

Package: debiasedInference (via r-universe)

September 30, 2026

Type Package

Title Bootstrap Inference with Debiased Nonparametric Estimators

Version 0.1.1

Description Implements debiased kernel density and local-linear regression estimators with empirical-bootstrap simultaneous confidence bands, as proposed by Cheng and Chen (2019) [doi:10.1214/19-EJS1575](https://doi.org/10.1214/19-EJS1575). Also provides bandwidth selectors and grid-based confidence sets for density level sets and inverse regression.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/mathcg/debiased-inference>,
<https://doi.org/10.1214/19-EJS1575>

BugReports <https://github.com/mathcg/debiased-inference/issues>

Repository <https://mathcg.r-universe.dev>

Date/Publication 2026-09-19 13:59:49 UTC

RemoteUrl <https://github.com/mathcg/debiased-inference>

RemoteRef HEAD

RemoteSha 0be716529dc2b9dac069c8ce4bac88482f3021c0

RemoteSubdir r

Contents

debiased_kde	2
debiased_local_linear	3
density_bandwidth	4
level_set	4
print.di_estimate	6

Index[7](#)

debiased_kde	<i>Debiased kernel density estimation and confidence bands</i>
--------------	--

Description

Evaluate the Gaussian debiased KDE in equation (3) of Cheng and Chen or construct its empirical-bootstrap simultaneous confidence band (Figure 2).

Usage

```
debiased_kde(x, points = NULL, bandwidth = NULL, tau = 1,
             grid_size = 200L)

kde_confidence_band(x, points = NULL, bandwidth = NULL, tau = 1,
                    confidence = 0.95, n_boot = 999L, studentized = FALSE,
                    random_state = NULL, grid_size = 200L)
```

Arguments

x	Numeric observations or a numeric matrix with observations in rows.
points	Optional evaluation vector or matrix; required for multivariate data.
bandwidth	Positive isotropic bandwidth selected for the ordinary KDE.
tau	Positive ratio h/b ; the paper recommends one.
grid_size	Generated grid size for one-dimensional data.
confidence	Confidence level strictly between zero and one.
n_boot	Positive number of empirical-bootstrap replicates.
studentized	Whether to use the variable-width band in Remark 1.
random_state	Optional local random seed; the caller's RNG state is preserved.

Value

debiased_kde() returns a di_estimate; kde_confidence_band() returns a di_band.

References

Cheng, G. and Chen, Y.-C. (2019). Nonparametric Inference via Bootstrapping the Debiased Estimator. *Electronic Journal of Statistics*, 13(1). doi:[10.1214/19EJS1575](https://doi.org/10.1214/19EJS1575).

Examples

```
x <- rnorm(50)
fit <- debiased_kde(x, bandwidth = 0.4)
band <- kde_confidence_band(x, bandwidth = 0.4, n_boot = 19,
                           random_state = 1)
```

debiased_local_linear *Debiased local-linear regression and confidence bands*

Description

Fit the debiased local-linear estimator in equation (5) of Cheng and Chen or construct its paired-bootstrap simultaneous confidence band (Figure 3).

Usage

```
debiased_local_linear(x, y, points = NULL, bandwidth = NULL, tau = 1,
  grid_size = 200L, n_folds = 5L, random_state = 0L)

regression_confidence_band(x, y, points = NULL, bandwidth = NULL,
  tau = 1, confidence = 0.95, n_boot = 999L, random_state = NULL,
  grid_size = 200L, n_folds = 5L, max_attempts = NULL)
```

Arguments

x	Numeric one-dimensional covariates.
y	Numeric responses of the same length as x.
points	Optional numeric evaluation grid.
bandwidth	Positive bandwidth selected for the ordinary smoother.
tau	Positive ratio h/b ; the paper recommends one.
grid_size	Generated grid size when points is omitted.
n_folds	Number of cross-validation folds for bandwidth selection.
random_state	Optional local random seed.
confidence	Confidence level strictly between zero and one.
n_boot	Positive number of paired-bootstrap replicates.
max_attempts	Maximum resamples attempted if singular fits occur.

Value

A di_estimate or di_band object.

References

Cheng, G. and Chen, Y.-C. (2019). Nonparametric Inference via Bootstrapping the Debiased Estimator. *Electronic Journal of Statistics*, 13(1). doi:[10.1214/19EJS1575](https://doi.org/10.1214/19EJS1575).

Examples

```
x <- seq(-1, 1, length.out = 50)
y <- sin(pi * x) + rnorm(50, sd = 0.1)
fit <- debiased_local_linear(x, y, bandwidth = 0.3)
band <- regression_confidence_band(x, y, bandwidth = 0.3,
  n_boot = 19, random_state = 1)
```

density_bandwidth	<i>Bandwidth selectors for ordinary nonparametric estimators</i>
-------------------	--

Description

Select the bandwidth on the ordinary estimator, as required by the debiasing method. Density estimation uses a normal-reference rule and regression uses K-fold cross-validation of the ordinary local-linear smoother. In one dimension, the density rule is $(4/(3n))^{1/5} \text{sd}(x)$, matching the paper's use of the normal-scale selector from the **ks** package.

Usage

```
density_bandwidth(x, method = "normal_reference", candidates = NULL,
  block_size = 512L)

regression_bandwidth(x, y, candidates = NULL, n_folds = 5L,
  random_state = 0L)
```

Arguments

x	Numeric observations or covariates.
y	Numeric responses.
method	For density estimation, either "normal_reference" or "cv".
candidates	Optional positive candidate bandwidths for cross-validation.
block_size	Positive pairwise-computation block size for density cross-validation.
n_folds	Number of cross-validation folds.
random_state	Optional local random seed.

Value

A positive numeric scalar.

level_set	<i>Grid-based level-set utilities</i>
-----------	---------------------------------------

Description

Estimate equality level sets on a one-dimensional or rectangular two- dimensional grid, invert a simultaneous band to obtain a confidence set, or calculate finite-point-cloud Hausdorff distance.

Usage

```

level_set(estimate, level)

invert_confidence_band(band, level)

hausdorff_distance(a, b)

density_level_set(x, level, points = NULL, bandwidth = NULL, tau = 1,
  grid_size = 200L)

density_level_set_confidence(x, level, points = NULL, bandwidth = NULL,
  tau = 1, confidence = 0.95, n_boot = 999L, method = "hausdorff",
  random_state = NULL, grid_size = 200L, max_attempts = NULL)

inverse_regression(x, y, level, points = NULL, bandwidth = NULL,
  tau = 1, grid_size = 200L, n_folds = 5L, random_state = 0L)

inverse_regression_confidence(x, y, level, points = NULL,
  bandwidth = NULL, tau = 1, confidence = 0.95, n_boot = 999L,
  method = "hausdorff", random_state = NULL, grid_size = 200L,
  n_folds = 5L, max_attempts = NULL)

```

Arguments

estimate	A <code>di_estimate</code> object.
band	A <code>di_band</code> object.
level	Finite target level.
a	First non-empty numeric vector or point matrix.
b	Second non-empty numeric vector or point matrix.
x	Numeric observations or covariates.
y	Numeric responses for inverse regression.
points	Optional evaluation grid. Density level sets accept a vector or a complete rectangular two-dimensional point matrix; inverse regression accepts a vector.
bandwidth	Optional positive bandwidth.
tau	Positive ratio h/b .
grid_size	Generated grid size when points are omitted.
confidence	Confidence level strictly between zero and one.
n_boot	Positive number of bootstrap replicates.
method	Either "hausdorff" or "inversion"; inverse regression additionally supports "normal" for a unique crossing.
random_state	Optional local random seed.
max_attempts	Maximum resamples attempted when sets are empty.
n_folds	Cross-validation folds for a regression bandwidth.

Value

Level-set and inverse-regression functions return a `di_set`; `hausdorff_distance()` returns a non-negative scalar.

<code>print.di_estimate</code>	<i>Print debiasedInference result objects</i>
--------------------------------	---

Description

Compact summaries of estimates, confidence bands, and level sets.

Usage

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

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

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

Arguments

<code>x</code>	A result object.
<code>...</code>	Unused.

Value

The input object, invisibly.

Index

debiased_kde, [2](#)
debiased_local_linear, [3](#)
density_bandwidth, [4](#)
density_level_set (level_set), [4](#)
density_level_set_confidence
 (level_set), [4](#)

hausdorff_distance (level_set), [4](#)

inverse_regression (level_set), [4](#)
inverse_regression_confidence
 (level_set), [4](#)
invert_confidence_band (level_set), [4](#)

kde_confidence_band (debiased_kde), [2](#)

level_set, [4](#)

print.di_band (print.di_estimate), [6](#)
print.di_estimate, [6](#)
print.di_set (print.di_estimate), [6](#)

regression_bandwidth
 (density_bandwidth), [4](#)
regression_confidence_band
 (debiased_local_linear), [3](#)