# Package: WData (via r-universe)

November 3, 2026

**Title** Statistical Inference for Weighted Data

**Version** 0.1.2

**Description** Analyzes and models data subject to sampling biases. Provides functions to estimate the density and cumulative distribution functions from biased samples of continuous distributions. Includes the estimators proposed by Bhattacharyya et al. (1988) <doi:10.1080/03610928808829825> and Jones (1991) <doi:10.2307/2337020> for density, and by Cox (2005, ISBN:052184939X) and Bose and Dutta (2022) <doi:10.1007/s00184-021-00824-3> for distribution, with different bandwidth selectors. Also includes a real length-biased dataset on shrub width from Muttlak (1988) <https://www.proquest.com/openview/3dd74592e623cdbcfa6176e85bd3d390/1?cbl=18750&diss=y&pq-origsite=gscholar>.

**License** GPL-3

**URL** https://github.com/noeliasanchmrt/WData

**BugReports** https://github.com/noeliasanchmrt/WData/issues

**Depends** R (>= 3.5.0)

**Imports** bayesmeta, KScorrect, progress, Rcpp, Rdpack, stats

**Suggests** roxygen2, testthat (>= 3.0.0), vdiffr

**RdMacros** Rdpack

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**Roxygen** list(markdown = TRUE)

**Config/roxygen2/version** 8.1.1

**Repository** https://noeliasanchmrt.r-universe.dev

**Date/Publication** 2026-11-02 23:13:12 UTC

**RemoteUrl** https://github.com/noeliasanchmrt/wdata

**RemoteRef** HEAD

**RemoteSha** 1c5160ce78102712d79e1fdeea855c726eadf40e

# Contents

bw.F.BD . . . . . 2
bw.f.BGM.boot1 . . . . . 4
bw.f.BGM.boot2 . . . . . 5
bw.f.BGM.cv . . . . . 7
bw.f.BGM.rt . . . . . 9
bw.F.SBC.cv . . . . . 11
bw.F.SBC.pi . . . . . 12
bw.F.SBC.rt . . . . . 14
cdf.bd . . . . . 15
cdf.cox . . . . . 18
df.bhatta . . . . . 19
df.jones . . . . . 20
rbiased . . . . . 22
shrub.data . . . . . 25

**Index** **27**

---

`bw.F.BD` *Bose and Dutta (2022) local bandwidth selector for Bose and Dutta (2022) kernel distribution estimator*

---

## Description

This function implements the local bandwidth selector proposed by Bose and Dutta (2022) for their own kernel distribution estimator.

## Usage

```
bw.F.BD(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
  y.seq,
  cy.seq
)
```

## Arguments

| | |
|---|---|
| y | A numeric vector containing the biased sample. |
| w | A function representing the bias function to be used. It must be evaluable and positive in each point of the sample y. By default, it is set to the length-biased function. |
| y.seq | A numeric vector containing the points on which the local bandwidth is estimated. |
| cy.seq | A numeric vector representing the constants to be used in the bandwidth estimation for each point of y.seq. Alternatively, a single numeric value can be provided, which will be used for all points in the y.seq vector. |

## Details

Local bandwidths selectors are estimated using the formula:

$$\widehat{h}_{F,\mathrm{BD},C(y)}(y) = \frac{C(y)\widehat{\sigma}_w}{(nw(y))^{1/3}},$$

where $C(y)$ is a positive parameter that depends on the point $y$ and $\widehat{\sigma}_w$ is an estimation of the standard deviation of the distribution given by

$$\widehat{\sigma}_w = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1}\left[\left(\frac{1}{n}\sum_{i=1}^{n}w(Y_i)\right) - \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1}\right]}.$$

The parameter $C(y)$ is provided to the function by using the argument cy.seq, which is a vector of positive values that is used to compute the bandwidth for each point in y.seq. Alternatively, a single numeric value can be provided, which will be used for all points in the y.seq vector. The simulations carried out by Bose and Dutta (2022) suggest that choosing $C(y) = 0.25$ or $C(y) = 0.5$ provides good results in the tail region of the distribution, with tails defined as points below the 5th percentile or above the 95th percentile. On the other hand, $C(y) = 1.3$ provides good results for the remaining points. If some bandwidths are not positive, they are replaced by the mean of the neighbors.

## Value

A numeric vector containing the bandwidths for each point in y.seq.

## References

Bose A, Dutta S (2022). "Kernel based estimation of the distribution function for length biased data." *Metrika*, **85**, 269–287.

## See Also

cdf.bd

## Examples

```
bw.F.BD(shrub.data$Width, y.seq = seq(0, 1, length.out = 512), cy.seq = rep(1, 512))
```

`bw.f.BGM.boot1` *Borrajo et al. (2017) bootstrap bandwidth selector for Jones (1991) kernel density estimator*

## Description

This function computes the bandwidth selector for Jones (1991) kernel density estimator using the bias-corrected bootstrap method developed by Borrajo et al. (2017).

## Usage

```
bw.f.BGM.boot1(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine"),
  bw0 = c("RT", "PI")
)
```

## Arguments

| | |
|---|---|
| `y` | A numeric vector containing the biased sample. |
| `w` | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample `y`. By default, it is set to the length-biased function. |
| `kernel` | A character string specifying the kernel function. Available options: `"gaussian"`, `"epanechnikov"`, `"rectangular"`, `"triangular"`, `"biweight"`, `"cosine"` and `"optcosine"`. |
| `bw0` | A character string specifying the method to determine the pilot bandwidth. Options are `"RT"` for rule of thumb and `"PI"` for plug-in bandwidth. Default is `"RT"`. |

## Details

When `bw0="RT"`, the bandwidth is given by

$$\widehat{h}_{f,\mathrm{B_{RT}}} = \left( \frac{R(K^*)\widehat{\mu}_w\widehat{\widehat{\mu}}_w}{n\eta(K^*)^2 R\left(\widehat{f}^{(2)}_{J,\widehat{h}_{f,0,\mathrm{RT}}}\right)} \right)^{1/5}, \quad \text{where} \quad \widehat{h}_{f,0,\mathrm{RT}} = \frac{n^{1/5}}{n^{1/7}}\widehat{h}_{f,\mathrm{RT}}.$$

$\widehat{h}_{f,\mathrm{RT}}$ is the value returned by `bw.f.BGM.rt`. An alternative is to consider the following pilot bandwidth:

$$\widehat{h}_{f,\mathrm{B_{opt}}} = \left( \frac{R(K^*)\widehat{\mu}_w\widehat{\widehat{\mu}}_w}{n\eta(K^*)^2 R\left(\widehat{f}^{(2)}_{J,\widehat{h}_{f,0,\mathrm{opt}}}\right)} \right)^{1/5}, \quad \text{where} \quad \widehat{h}_{f,0,\mathrm{opt}} = \left( \frac{5\widehat{\mu}_w\widehat{\widehat{\mu}}_w R\left(L^{(2)}\right)}{2n\eta(L)R\left(f^{(3)}\right)} \right)^{1/7}$$

and $R(f^{(3)})$ is estimated under the assumption that $f$ is gaussian, which is implemented by setting `bw0="PI"`. The quantities $R(K^*)$ and $\eta(K^*)^2$ depend only on the kernel and are defined as

$$R(K^*) = \int_{-\infty}^{+\infty} K^*(u)^2 du \quad \text{and} \quad \eta(K^*) = \int_{-\infty}^{+\infty} u^2 K^*(u) du.$$

The estimators $\widehat{\mu}_w$ and $\widehat{\bar{\mu}}_w$ are given by

$$\widehat{\mu}_w = n \left( \sum_{i=1}^{n} \frac{1}{w(Y_i)} \right)^{-1} \quad \text{and} \quad \widehat{\bar{\mu}}_w = \frac{\widehat{\mu}_w}{n} \sum_{i=1}^{n} \frac{1}{w(Y_i)^2}.$$

## Value

The bootstrap bandwidth value.

## References

Borrajo MI, González-Manteiga W, Martínez-Miranda MD (2017). "Bandwidth selection for kernel density estimation with length-biased data." *Journal of Nonparametric Statistics*, **29**(3), 636–668.

Jones MC (1991). "Kernel density estimation for length biased data." *Biometrika*, **78**(3), 511–519.

## See Also

`df.jones`

## Examples

```
# Bandwidth value using bootstrap method with "RT" as pilot bandwidth
bw.f.BGM.boot1(y = shrub.data$Width)
# Bandwidth value using bootstrap method with "PI" as pilot bandwidth
bw.f.BGM.boot1(y = shrub.data$Width, bw0 = "PI")
```

---

`bw.f.BGM.boot2` *Borrajo et al. (2017) bootstrap bandwidth selector for Jones (1991) kernel density estimator*

---

## Description

This function computes the bandwidth selector for Jones (1991) kernel density estimator using the bias-corrected bootstrap method developed by Borrajo et al. (2017).

## Usage

```
bw.f.BGM.boot2(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine"),
  bw0 = 1/8 * n^(-1/9),
  lower = IQR(y) * n^(-0.2) * 2000^(-1),
  upper = IQR(y) * (log(n)/n)^(0.2) * 500,
  nh = 200L,
  tol = 0.1 * lower,
  from = min(y) - (sort(y)[5] - min(y)),
  to = max(y) + (max(y) - sort(y, decreasing = TRUE)[5]),
  plot = TRUE
)
```

## Arguments

| | |
|---|---|
| y | A numeric vector containing the biased sample. |
| w | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample y. By default, it is set to the length-biased function. |
| kernel | A character string specifying the kernel function. Available options: "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine" and "optcosine". |
| bw0 | The bandwidth value to be used in density. It can also be a character string specifying a bandwidth selection method. Options available can be checked in bw.nrd. By default it is set to 1/8 n^(-1/9), where n is the sample size. |
| lower | Numeric value specifying the lower bound for bandwidth selection. Default is computed based on the interquartile range (IQR) and sample size. |
| upper | Numeric value specifying the upper bound for bandwidth selection. Default is computed based on the interquartile range (IQR) and sample size. |
| nh | An integer specifying the number of points in the grid to evaluate the mean integrated squared error function. Default is 200. |
| tol | Tolerance value used to check whether the minimum found lies at the boundaries of the interval; that is, the function will return a warning if the window minimizing the cross-validation function lies within [lower, lower+tol] or [upper-tol, upper]. Default is 10% of the lower bound. |
| from | Numeric value specifying the lower bound to be used in density. Default is computed based on the range of input data. |
| to | Numeric value specifying the upper bound to be used in density. Default is computed based on the range of input data. |
| plot | Logical value indicating whether to plot the mean integrated squared error function. Default is TRUE. |

## Details

The bandwidth returned is the one minimizing $\mathrm{MISE}^*$ over a compact interval $[h_1, h_2]$ (determined by arguments `lower` and `upper`), i.e.,

$$\widehat{h}_{f,\mathrm{B}} = \arg \min_{h_f \in [h_1, h_2]} \int_{-\infty}^{+\infty} \mathrm{MSE}^* \left( \widehat{f}^*_{\mathrm{J},h_f}(y) \right) dy.$$

$\mathrm{MISE}^*$ and $\mathrm{MSE}^*$ correspond with the expression of the mean integrated squared error and the mean squared error of the bootstrap estimator $\widehat{f}^*_{\mathrm{J},h_f}$ provided by Borrajo et al. (2017).

## Value

The bootstrap bandwidth value.

## References

Borrajo MI, González-Manteiga W, Martínez-Miranda MD (2017). "Bandwidth selection for kernel density estimation with length-biased data." *Journal of Nonparametric Statistics*, **29**(3), 636–668.

Jones MC (1991). "Kernel density estimation for length biased data." *Biometrika*, **78**(3), 511–519.

## See Also

`df.jones`

## Examples

```
bw.f.BGM.boot2(shrub.data$Width, nh = 50L)
```

---

`bw.f.BGM.cv` *Cross-validation bandwidth selector for Jones (1991) kernel density estimator*

---

## Description

This function estimates the bandwidth for Jones (1991) kernel density estimator using cross-validation criteria from Guillamón et al. (1998). It iterates through a range of bandwidth values and computes the cross-validation score for each bandwidth. The bandwidth that minimizes the cross-validation function is selected as the optimal bandwidth.

## Usage

```
bw.f.BGM.cv(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
```

```
  kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine"),
  lower = IQR(y) * n^(-0.2) * 2000^(-1),
  upper = IQR(y) * (log(n))^(0.2) * n^(-0.2) * 500,
  nh = 200L,
  tol = 0.1 * lower,
  plot = TRUE
)
```

## Arguments

| | |
|---|---|
| y | A numeric vector containing the biased sample. |
| w | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample the sample y. By default, it is set to the length-biased function. |
| kernel | A character string specifying the kernel function. Available options: "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine" and "optcosine". |
| lower | Numeric value specifying the lower bound for bandwidth selection. Default is computed based on the interquartile range (IQR) and sample size. |
| upper | Numeric value specifying the upper bound for bandwidth selection. Default is computed based on the interquartile range (IQR) and sample size. |
| nh | An integer specifying the number of points to evaluate the cross-validation function. Default is 200. |
| tol | Tolerance value used to check whether the minimum found lies at the boundaries of the interval; that is, the function will return a warning if the window minimizing the cross-validation function lies within [lower, lower+tol] or [upper-tol, upper]. Default is 10% of the lower bound. |
| plot | Logical value indicating whether to plot the cross-validation function. Default is TRUE. |

## Details

The optimal bandwidth is the one that minimizes the cross-validation function, i.e.,

$$\widehat{h}_{f,\mathrm{CV}} = \arg\min_{h_f>0} \mathrm{CV}(h_f) = \arg\min_{h_f>0} \int_{-\infty}^{+\infty} \widehat{f}_{\mathrm{J},h_f}(y)^2 dy - 2\widehat{\mathbb{E}}[\widehat{f}_{\mathrm{J},h_f}].$$

It holds that

$$\int_{-\infty}^{+\infty} \widehat{f}_{\mathrm{J},h_f}(y)^2 dy = \frac{\widehat{\mu}_w^2}{n^2 h_f} \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{w(Y_i)}\frac{1}{w(Y_j)} (K \circ K)\left(\frac{Y_i - Y_j}{h_f}\right),$$

where $\circ$ denotes convolution between two functions and $\widehat{\mathbb{E}}[\widehat{f}_{\mathrm{J},h_f}]$ is computed as

$$\widehat{\mathbb{E}}[\widehat{f}_{\mathrm{J},h_f}] = \frac{\widehat{\mu}_w}{n} \sum_{i=1}^{n} \frac{1}{w(Y_i)} \left(\sum_{j\neq i} \frac{1}{w(Y_j)}\right)^{-1} \left(\sum_{j\neq i} \frac{1}{w(Y_j)} K_{h_f}(Y_i - Y_j)\right).$$

This function computes the bandwidth that minimizes the cross validation function, CV, on the interval $I$ determined by `lower` and `upper`. By default, $I$ is the one suggested by Borrajo et al. (2017):

$$I = \left[ \frac{\text{IQR}}{2000n^{1/5}}, \frac{500\text{IQR}\log(n)^{1/5}}{n^{1/5}} \right],$$

where IQR is the interquartile range.

## Value

The optimal bandwidth value based on cross-validation criteria.

## References

Borrajo MI, González-Manteiga W, Martínez-Miranda MD (2017). "Bandwidth selection for kernel density estimation with length-biased data." *Journal of Nonparametric Statistics*, **29**(3), 636–668.

Guillamón A, Navarro J, Ruiz JM (1998). "Kernel density estimation using weighted data." *Communications in Statistics - Theory and Methods*, **27**(9), 2123-2135.

Jones MC (1991). "Kernel density estimation for length biased data." *Biometrika*, **78**(3), 511–519.

## See Also

`df.jones`

## Examples

```
bw.f.BGM.cv(shrub.data$Width)
bw.f.BGM.cv(shrub.data$Width, kernel = "epanechnikov")
```

---

`bw.f.BGM.rt` *Rule of thumb bandwidth selector for Jones (1991) kernel density estimator*

---

## Description

This function computes the bandwidth selector for Jones (1991) density estimator using the rule of thumb.

## Usage

```
bw.f.BGM.rt(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine")
)
```

## Arguments

| | |
|---|---|
| y | A numeric vector containing the biased sample. |
| w | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample y. By default, it is set to the length-biased function. |
| kernel | A character string specifying the kernel function. Available options: "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine" and "optcosine". |

## Details

The bandwidth is given by

$$\widehat{h}_{f,\mathrm{RT}} = \left(\frac{8\sqrt{\pi}\widehat{\mu}_w\widehat{\bar{\mu}}_w R(K)}{3n\eta(K)^2}\right)^{1/5}\widehat{\sigma}_w,$$

where $R(K)$ and $\eta(K)$ depend only on the kernel and are defined as

$$R(K) = \int_{-\infty}^{+\infty} K(u)^2 du \quad \text{and} \quad \eta(K) = \int_{-\infty}^{+\infty} u^2 K(u) du.$$

The estimators $\widehat{\mu}_w$ and $\widehat{\bar{\mu}}_w$ are given by

$$\widehat{\mu}_w = n\left(\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1} \quad \text{and} \quad \widehat{\bar{\mu}}_w = \frac{\widehat{\mu}_w}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)^2}.$$

$\widehat{\sigma}_w$ is an estimation of the standard deviation of the distribution given by

$$\widehat{\sigma}_w = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1}\left[\left(\frac{1}{n}\sum_{i=1}^{n}w(Y_i)\right) - \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1}\right]}.$$

## Value

The rule of thumb bandwidth value.

## References

Jones MC (1991). "Kernel density estimation for length biased data." *Biometrika*, **78**(3), 511–519.

## See Also

df.jones

## Examples

```
bw.f.BGM.rt(shrub.data$Width)
bw.f.BGM.rt(shrub.data$Width, kernel = "epanechnikov")
```

`bw.F.SBC.cv` *Cross-validation bandwidth selector for Bose and Dutta (2022) kernel distribution estimator*

## Description

This function performs bandwidth selection for Bose and Dutta (2022) kernel distribution estimator using cross-validation criteria. It iterates through a range of bandwidth values and computes the cross-validation score for each bandwidth. The bandwidth that minimizes the cross-validation function is selected as the optimal bandwidth.

## Usage

```
bw.F.SBC.cv(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine"),
  lower = IQR(y) * length(y)^(-1/3) * 0.05,
  upper = IQR(y) * length(y)^(-1/3) * 5,
  nh = 200L,
  tol = 0.1 * lower,
  plot = TRUE
)
```

## Arguments

| | |
|---|---|
| `y` | A numeric vector containing the biased sample. |
| `w` | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample the sample `y`. By default, it is set to the length-biased function. |
| `kernel` | A character string specifying the kernel function. Available options: `"gaussian"`, `"epanechnikov"`, `"rectangular"`, `"triangular"`, `"biweight"`, `"cosine"` and `"optcosine"`. |
| `lower` | Numeric value specifying the lower bound for bandwidth selection. Default is computed based on the interquartile range (IQR) and sample size. |
| `upper` | Numeric value specifying the upper bound for bandwidth selection. Default is computed based on the interquartile range (IQR) and sample size. |
| `nh` | An integer specifying the number of points to evaluate the cross-validation function. Default is 200. |
| `tol` | Tolerance value used to check whether the minimum found lies at the boundaries of the interval; that is, the function will return a warning if the window minimizing the cross-validation function lies within `[lower, lower+tol]` or `[upper-tol, upper]`. Default is 10% of the lower bound. |

`plot` A logical value indicating whether to plot the cross-validation function. Default is `TRUE`.

## Details

The optimal bandwidth is obtained as the one that minimizes the cross-validation function, that is,

$$\widehat{h}_{F,\mathrm{CV}} = \arg\min_{h_F>0} \frac{1}{n}\sum_{j=1}^{n}\left(\frac{\widehat{\mu}_w}{w(Y_j)}\mathbb{I}(y \geq Y_j) - \widehat{F}_{h_F,-j}(y)\right)^2, \quad \text{with} \quad \widehat{\mu}_w = n\left(\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1}$$

and $\widehat{F}_{h_F,-j}$ is the Bose and Dutta (2022) kernel distribution estimator without the observation $Y_j$.

## Value

The optimal bandwidth based on cross-validation criteria.

## References

Bose A, Dutta S (2022). "Kernel based estimation of the distribution function for length biased data." *Metrika*, **85**, 269–287.

## See Also

`cdf.bd`

## Examples

```
bw.F.SBC.cv(shrub.data$Width)
bw.F.SBC.cv(shrub.data$Width, kernel = "epanechnikov")
```

---

`bw.F.SBC.pi` *Plug-in bandwidth selector for Bose and Dutta (2022) kernel distribution estimator*

---

## Description

This function computes the bandwidth selector for Bose and Dutta (2022) kernel distribution estimator using the plug-in method.

## Usage

```
bw.F.SBC.pi(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine")
)
```

## Arguments

| | |
|---|---|
| y | A numeric vector containing the biased sample. |
| w | A function representing the bias function to be used. It must be evaluable and positive in each point of the sample the sample y. By default, it is set to the length-biased function. |
| kernel | A character string specifying the kernel function. Available options: "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine" and "optcosine". |

## Details

The bandwidth is given by:

$$\widehat{h}_{F,\mathrm{PI}} = \left( \frac{\widehat{\mu}_w \widehat{\bar{\mu}}_w \kappa(K)}{n\eta(K)^2 R\left(\widehat{f}^{(1)}_{\mathrm{J},\widehat{h}_{F,0,\mathrm{opt}}}\right)} \right)^{1/3},$$

where $\kappa(K)$ and $\eta(K)$ depend only on the kernel and are defined as

$$\kappa(K) = \int_{-\infty}^{+\infty} 2uW(u)K(u)du \quad \text{and} \quad \eta(K) = \int_{-\infty}^{+\infty} u^2 K(u)du,$$

where $W$ is the kernel distribution function associated with the kernel density function $K$. The estimators $\widehat{\mu}_w$ and $\widehat{\bar{\mu}}_w$ are given by

$$\widehat{\mu}_w = n\left(\sum_{i=1}^{n} \frac{1}{w(Y_i)}\right)^{-1} \quad \text{and} \quad \widehat{\bar{\mu}}_w = \frac{\widehat{\mu}_w}{n}\sum_{i=1}^{n} \frac{1}{w(Y_i)^2}.$$

$\widehat{h}_{F,0,\mathrm{opt}}$ is an estimator of

$$h_{F,0,\mathrm{opt}} = \left( \frac{3\mu_w \bar{\mu}_w R\left(L^{(1)}\right)}{2n\eta(L)R\left(f^{(2)}\right)^2} \right)^{1/5},$$

where $R\left(f^{(2)}\right)$ is estimated assuming that $f$ follows a gaussian distribution and $\mu_w$ and $\bar{\mu}_w$ are estimated by $\widehat{\mu}_w$ and $\widehat{\bar{\mu}}_w$ as defined above.

## Value

The bandwidth value using the plug-in method.

## References

Bose A, Dutta S (2022). "Kernel based estimation of the distribution function for length biased data." *Metrika*, **85**, 269–287.

## See Also

cdf.bd

## Examples

```
bw.F.SBC.pi(shrub.data$Width, kernel = "epanechnikov")
```

---

`bw.F.SBC.rt` *Rule of thumb bandwidth selector for Bose and Dutta (2022) kernel distribution estimator*

---

## Description

This function computes the bandwidth selector for Bose and Dutta (2022) kernel distribution estimator using the rule of thumb.

## Usage

```
bw.F.SBC.rt(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine")
)
```

## Arguments

| | |
|---|---|
| `y` | A numeric vector containing the biased sample. |
| `w` | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample `y`. By default, it is set to the length-biased function. |
| `kernel` | A character string specifying the kernel function. Available options: `"gaussian"`, `"epanechnikov"`, `"rectangular"`, `"triangular"`, `"biweight"`, `"cosine"` and `"optcosine"`. |

## Details

The bandwidth is computed as follows:

$$\widehat{h}_{F,\mathrm{RT}} = \left( \frac{4\sqrt{\pi}\widehat{\mu}_w \widehat{\widetilde{\mu}}_w \kappa(K)}{n\eta(K)^2} \right)^{1/3} \widehat{\sigma},$$

where both $\kappa(K)$ and $\eta(K)$ depend only on the kernel and are defined as

$$\kappa(K) = \int_{-\infty}^{+\infty} 2uW(u)K(u)du \quad \text{and} \quad \eta(K) = \int_{-\infty}^{+\infty} u^2 K(u)du,$$

where $W$ is the kernel distribution function associated with the kernel density function $K$. The estimators $\widehat{\mu}_w$ and $\widehat{\bar{\mu}}_w$ are given by

$$\widehat{\mu}_w = n\left(\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1} \quad \text{and} \quad \widehat{\bar{\mu}}_w = \frac{\widehat{\mu}_w}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)^2}.$$

$\widehat{\sigma}$ is an estimation of the standard deviation of the distribution given by

$$\widehat{\sigma}_w = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1}\left[\left(\frac{1}{n}\sum_{i=1}^{n}w(Y_i)\right) - \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1}\right]}.$$

## Value

The optimal bandwidth value for Bose and Dutta (2022) kernel distribution estimator based on the rule of thumb.

## References

Bose A, Dutta S (2022). "Kernel based estimation of the distribution function for length biased data." *Metrika*, **85**, 269–287.

## See Also

cdf.bd

## Examples

```
bw.F.SBC.rt(shrub.data$Width)
bw.F.SBC.rt(shrub.data$Width, kernel = "epanechnikov")
```

---

cdf.bd *Bose and Dutta (2022) kernel distribution estimator*

---

## Description

This function computes Bose and Dutta (2022) kernel distribution estimator given a sample and the corresponding biased function.

## Usage

```
cdf.bd(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
  y.seq,
```

```
  bw = "bw.F.SBC.rt",
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
    "cosine", "optcosine"),
  from,
  to,
  nb = 512L,
  plot = TRUE,
  correction = c("none", "left", "right", "both"),
  ...
)
```

## Arguments

| | |
|---|---|
| y | A numeric vector containing the biased sample. |
| w | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample y. By default, it is set to the length-biased function. |
| y.seq | A numeric vector specifying the points where the distribution is estimated. Alternatively, from, to and nb can be used to define the evaluation points. |
| bw | The bandwidth to be used in the distribution estimation. bw can also be a character string giving a rule to choose the bandwidth. In this case, options available are bw.F.SBC.rt, bw.F.BD, bw.F.SBC.cv and bw.F.SBC.pi. Default is bw.F.SBC.rt. |
| kernel | A character string specifying the kernel function. Available options: "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine" and "optcosine". |
| from | Numeric value specifying the lower bound of the grid where the estimator is computed when y.seq is not provided. Default is computed based on the range of input data. |
| to | Numeric value specifying the upper bound of the grid where the estimator is computed when y.seq is not provided. Default is computed based on the range of input data. |
| nb | An integer specifying the number of points at which the estimator is computed when y.seq is not provided. Default is 512. |
| plot | A logical value indicating whether to plot the estimation. Default is TRUE. |
| correction | A character string specifying the boundary correction to be applied. Options are "none", "left", "right" and "both". Default is "none". |
| ... | Additional arguments to be passed to bandwidth selection functions. |

## Details

Bose and Dutta (2022) kernel distribution estimator is expressed as

$$\widehat{F}_{h_F}(y) = \frac{\widehat{\mu}_w}{n} \sum_{i=1}^{n} \frac{1}{w(Y_i)} W_{h_F}\left(y - Y_i\right), \quad \text{where} \quad \widehat{\mu}_w = n \left( \sum_{i=1}^{n} \frac{1}{w(Y_i)} \right)^{-1},$$

$h_F$ is the bandwidth, $W$ is the kernel distribution function and $W_{h_F}(u) = W(u/h_F)$. Bose and Dutta (2022) propose a truncation correction for variables with compact support $[a, b]$ for the estimator as follows:

$$\widehat{F}_{[a,b],h_F}(y) = \begin{cases} 0, & y < a \\ \dfrac{\widehat{F}_{h_F}(y) - \widehat{F}_{h_F}(a)}{\widehat{F}_{h_F}(b) - \widehat{F}_{h_F}(a)}, & a \leq y < b \\ 1, & y \geq b. \end{cases}$$

The truncation correction is also valid for variables supported on $[a, +\infty)$ or $(-\infty, b]$, replacing $\widehat{F}_{h_F}(b)$ by 1 or $\widehat{F}_{h_F}(a)$ by 0, respectively, in the above expression. This correction is implemented in the `correction` argument, which can take values "none", "left", "right" or "both". If "left", the estimator is corrected to 0 for values less than the minimum of `y.seq`; if "right", it is corrected to 1 for values greater than the maximum of `y.seq`; if "both", it applies both corrections simultaneously.

## Value

A list with the following components:

| | |
|---|---|
| `y.seq` | The points where the distribution is estimated. |
| `F.hat` | The estimated distribution values. |
| `bw` | The bandwidth value. |
| `n` | The sample size after removal of `NaN`, `Na` and `Inf`. |
| `call` | The call which produced the result. |
| `has.na` | Logical; indicates whether the original vector `y` contains any `NaN`, `Na` or `Inf`. |

## References

Bose A, Dutta S (2022). "Kernel based estimation of the distribution function for length biased data." *Metrika*, **85**, 269–287.

## See Also

`bw.F.SBC.rt`, `bw.F.BD`, `bw.F.SBC.cv`, `bw.F.SBC.pi`

## Examples

```
cdf.bd(shrub.data$Width, kernel = "epanechnikov")
cdf.bd(shrub.data$Width, bw = "bw.F.SBC.cv")
```

`cdf.cox` *Cox (2005) distribution estimator*

## Description

This function computes Cox (2005) distribution estimator given a sample and the corresponding biased function.

## Usage

```
cdf.cox(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 }
)
```

## Arguments

| | |
|---|---|
| `y` | A numeric vector containing the biased sample. |
| `w` | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample `y`. By default, it is set to the length-biased function. |

## Details

Cox (2005) distribution estimator is expressed as

$$\widehat{F}_n(y) = \frac{\widehat{\mu}_w}{n} \sum_{i=1}^{n} \frac{1}{w(Y_i)} \mathbb{I}(Y_i \leq y), \quad \text{where} \quad \widehat{\mu}_w = n \left( \sum_{i=1}^{n} \frac{1}{w(Y_i)} \right)^{-1}.$$

## Value

A function of class `ecdf`, inheriting from the `stepfun` class, and hence inheriting a `knots` method.

## References

Cox D (2005). "Some sampling problems in technology." In Hand D, Herzberg A (eds.), *Selected Statistical Papers of Sir David Cox*, volume 1, 81–92. Cambridge University Press.

## Examples

```
cdf.cox(y = shrub.data$Width)
```

`df.bhatta` *Bhattacharyya et al. (1988) density estimator*

## Description

This function computes Bhattacharyya et al. (1988) density estimator given a sample and the corresponding biased function.

## Usage

```
df.bhatta(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
  plot = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `y` | A numeric vector containing the biased sample. |
| `w` | A function representing the bias function to be used. It must be evaluable and positive in each point of the sample `y`. By default, it is set to the length-biased function. |
| `plot` | Logical indicating whether to plot the estimated density. Default is `TRUE`. |
| `...` | Additional arguments to be passed to `density` function as, for instance, `kernel` or `bw`:<br>• `kernel` A character string giving the kernel to be used. This must partially match one of "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine", with default "gaussian", and may be abbreviated to a unique prefix (single letter).<br>• `bw` The smoothing bandwidth to be used in the density estimation. `bw` can also be a character string giving a rule to choose the bandwidth. Options available can be checked in `bw.nrd`. Default is `"nrd0"`. |

## Details

Bhattacharyya et al. (1988) density estimator is computed as follows:

$$\widehat{f}_{\mathrm{B},h_g}(y) = \widehat{\mu}_w w(y)^{-1} \widehat{g}_{h_g}(y), \quad \text{where} \quad \widehat{\mu}_w = n \left( \sum_{i=1}^{n} \frac{1}{w(Y_i)} \right)^{-1},$$

and $\widehat{g}_{h_g}(y)$ is the kernel density estimate of the given data `y` using `density` function with main arguments `bw` and `kernel`.

## Value

A list with the following components:

| | |
|---|---|
| y.seq | The points where the density is estimated. |
| f.hat | The estimated density values. |
| bw | The bandwidth value. |
| n | The sample size after removal of NaN, Na and Inf. |
| call | The call which produced the result. |
| has.na | Logical; indicates whether the original vector y contains any NaN, Na or Inf. |

## References

Bhattacharyya BB, Franklin LA, Richardson GD (1988). "A comparison of nonparametric unweighted and length-biased density estimation of fibres." *Communications in Statistics - Theory and Methods*, **17**(11), 3629–3644.

## Examples

```
# Rule of thumb
df.bhatta(shrub.data$Width, bw = "nrd0")
# Cross Validation
df.bhatta(shrub.data$Width, bw = "ucv")
# Sheather & Jones
bhata_sj <- df.bhatta(shrub.data$Width, bw = "SJ-ste")
# Rectangular kernel
df.bhatta(shrub.data$Width, bw = "nrd0", kernel = "epanechnikov")
```

---

# df.jones *Jones (1991) kernel density estimator*

---

## Description

This function computes Jones (1991) kernel density estimator given a sample and the corresponding biased function.

## Usage

```
df.jones(
  y,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
  y.seq,
  bw = "bw.f.BGM.rt",
 kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight",
```

```
    "cosine", "optcosine"),
  from,
  to,
  nb = 512L,
  plot = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| y | A numeric vector containing the biased sample. |
| w | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample y. By default, it is set to the length-biased function. |
| y.seq | A numeric vector specifying the points where the density is estimated. Alternatively, from, to and nb can be used to define the evaluation points. |
| bw | The smoothing bandwidth to be used in the density estimation. bw can also be a character string giving a rule to choose the bandwidth. In this case, options available are bw.f.BGM.rt, bw.f.BGM.cv, bw.f.BGM.boot1 and bw.f.BGM.boot2. Default is bw.f.BGM.rt. |
| kernel | A character string specifying the kernel function. Available options: "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine" and "optcosine". |
| from | Numeric value specifying the lower bound of the grid where the estimator is computed when y.seq is not provided. Default is computed based on the range of input data. |
| to | Numeric value specifying the upper bound of the grid where the estimator is computed when y.seq is not provided. Default is computed based on the range of input data. |
| nb | An integer specifying the number of points at which the estimator is computed when y.seq is not provided. Default is 512. |
| plot | A logical value indicating whether to plot the density estimation. Default is TRUE. |
| ... | Additional arguments to be passed to bandwidth selection functions. |

## Details

Jones (1991) kernel density estimator is expressed as

$$\widehat{f}_{\mathrm{J},h_f}(y) = \frac{\widehat{\mu}_w}{n}\sum_{i=1}^{n}\frac{1}{w(Y_i)}K_{h_f}(y-Y_i), \quad \text{where} \quad \widehat{\mu}_w = n\left(\sum_{i=1}^{n}\frac{1}{w(Y_i)}\right)^{-1},$$

$h_f$ is the bandwidth, $K$ is the kernel density function and $K_{h_f}(u) = 1/h_f K\left(u/h_f\right)$.

## Value

A list with the following components:

| | |
|---|---|
| y.seq | The points where the density is estimated. |
| f.hat | The estimated density values. |
| bw | The bandwidth value. |
| n | The sample size after removal of NaN, Na and Inf. |
| call | The call which produced the result. |
| has.na | Logical; indicates whether the original vector y contains any NaN, Na or Inf. |

## References

Jones MC (1991). "Kernel density estimation for length biased data." *Biometrika*, **78**(3), 511–519.

## See Also

bw.f.BGM.rt, bw.f.BGM.cv, bw.f.BGM.boot1 , bw.f.BGM.boot2

## Examples

```
# Rule of thumb
df.jones(y = shrub.data$Width, kernel = "epanechnikov", bw = "bw.f.BGM.rt")
# Cross Validation
df.jones(y = shrub.data$Width, kernel = "epanechnikov", bw = "bw.f.BGM.cv")
# Bootstrap
df.jones(y = shrub.data$Width, kernel = "epanechnikov", bw = "bw.f.BGM.boot1", bw0 = "RT")
df.jones(y = shrub.data$Width, kernel = "epanechnikov", bw = "bw.f.BGM.boot1", bw0 = "PI")

df.jones(y = shrub.data$Width, kernel = "epanechnikov", bw = "bw.f.BGM.boot2", nh = 50L)
```

---

# rbiased *Generate a biased sample using Neumann (1951) acceptance-rejection method*

---

## Description

This function generates a biased sample of size n using Neumann (1951) acceptance-rejection method. The generated sample is biased according to the provided bias function w, with respect to the unbiased density function fx.

## Usage

```
rbiased(
  n,
  w = function(y) {
     ifelse(y >= 0, y, NA)
 },
  fx,
  lim = 0.01,
  plot = TRUE,
  stop = TRUE,
  shape1,
  shape2,
  location,
  scale,
  df,
  ncp,
  rate,
  df1,
  df2,
  shape,
  meanlog,
  sdlog,
  min,
  max,
  mgshape,
  mgscale,
  mgweight,
  pro,
  mean,
  sd
)
```

## Arguments

| | |
|---|---|
| n | Sample size. |
| w | A function representing the bias function applied to the data points. It must be evaluable and positive in each point of the sample y. By default, it is set to the length-biased function. |
| fx | Unbiased density function. Values allowed are Beta (beta), Cauchy (cauchy), Chi-Square (chisq), Exponential (exp), F (f), Gamma (gamma), Logistic (logis), Log Normal (lnorm), Normal (norm), Student t (t), Continuous Uniform (unif), Weibull (weibull), Mixture of gaussian distributions (mixnorm) and Mixture of gamma distributions (weighted sum of gamma densities). |
| lim | Lower and upper limits for the range where the bias is significant and, hence, where $c = \max_{y \in \mathbb{R}} w(y)/\mu_w > 0$ must be searched. |
| plot | Logical value indicating whether to generate a plot of the biased sample. Default is TRUE. |

| | |
|---|---|
| stop | Logical value indicating whether to stop when bias function can not be evaluated in a generated value. Default is TRUE. If FALSE value is discarded and a new one is generated. |
| shape1, shape2 | Additional arguments to be passed to the unbiased density function fx when set to the beta distribution. |
| df, ncp | Additional arguments to be passed to the unbiased density function fx when set to the chisq and t distributions. |
| df1, df2 | Additional arguments to be passed to the unbiased density function fx when set to the f distribution. |
| shape, rate, scale, location | Additional arguments to be passed to the unbiased density function fx when set to the cauchy, logis, exp, gamma and weibull distributions. |
| min, max | Additional arguments to be passed to the unbiased density function fx when set to the unif distribution. |
| mgshape, mgscale, mgweight | Shape, scale, and mixing weights for the gamma components when fx = "mgamma". If mgweight is omitted, equal weights are used. |
| mean, sd, pro, meanlog, sdlog | Additional arguments to be passed to the unbiased density function fx when set to the norm, mixnorm and lnorm distributions. |

## Details

This function implements Neumann (1951) acceptance-rejection method to generate a biased sample given an unbiased density function fx and a bias function w.

## Value

A numeric vector containing a biased sample from density fx and bias function w.

## References

Neumann V (1951). "Various techniques used in connection with random digits." *Notes by G. E. Forsythe, Journal of Research of the National Bureau of Standards, Applied Math Series*, **12**(3), 36–38.

## Examples

```
# Generate a length-biased sample of size 100 from an exponential distribution
rbiased(n = 100, fx = "exp", rate = 2, plot = FALSE)

# Generate a length-biased sample from a gamma distribution
rbiased(n = 100, fx = "gamma", rate = 1.5^2, shape = 1.5)

# Generate a biased sample from a gaussian distribution
custom_bias <- function(y) {
  y^2
}
rbiased(n = 100, w = custom_bias, fx = "norm", mean = 3, sd = 10, plot = TRUE)
```

```
# Generate a biased sample from a mixture of gaussian distributions
custom_bias <- function(y) {
  sqrt(abs(y)) + 5
}
rbiased(
  n = 100, w = custom_bias, fx = "mixnorm", pro = rep(1 / 3, 3), mean = c(0.25, 0.5, 0.75),
  sd = rep(0.075, 3)
)
```

---

`shrub.data` *Shrub data*

---

## Description

Dataset containing the size of the *Cercocarpus montanus* species in an ancient quarry.

## Usage

```
data(shrub.data)
```

## Format

A `data.frame` with the following variables:

| | |
|---|---|
| `Replica` | Replica identifier (I or II). |
| `Transect` | Transect identifier (1,2 or 3). |
| `Number` | Shrub/clump identifier. |
| `Intercept` | Length of the intersection of the clump of overlapping shrubs with the transect. |
| `Width` | Width between two lines tangent to the shrub and parallel to the transect. |
| `Height` | Maximum height of the shrub encountered by the transect. |
| `Stems` | Number of stems on the shrub encountered by the transect. |

## Details

During the fall semester of 1986, students in a graduate course on biological sampling techniques, taught by Lyman L. McDonald at the University of Wyoming, conducted a field study using the linear transect method to measure the size of Cercocarpus montanus shrubs in an old limestone quarry located just east of Laramie, Wyoming. In this area, rock fissures run predominantly north to south, and vegetation is denser within them. To align with this structure, a baseline was established

approximately parallel to the fissures, and six transects were placed perpendicular to this baseline, grouped into two independent replicates (I and II), each comprising three equally spaced transects.

Students walked along the transects and recorded all Cercocarpus montanus individuals intersected. For each shrub, they measured maximum height, width (defined as the greatest distance between two tangents to the shrub's contour, parallel to the transect), and the number of stems. Given the rhizomatous nature of the species and possible interconnections between neighboring shrubs, an individual was defined as a group of stems at the base separated by at least fifteen centimeters from the nearest neighbor. Additionally, the length of intersection with the transect line was recorded for each shrub cluster.

Due to the nature of the sampling method, wider shrubs had a higher probability of being intersected by a transect. As a result, the sample of shrub widths exhibits length bias. Measurements of height and number of stems are also affected by sampling bias, although the associated bias function is more complex, as it depends on the relationship between width and these other morphological features.

For further details on the sampling protocol and data structure, see Muttlak (1988).

## Source

https://www.proquest.com/docview/303721613?%20T&fromopenview=true&pq-origsite=gscholar&sourcetype=Dissertations%20

## References

Muttlak HA (1988). *Some aspects of ranked set sampling with size biased probability of selection*. Ph.D. thesis, University of Wyoming.

## Examples

```
data(shrub.data)
names(shrub.data)
str(shrub.data)
class(shrub.data)
```

# Index


∗ **datasets**
shrub.data, 25

beta, *23, 24*
bw.F.BD, 2, *16, 17*
bw.f.BGM.boot1, 4, *21, 22*
bw.f.BGM.boot2, 5, *21, 22*
bw.f.BGM.cv, 7, *21, 22*
bw.f.BGM.rt, *4*, 9, *21, 22*
bw.F.SBC.cv, 11, *16, 17*
bw.F.SBC.pi, 12, *16, 17*
bw.F.SBC.rt, 14, *16, 17*
bw.nrd, *6*, *19*

cauchy, *23, 24*
cdf.bd, *3*, *12, 13*, 15, *15*
cdf.cox, 18
chisq, *23, 24*

density, *6*, *19*
df.bhatta, 19
df.jones, *5*, *7*, *9, 10*, 20

exp, *23, 24*

f, *23, 24*

gamma, *23, 24*

knots, *18*

lnorm, *23, 24*
logis, *23, 24*

mixnorm, *23, 24*

norm, *23, 24*

rbiased, 22

shrub.data, 25
stepfun, *18*

t, *23, 24*

unif, *23, 24*

weibull, *23, 24*