`	`TRUE` or `FALSE` (default), plotting either pointwise confidence intervals (`FALSE`) or uniform confidence bands (`TRUE`).
`CIsimul`	Positive integer, specifies the number of simulations used to construct critical values (default is `2000`). This option is ignored if `CIuniform=FALSE`.
`...`	Additional options, including (i) `grid` specifies a subset of grid points to display the bandwidth; (ii) `gridIndex` specifies the indices of grid points to display the bandwidth (this is the same as `parm`);(iii) `CIuniform` specifies whether displaying pointwise confidence intervals (`FALSE`, default) or the uniform confidence band (`TRUE`); (iv) `CIsimul` specifies the number of simulations used to construct critical values (default is 2000).

Value

`Estimate`	A matrix containing grid points, estimates and confidence interval end points using p- and q-th order local polynomials as well as bias-corrected estimates and corresponding confidence intervals.
`crit_val`	The critical value used in computing the confidence interval end points.

Author(s)

Matias D. Cattaneo, Princeton University. <`cattaneo@princeton.edu`>.

Rajita Chandak (maintainer), Princeton University. <`rchandak@princeton.edu`>.

Michael Jansson, University of California Berkeley. <`mjansson@econ.berkeley.edu`>.

Xinwei Ma, University of California San Diego. <`x1ma@ucsd.edu`>.

See Also

lpcde for local polynomial conditional density estimation.

Supported methods: coef.lpcde, confint.lpcde, plot.lpcde, print.lpcde, summary.lpcde, vcov.lpcde

Examples

```
n=100
x_data = as.matrix(rnorm(n, mean=0, sd=1))
y_data = as.matrix(rnorm(n, mean=0, sd=1))
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
# density estimation
model1 = lpcde::lpcde(x_data=x_data, y_data=y_data, y_grid=y_grid, x=0, bw=0.5)
confint(model1)
```

lpbwcde *Data-driven bandwidth selection for local polynomial conditional density estimators*

Description

lpbwcde implements the bandwidth selection methods for local polynomial based conditional density (and derivatives) estimation proposed and studied in (Cattaneo et al. 2024).

Companion command: lpcde for estimation and robust bias-corrected inference.

Related Stata and R packages useful for nonparametric estimation and inference are available at https://nppackages.github.io/.

Usage

```
lpbwcde(
  y_data,
  x_data,
  x,
  y_grid = NULL,
  p = NULL,
  q = NULL,
  grid_spacing = "",
  ng = NULL,
  mu = NULL,
  nu = NULL,
  kernel_type = c("epanechnikov", "triangular", "uniform"),
  bw_type = c("imse-rot", "mse-rot"),
  regularize = NULL
)
```

Arguments

`y_data` Numeric matrix/data frame, the raw data of independent.

`x_data` Numeric matrix/data frame, the raw data of covariates.

`x` Numeric, specifies the evaluation point in the x-direction. Default is median of the dataset.

`y_grid` Numeric, specifies the grid of evaluation points. When set to default, grid points will be chosen as 0.05-0.95 percentiles of the data, with a step size of 0.05.

`p` Nonnegative integer, specifies the order of the local polynomial for `Y` used to construct point estimates. (Default is `2`.)

`q` Nonnegative integer, specifies the order of the local polynomial for `X` used to construct point estimates. (Default is `1`.)

`grid_spacing` String, If equal to "quantile" will generate quantile-spaced grid evaluation points, otherwise will generate equally spaced points.

`ng` Int, number of grid points to be used in generating bandwidth estimates.

`mu` Nonnegative integer, specifies the derivative with respect to `Y` of the distribution function to be estimated. `0` for the distribution function, `1` (default) for the density funtion, etc.

`nu` Nonnegative integer, specifies the derivative with respect to `X` of the distribution function to be estimated.

`kernel_type` String, specifies the kernel function, should be one of `"triangular"`, `"uniform"` or `"epanechnikov"`.

`bw_type` String, specifies the method for data-driven bandwidth selection. This option will be ignored if `bw` is provided. Implementable with `"mse-rot"` (default, mean squared error-optimal bandwidth selected for each grid point)

`regularize` Boolean (default TRUE). Option to regularize bandwidth selection to have atleast 20+max(p, q)+1 datapoints when evaluating the estimator.

Value

`BW` A matrix containing (1) `y_grid` (grid point), (2) `bw` (bandwidth)

`opt` A list containing options passed to the function.

Author(s)

Matias D. Cattaneo, Princeton University. <`cattaneo@princeton.edu`>.

Rajita Chandak (maintainer), Princeton University. <`rchandak@princeton.edu`>.

Michael Jansson, University of California Berkeley. <`mjansson@econ.berkeley.edu`>.

Xinwei Ma, University of California San Diego. <`x1ma@ucsd.edu`>.

References

Cattaneo MD, Chandak R, Jansson M, Ma X (2024). “Local Polynomial Conditional Density Estimators.” *Bernoulli*.

See Also

Supported methods: `coef.lpbwcde`, `print.lpbwcde`, `summary.lpbwcde`.

Examples

```
# Generate a random sample
set.seed(42);
x_data = rnorm(2000)
y_data = rnorm(2000, mean=x_data)
x = 0

# Construct bandwidth
bw1 <- lpbwcde(y_data = y_data, x_data = x_data, x=x, bw_type = "mse-rot")
summary(bw1)

# Display bandwidths for a subset of y_grid points
summary(bw1, y_grid=bw1$BW[2:5, "y_grid"])
```

---

lpcde *Local polynomial conditional density estimation*

Description

lpcde implements the local polynomial regression based conditional density (and derivatives). The estimator proposed in (Cattaneo et al. 2024). Robust bias-corrected inference methods, both pointwise (confidence intervals) and uniform (confidence bands), are also implemented.

Usage

```
lpcde(
  x_data,
  y_data,
  y_grid = NULL,
  x = NULL,
  bw = NULL,
  p = NULL,
  q = NULL,
  p_RBC = NULL,
  q_RBC = NULL,
  mu = NULL,
  nu = NULL,
  rbc = TRUE,
  ng = NULL,
  cov_flag = c("full", "diag", "off"),
  normalize = FALSE,
  nonneg = FALSE,
  grid_spacing = "",
  kernel_type = c("epanechnikov", "triangular", "uniform"),
  bw_type = NULL
)
```

Arguments

x_data	Numeric matrix/data frame, the raw data of covariates.

y_data Numeric matrix/data frame, the raw data of independent.

y_grid Numeric, specifies the grid of evaluation points in the y-direction. When set to default, grid points will be chosen as 0.05-0.95 percentiles of the data, with a step size of 0.05 in y-direction.

x Numeric, specifies the grid of evaluation points in the x-direction. When set to default, the evaluation point will be chosen as the median of the x data. To generate estimates for multiple conditioning values, please loop over the x values and evaluate the lpcde function at each point.

bw Numeric, specifies the bandwidth used for estimation. Can be (1) a positive scalar (common bandwidth for all grid points); or (2) a positive numeric vector/matrix specifying bandwidths for each grid point (should be the same dimension as grid).

p Nonnegative integer, specifies the order of the local polynomial for Y used to construct point estimates. (Default is 2.)

q Nonnegative integer, specifies the order of the local polynomial for X used to construct point estimates. (Default is 1.)

p_RBC Nonnegative integer, specifies the order of the local polynomial for Y used to construct bias-corrected point estimates. (Default is p+1.)

q_RBC Nonnegative integer, specifies the order of the local polynomial for X used to construct bias-corrected point estimates. (Default is q+1.)

mu Nonnegative integer, specifies the derivative with respect to Y of the distribution function to be estimated. 0 for the distribution function, 1 (default) for the density funtion, etc.

nu Nonnegative integer, specifies the derivative with respect to X of the distribution function to be estimated. Default value is 0.

rbc Boolean. TRUE (default) for rbc calcuations, required for valid uniform inference.

ng Int, number of grid points to be used. generates evenly space points over the support of the data.

cov_flag String, specifies covariance computation. Must be one of "full" (default), "diag" or "off".

normalize Boolean, False (default) returns original estimator, True normalizes estimates to integrate to 1.

nonneg Boolean, False (default) returns original estimator, True returns maximum of estimate and 0.

grid_spacing String, If equal to "quantile" will generate quantile-spaced grid evaluation points, otherwise will generate equally spaced points.

kernel_type String, specifies the kernel function, should be one of "triangular", "uniform", and "epanechnikov"(default).

bw_type String, specifies the method for data-driven bandwidth selection. This option will be ignored if bw is provided. Implementable with "mse-dpi" (default, mean squared error-optimal bandwidth selected for each grid point)

Details

Bias correction is only used for the construction of confidence intervals/bands, but not for point estimation. The point estimates, denoted by est, are constructed using local polynomial estimates of order p and q, while the centering of the confidence intervals/bands, denoted by est_RBC, are constructed using local polynomial estimates of order p_RBC and q_RBC. The confidence intervals/bands take the form: [est_RBC - cv * SE(est_RBC) , est_RBC + cv * SE(est_RBC)], where cv denotes the appropriate critical value and SE(est_RBC) denotes an standard error estimate for the centering of the confidence interval/band. As a result, the confidence intervals/bands may not be centered at the point estimates because they have been bias-corrected. Setting p_RBC equal to p and q_RBC to q, results on centered at the point estimate confidence intervals/bands, but requires undersmoothing for valid inference (i.e., (I)MSE-optimal bandwdith for the density point estimator cannot be used). Hence the bandwidth would need to be specified manually when q=p, and the point estimates will not be (I)MSE optimal. See Cattaneo, Jansson and Ma (2020a, 2020b) for details, and also Calonico, Cattaneo, and Farrell (2018, 2020) for robust bias correction methods.

Sometimes the density point estimates may lie outside of the confidence intervals/bands, which can happen if the underlying distribution exhibits high curvature at some evaluation point(s). One possible solution in this case is to increase the polynomial order p or to employ a smaller bandwidth.

Value

Estimate	A matrix containing (1) grid (grid points), (2) bw (bandwidths), (3) est (point estimates with p-th and q-th order local polynomial), (4) est_RBC (point estimates with p_RBC-th and q_RBC-th order local polynomial), (5) se (standard error corresponding to est. Set to NA if cov_flag="off"). (6) se_RBC (standard error corresponding to est_RBC). Set to NA if cov_flag="off"
CovMat	The variance-covariance matrix corresponding to est. Will be 0 if cov_flag="off" or a diagonal matrix if cov_flag="diag".
opt	A list containing options passed to the function.

Author(s)

Matias D. Cattaneo, Princeton University. <cattaneo@princeton.edu>.

Rajita Chandak (maintainer), Princeton University. <rchandak@princeton.edu>.

Michael Jansson, University of California Berkeley. <mjansson@econ.berkeley.edu>.

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>.

References

Cattaneo MD, Chandak R, Jansson M, Ma X (2024). “Local Polynomial Conditional Density Estimators.” *Bernoulli*.

Calonico S, Cattaneo MD, Farrell MH (2018). “On the Effect of Bias Estimation on Coverage Accuracy in Nonparametric Inference.” *Journal of the American Statistical Association*, **113**(522), 767–779.

Calonico S, Cattaneo MD, Farrell MH (2022). “Coverage Error Optimal Confidence Intervals for Local Polynomial Regression.” *Bernoulli*, **28**(4), 2998–3022.

Cattaneo MD, Jansson M, Ma X (2020). “Simple local polynomial density estimators.” *J. Amer. Statist. Assoc.*, **115**(531), 1449–1455.

See Also

Supported methods: coef.lpcde, confint.lpcde, plot.lpcde, print.lpcde, summary.lpcde, vcov.lpcde

Examples

```
#Density estimation example
n=500
x_data = matrix(rnorm(n, mean=0, sd=1))
y_data = matrix(rnorm(n, mean=x_data, sd=1))
y_grid = seq(from=-1, to=1, length.out=5)
model1 = lpcde::lpcde(x_data=x_data, y_data=y_data, y_grid=y_grid, x=0, bw=0.5)
#summary of estimation
summary(model1)
```

---

mvec *Polynomial order vector*

Description

Generates list of all combinations of length less than or equal to d of numbers that add up to n.

Usage

```
mvec(n, d)
```

Arguments

n	Total value of each combination
d	Maximum length of combinations

---

plot.lpcde *Plot method for local polynomial density conditional estimation*

Description

The plot method for local polynomial density objects. A standard ggplot2 object is returned, hence can be used for further customization.

Usage

```
## S3 method for class 'lpcde'
plot(
  ...,
  alpha = NULL,
  type = NULL,
  lty = NULL,
  lwd = NULL,
  lcol = NULL,
  pty = NULL,
  pwd = NULL,
  pcol = NULL,
  y_grid = NULL,
  CItype = NULL,
  CIuniform = FALSE,
  CIsimul = 2000,
  CIshade = NULL,
  CIcol = NULL,
  title = NULL,
  xlabel = NULL,
  ylabel = NULL,
  legendTitle = NULL,
  legendGroups = NULL,
  rbc = FALSE
)
```

Arguments

`...` Class "lpcde" object, obtained from calling `lpcde`.

`alpha` Numeric scalar between 0 and 1, specifies the significance level for plotting confidence intervals/bands.

`type` String, one of `"line"` (default), `"points"` and `"both"`, specifies how the point estimates are plotted. If more than one is provided, they will be applied to each data series accordingly.

`lty` Line type for point estimates, only effective if `type` is `"line"` or `"both"`. `1` for solid line, `2` for dashed line, `3` for dotted line. For other options, see the instructions for `ggplot2` . If more than one is provided, they will be applied to each data series accordingly.

`lwd` Line width for point estimates, only effective if `type` is `"line"` or `"both"`. Should be strictly positive. For other options, see the instructions for `ggplot2` . If more than one is provided, they will be applied to each data series accordingly.

`lcol` Line color for point estimates, only effective if `type` is `"line"` or `"both"`. `1` for black, `2` for red, `3` for green, `4` for blue. For other options, see the instructions for `ggplot2` . If more than one is provided, they will be applied to each data series accordingly.

`pty` Scatter plot type for point estimates, only effective if `type` is `"points"` or `"both"`. For options, see the instructions for `ggplot2` . If more than one is provided, they will be applied to each data series accordingly.

`pwd` Scatter plot size for point estimates, only effective if `type` is `"points"` or `"both"`. Should be strictly positive. If more than one is provided, they will be applied to each data series accordingly.

`pcol` Scatter plot color for point estimates, only effective if `type` is `"points"` or `"both"`. `1` for black, `2` for red, `3` for green, `4` for blue. For other options, see the instructions for `ggplot2` . If more than one is provided, they will be applied to each data series accordingly.

`y_grid` Numeric vector, specifies a subset of grid points to plot point estimates. This option is effective only if `type` is `"points"` or `"both"`; or if `CItype` is `"ebar"` or `"all"`.

`CItype` String, one of `"region"` (shaded region, default), `"line"` (dashed lines), `"ebar"` (error bars), `"all"` (all of the previous) or `"none"` (no confidence region), how the confidence region should be plotted. If more than one is provided, they will be applied to each data series accordingly.

`CIuniform` TRUE or FALSE (default), plotting either pointwise confidence intervals (FALSE) or uniform confidence bands (TRUE).

`CIsimul` Positive integer, specifies the number of simulations used to construct critical values (default is 2000). This option is ignored if CIuniform=FALSE.

`CIshade` Numeric, specifies the opaqueness of the confidence region, should be between 0 (transparent) and

1. Default is 0.2. If more than one is provided, they will be applied to each data series accordingly.

`CIcol` Color of the confidence region. 1 for black, 2 for red, 3 for green, 4 for blue. For other options, see the instructions for ggplot2 . If more than one is provided, they will be applied to each data series accordingly.

`title`, `xlabel`, `ylabel`
Strings, specifies the title of the plot and labels for the x- and y-axis.

`legendTitle` String, specifies the legend title.

`legendGroups` String vector, specifies the group names used in legend.

`rbc` TRUE or FALSE (default), plotting confidence intervals and bands with standard estimates (FALSE) or RBC estimates (TRUE).

Value

`Figure` A standard ggplot2 object is returned, hence can be used for further customization.

Author(s)

Matias D. Cattaneo, Princeton University. <cattaneo@princeton.edu>.

Rajita Chandak (maintainer), Princeton University. <rchandak@princeton.edu>

Michael Jansson, University of California Berkeley. <mjansson@econ.berkeley.edu>.

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>.

See Also

lpcde for local polynomial density estimation. Supported methods: coef.lpcde, confint.lpcde, plot.lpcde, print.lpcde, summary.lpcde, vcov.lpcde

---

poly_base *Polynomial basis vector expansion*

Description

Generate polynomial basis vector up to order p. has multivariate functionality as described in the main paper normalized by factorials in denominator. NOTE: currently works only up to 4th degree polynomial expansion for multivariate x.

Usage

```
poly_base(x, p)
```

Arguments

x	A number or vector.
p	A number (integer).

Value

Polynomial basis of x up to degree p.

Examples

```
poly_base(x = 2, p = 5)
```

print.lpbwcde *Print method for local polynomial conditional density bandwidth selection*

Description

The print method for local polynomial conditional density bandwidth selection objects.

Usage

```
## S3 method for class 'lpbwcde'
print(x, ...)
```

Arguments

x	Class "lpbwcde" object, obtained by calling lpbwcde.
...	Other arguments.

Value

Display output	A list of specified options provided to the function.

Author(s)

Matias D. Cattaneo, Princeton University. <cattaneo@princeton.edu>.

Rajita Chandak (maintainer), Princeton University. <rchandak@princeton.edu>.

Michael Jansson, University of California Berkeley. <mjansson@econ.berkeley.edu>.

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>.

See Also

lpbwcde for data-driven bandwidth selection.

Supported methods: coef.lpbwcde, print.lpbwcde, summary.lpbwcde.

Examples

```
n=100
x_data = as.matrix(rnorm(n, mean=0, sd=1))
y_data = as.matrix(rnorm(n, mean=0, sd=1))
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
# bandwidth selection
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
model2 = lpcde::lpbwcde(y_data=y_data, x_data=x_data, x=0, y_grid = y_grid, bw_type = "mse-rot")
print(model2)
```

print.lpcde *Print method for local polynomial conditional density estimation*

Description

The print method for local polynomial conditional density objects.

Usage

```
## S3 method for class 'lpcde'
print(x, ...)
```

Arguments

`x`	Class "lpcde" object, obtained from calling `lpcde`.
`...`	Additional options.

Value

`Display output`	summary of inputs to `lpcde`

Author(s)

Matias D. Cattaneo, Princeton University. <`cattaneo@princeton.edu`>.

Rajita Chandak (maintainer), Princeton University. <`rchandak@princeton.edu`>.

Michael Jansson, University of California Berkeley. <`mjansson@econ.berkeley.edu`>.

Xinwei Ma, University of California San Diego. <`x1ma@ucsd.edu`>.

See Also

`lpcde` for local polynomial conditional density estimation. Supported methods: `coef.lpcde`, `confint.lpcde`, `plot.lpcde`, `print.lpcde`, `summary.lpcde`, `vcov.lpcde`

Examples

```
n=100
x_data = as.matrix(rnorm(n, mean=0, sd=1))
y_data = as.matrix(rnorm(n, mean=0, sd=1))
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
# density estimation
model1 = lpcde::lpcde(x_data=x_data, y_data=y_data, y_grid=y_grid, x=0, bw=0.5)
print(model1)
```

summary.lpbwcde *Summary method for local polynomial conditional density bandwidth selection*

Description

The summary method for local polynomial conditional density bandwidth selection objects.

Usage

```
## S3 method for class 'lpbwcde'
summary(object, ...)
```

Arguments

object	Class "lpbwcde" object, obtained by calling lpbwcde.
...	Additional options, including (i) y_grid specifies a subset of y_grid points to display the bandwidth; (ii) gridIndex specifies the indices of y_grid points to display the bandwidth.

Value

Display output	A list of specified options and a matrix of grid points, bandwidth, and effective sample size.

Author(s)

Matias D. Cattaneo, Princeton University. <cattaneo@princeton.edu>.

Rajita Chandak (maintainer), Princeton University. <rchandak@princeton.edu>.

Michael Jansson, University of California Berkeley. <mjansson@econ.berkeley.edu>.

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>.

See Also

lpbwcde for data-driven bandwidth selection.

Supported methods: coef.lpbwcde, print.lpbwcde, summary.lpbwcde.

Examples

```
n=100
x_data = as.matrix(rnorm(n, mean=0, sd=1))
y_data = as.matrix(rnorm(n, mean=0, sd=1))
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
# bandwidth selection
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
model2 = lpcde::lpbwcde(y_data=y_data, x_data=x_data, x=0, y_grid = y_grid, bw_type = "mse-rot")
summary(model2)
```

---

summary.lpcde *Summary method for local polynomial density conditional estimation*

Description

The summary method for local polynomial conditional density objects.

Usage

```
## S3 method for class 'lpcde'
summary(object, ...)
```

Arguments

`object`	Class "lpcde" object, obtained from calling `lpcde`.
`...`	Additional options, including (i)`y_grid` specifies a subset of grid points in y-directions to display results; (ii) `gridIndex` specifies the indices of grid points to display results; (iii) `alpha` specifies the significance level; (iv) `CIuniform` specifies whether displaying pointwise confidence intervals (`FALSE`, default) or the uniform confidence band (`TRUE`); (v) `CIsimul` specifies the number of simulations used to construct critical values (default is `2000`).

Value

`Display output`	A list of specified options and a matrix of grid points and estimates.

Author(s)

Matias D. Cattaneo, Princeton University. <cattaneo@princeton.edu>.

Rajita Chandak (maintainer), Princeton University. <rchandak@princeton.edu>

Michael Jansson, University of California Berkeley. <mjansson@econ.berkeley.edu>.

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>.

See Also

`lpcde` for local polynomial conditional density estimation. Supported methods: `coef.lpcde`, `confint.lpcde`, `plot.lpcde`, `print.lpcde`, `summary.lpcde`, `vcov.lpcde`

Examples

```
n=100
x_data = as.matrix(rnorm(n, mean=0, sd=1))
y_data = as.matrix(rnorm(n, mean=0, sd=1))
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
# density estimation
model1 = lpcde::lpcde(x_data=x_data, y_data=y_data, y_grid=y_grid, x=0, bw=0.5)
summary(model1)
```

---

vcov.lpcde *Variance-Covariance*

Description

The vcov method for local polynomial conditional density objects.

Usage

```
## S3 method for class 'lpcde'
vcov(object, ...)
```

Arguments

object	Class "lpdensity" object, obtained by calling lpcde.
...	Additional options.

Details

Vcov method for local polynomial density conditional estimation

Value

stdErr	A matrix containing grid points and standard errors using p- and q-th order local polynomials.
CovMat	The variance-covariance matrix corresponding to est.
CovMat_RBC	The variance-covariance matrix corresponding to est_RBC.

Author(s)

Matias D. Cattaneo, Princeton University. <cattaneo@princeton.edu>.

Rajita Chandak (maintainer), Princeton University. <rchandak@princeton.edu>.

Michael Jansson, University of California Berkeley. <mjansson@econ.berkeley.edu>.

Xinwei Ma, University of California San Diego. <x1ma@ucsd.edu>.

See Also

lpcde for local polynomial conditional density estimation.

Supported methods: plot.lpcde, print.lpcde, summary.lpcde,

Examples

```
n=100
x_data = as.matrix(rnorm(n, mean=0, sd=1))
y_data = as.matrix(rnorm(n, mean=0, sd=1))
y_grid = stats::quantile(y_data, seq(from=0.1, to=0.9, by=0.1))
# density estimation
model1 = lpcde::lpcde(x_data=x_data, y_data=y_data, y_grid=y_grid, x=0, bw=0.5)
vcov(model1)
```

Index