

# Package ‘L2DensityGoFtest’

July 21, 2025

**Encoding** UTF-8

**Type** Package

**Title** Density Goodness-of-Fit Test

**Version** 0.6.0

**Author** Dimitrios Bagkavos [aut, cre]

**Maintainer** Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

**Description** Provides functions for the implementation of a density goodness-of-fit test, based on piecewise approximation of the L2 distance.

**Imports** fGarch, nor1mix, boot, mvtnorm

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-01-27 19:30:02 UTC

## Contents

cutoff.asymptotic . . . . .	2
cutoff.bootstrap . . . . .	3
cutoff.edgeworth . . . . .	4
hopt.be . . . . .	6
hopt.edgeworth . . . . .	7
kde . . . . .	8
Kernels . . . . .	9
NDistDens . . . . .	10
S.n . . . . .	11
S.n.Boot . . . . .	13
S.nd . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

cutoff.asymptotic	<i>Asymptotically normal critical value for the goodness-of-fit test statistic <math>\hat{S}_n(h)</math> of Bagkavos, Patil and Wood (2021)</i>
-------------------	-------------------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Implements an asymptotically normal critical value for testing the goodness-of-fit of a parametrically estimated density with the test statistic [S.n](#).

### Usage

```
cutoff.asymptotic(dist, p1, p2, sig.lev)
```

### Arguments

dist	The null distribution.
p1	Parameter 1 (vector or object) for the null distribution.
p2	Parameter 2 (vector or object) for the null distribution.
sig.lev	Significance level of the hypothesis test.

### Details

Implements the asymptotic critical value defined in Remark 1, Bagkavos, Patil and Wood (2021), equal to  $z_\alpha \sigma_{0, \theta_0}$  where  $z_\alpha$  is the  $1 - \alpha$  quantile of the normal distribution and

$$\sigma_{0, \theta_0}^2 = 2 \left( \int K^2(u) du \right) \left( \int f_0^2(x; \theta_0) dx \right).$$

### Value

A scalar, the estimate of the asymptotic critical value at the given significance level.

### Author(s)

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

### References

Bagkavos, Patil and Wood: Nonparametric goodness-of-fit testing for a continuous multivariate parametric model, (2021), under review.

### See Also

[cutoff.edgeworth](#), [cutoff.bootstrap](#)

---

cutoff.bootstrap	<i>Bootstrap critical value for the goodness-of-fit test statistic <math>\hat{S}_n(h)</math> of Bagkavos, Patil and Wood (2021)</i>
------------------	-------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Implements a bootstrap critical value for testing the goodness-of-fit of a parametrically estimated density with the test statistic  $S_n$ .

### Usage

```
cutoff.bootstrap(xin, M, sim, dist, h.use, kfun, p1, p2, sig.lev)
```

### Arguments

<code>xin</code>	A vector of data points - the available sample.
<code>M</code>	Number of bootstrap replications.
<code>sim</code>	A character string indicating the type of simulation required: "ordinary" (the default), "parametric", "balanced", "permutation", or "antithetic".
<code>dist</code>	The null distribution.
<code>h.use</code>	The test statistic bandwidth, best implemented with <a href="https://hopt.be">hopt.be</a> .
<code>kfun</code>	The kernel to use in the density estimates used in the bandwidth expression.
<code>p1</code>	Parameter 1 (vector or object) for the null distribution.
<code>p2</code>	Parameter 2 (vector or object) for the null distribution.
<code>sig.lev</code>	Significance level of the hypothesis test.

### Details

Implements the bootstrap based finite sample critical value defined in Section 2.6, Bagkavos, Patil and Wood (2021), and calculated as follows:

1. Resample the observations  $\mathcal{X} = \{X_1, \dots, X_n\}$  to obtain  $M$  bootstrap samples, denoted by  $\mathcal{X}_m^* = \{X_{1m}^*, \dots, X_{nm}^*\}$ , where for each  $m = 1, \dots, M$ ,  $\mathcal{X}_m^*$  is sampled randomly, with replacement, from  $\mathcal{X}$ . Write  $\hat{\theta} = \theta(\mathcal{X})$  for the estimator of  $\theta$  based on the original sample  $\mathcal{X}$  and, for each  $m$ , define the bootstrap estimator of  $\theta$  by  $\hat{\theta}_m^* = \theta(\mathcal{X}_m^*)$ , where  $\theta(\cdot)$  is the relevant functional for the parameter  $\theta$ .
2. For  $m = 1, \dots, M$ , use  $\mathcal{X}_m^* = \{X_{1m}^*, \dots, X_{nm}^*\}$  and  $\hat{\theta}_m^*$  from the previous step to calculate  $n\Delta^{2d}h^{-d/2}\hat{S}_{n,m}^*(h\rho)$ ,  $m = 1, \dots, M$ .
3. Calculate  $\ell_\alpha^*$  as the  $1 - \alpha$  empirical quantile of the values  $n\Delta^{2d}h^{-d/2}\hat{S}_{n,m}^*(h\rho)$ ,  $m = 1, \dots, M$ . Then  $\ell_\alpha^*$  approximately satisfies  $P^*[n\Delta^{2d}h^{-d/2}\hat{S}_{n,m}^*(h\rho) > \ell_\alpha^*] = 1 - \alpha$ , where  $P^*$  indicates the bootstrap probability measure conditional on  $\mathcal{X}$ .

### Value

A scalar, the estimate of the bootstrap critical value at the given significance level.

**Author(s)**

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

**References**

Bagkavos, Patil and Wood: Nonparametric goodness-of-fit testing for a continuous multivariate parametric model, (2021), under review.

Gao and Gijbels, Bandwidth selection in nonparametric kernel testing, pp. 1584-1594, JASA (2008)

**See Also**

[cutoff.asymptotic](#), [cutoff.edgeworth](#)

**Examples**

```
library(nor1mix)
library(boot)
SampleSize<-80
M<-1000
dist<- "normixt"
kfun<- Epanechnikov
p1 <-MW.nm2
p2 <-1
sig.lev <- 0.05

sim<-"ordinary"
## Not run:
#Run the following to compare the asymptotic and bootstrap cut-off points on 4 occasions:
for(i in 15:18)
{
  set.seed(i)
  xin<-rnormMix(SampleSize, p1)
  h.use <- hopt.be(xin)
  l.a.a<-cutoff.asymptotic( dist,  p1, p2, sig.lev )
  l.a.b<- cutoff.bootstrap(xin, M,  sim, dist, h.use,  kfun, p1, p2, sig.lev)
  #print the result of each iteration:
  cat("Asympt. cut.off= ", l.a.a, "Boot. cut.off= ", l.a.b,  "\n")
}

## End(Not run)
```

---

cutoff.edgeworth

*Critical value based on Edgeworth expansion of the size function for the density goodness-of-fit test  $\hat{S}_n(h)$  of Bagkavos, Patil and Wood (2021)*

---

**Description**

Implements the critical value for the density goodness-of-fit test [S.n](#), approximating via an Edgeworth expansion the size function of the test statistic [S.n](#).

**Usage**

```
cutoff.edgeworth(xin, dist, kfun, p1, p2, sig.lev)
```

**Arguments**

xin	A vector of data points - the available sample.
dist	The null distribution.
kfun	The kernel to use in the density estimates used in the bandwidth expression.
p1	Parameter 1 (vector or object) for the null distribution.
p2	Parameter 2 (vector or object) for the null distribution.
sig.lev	Significance level of the hypothesis test.

**Details**

Implements the critical value for the density goodness-of-fit test [S.n](#), approximating via an Edgeworth expansion the size function of the test statistic [S.n](#), given by

$$l_\alpha = z_\alpha + d_0\sqrt{h} + d_2(n\sqrt{h})^{-1}$$

where  $z_\alpha$  is the  $1 - \alpha$  quantile of the normal distribution and  $d_0 = d_1 - C_{H_0}$  and

$$d_j = (z_\alpha^2 - 1)c_j, j = 1, 2$$

with

$$c_1 = \frac{4K^{(3)}(0)\mu_2^3\nu_3}{3\sigma^3}, c_2 = \frac{\mu_3^2K^2(0)}{\sigma^3}, \mu_i = \int K^i(x) dx, i = 1, \dots$$

and

$$C_{H_0} = 2(Ef'_0(\theta_0))^2 \Delta^{-1}, \nu_i = E\{f^i(x)\} = \int f^{i+1}(x) dx, i = 1, \dots$$

This critical value is the density function equivalent to the critical value estimate obtained in the closely related regression setting in Gao and Gijbels (2008) and is suitable for finite sample implementations of the test.

**Value**

A scalar, the estimate of the critical value at the given significance level.

**Author(s)**

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

**References**

Gao and Gijbels, Bandwidth selection in nonparametric kernel testing, pp. 1584-1594, JASA (2008)

**See Also**

[cutoff.asymptotic](#), [cutoff.bootstrap](#)

---

hopt.be

*Power-optimal bandwidth for the test statistic  $\hat{S}_n(h)$*

---

**Description**

Implements an optimal, with respect to Berry-Esseen bound, bandwidth for the density goodness-of-fit test  $\hat{S}_n(h)$  of Bagkavos, Patil and Wood (2021).

**Usage**

`hopt.be(xin)`

**Arguments**

`xin`                    A vector of data points - the available sample.

**Details**

Implements the Berry-Esseen bound optimal bandwidth defined in (18), Bagkavos, Patil and Wood (2022), given by

$$h = n^{-1/2} \sqrt{\frac{\hat{\nu}_p R_4(K)}{\rho_*^2 \hat{\nu}_4 I_0(K)}},$$

where

$$\hat{\nu}_p = n^{-1} \sum_{j=1}^n \hat{f}(X_j; \hat{h}_a),$$

and  $\hat{h}_a$  is the density optimal bandwidth calculated by a reference to a parametric distribution,  $\rho_* = 1$  and

$$R_4(K) = \int K^4(x) dx.$$

**Value**

The estimate of the Berry-Esseen optimal bandwidth.

**Author(s)**

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

**References**

Bagkavos, Patil and Wood: Nonparametric goodness-of-fit testing for a continuous multivariate parametric model, (2021), under review.

**See Also**

[hopt.edgeworth](#)

---

hopt.edgeworth      *Power-optimal bandwidth for the density goodness-of-fit test S.n.*

---

**Description**

Implements the power-optimal bandwidth for density goodness-of-fit test [S.n](#) based on optimization of the test statistic's power function.

**Usage**

```
hopt.edgeworth(xin, dist, kfun, p1, p2, sig.lev)
```

**Arguments**

xin	A vector of data points - the available sample.
dist	The null distribution.
kfun	The kernel to use in the density estimates used in the bandwidth expression.
p1	Parameter 1 (vector or object) for the null distribution.
p2	Parameter 2 (vector or object) for the null distribution.
sig.lev	Significance level of the hypothesis test.

**Details**

Implements: the power-optimal bandwidth for the test statistic [S.n](#) given by

$$h = \left\{ \frac{\sqrt{2}K^{(3)}(0)}{3R(K)^{3/2}} \frac{\nu_2}{R(f)^{3/2}} \right\}^{-1/2} \left\{ \frac{n \int \Delta_n^2(x) f^2(x) dx}{\sigma^2 \{2\nu_2 R(K)\}^{1/2}} \right\}^{-3/2} .$$

This bandwidth rule is the density function equivalent bandwidth rule obtained in the closely related regression setting in Gao and Gijbels (2008) and is designed to optimize the test's power subject to keeping the size constant.

**Value**

A scalar, the estimate the power-optimal bandwidth.

**Author(s)**

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

**References**

Gao and Gijbels, Bandwidth selection in nonparametric kernel testing, pp. 1584-1594, JASA (2008)

**See Also**

[hopt.be](http://hopt.be)

---

kde

*Kernel Density Estimation*

---

**Description**

Implements the (classical) kernel density estimator, see (2.2a) in Silverman (1986).

**Usage**

```
kde(xin, xout, h, kfun)
```

**Arguments**

<code>xin</code>	A vector of data points. Missing values not allowed.
<code>xout</code>	A vector of grid points at which the estimate will be calculated.
<code>h</code>	A scalar, the bandwidth to use in the estimate, e.g. <code>bw.nrd(xin)</code>
<code>kfun</code>	Kernel function to use. Supported kernels: <a href="#">Epanechnikov</a> , <a href="#">Biweight</a> , <a href="#">Gaussian</a> , <a href="#">Rectangular</a> , <a href="#">Triangular</a> .

**Details**

The classical kernel density estimator is given by

$$\hat{f}(x; h) = n^{-1} \sum_{i=1}^n K_h(x - X_i)$$

$h$  is determined by a bandwidth selector such as Silverman's default plug-in rule.

**Value**

A vector with the density estimates at the designated points `xout`.

**Author(s)**

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>



**References**

Silverman (1986), Density Estimation for Statistics and Data Analysis, Chapman and Hall, London.

**Examples**

```
x<-seq(-5, 5,length=100)           #design points where the estimate will be calculated
plot(x, dnorm(x), type="l", xlab = "x", ylab="density") #plot true density function
SampleSize <- 100
ti<- rnorm(SampleSize)             #draw a random sample from the actual distribution

huse<-bw.nrd(ti)
arg2<-kde(ti, x, huse, Epanechnikov) #Calculate the estimate
lines(x, arg2, lty=2)              #draw the result on the graphics device.
```

---

Kernels

*Kernel functions*


---

**Description**

Implements various kernel functions, including boundary, integrated and discrete kernels for use in the definition of the nonparametric estimates

**Usage**

```
Biweight(x, ...)
Epanechnikov(x, ...)
Triangular(x, ...)
Gaussian(x, ...)
Rectangular(x, ...)
Epanechnikov2(x)
```

**Arguments**

```
x           A vector of data points where the kernel will be evaluated.
...         Further arguments.
```

**Details**

Implements the Biweight, Triangular, Gaussian, Rectangular and Epanechnikov (including the alternative version in Epanechnikov2) kernels.

**Value**

The value of the kernel at  $x$

**References**

Wand and Jones, (1996), Kernel Smoothing, Chapman and Hall, London

---

`NDistDens`*Select null distribution*

---

**Description**

Implements the selection of null distribution; to be used within the implementation of the test statistic  $S_n$

**Usage**

```
NDistDens(x, dist, p1, p2)
```

**Arguments**

<code>x</code>	A vector of data points - the available sample size.
<code>dist</code>	The null distribution.
<code>p1</code>	Argument 1 (vector or object) for the null distribution.
<code>p2</code>	Argument 2 (vector or object) for the null distribution.

**Details**

Implements the null distribution evaluation at designated points, given the parameters `p1` and `p2`.

**Value**

A vector containing the density values of the designated distribution

**Author(s)**

Dimitrios Bagkavos

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

**References**

Bagkavos, Patil and Wood: Nonparametric goodness-of-fit testing for a continuous multivariate parametric model, (2021), under review.

---

S.n *Density goodness-of-fit test statistic based on discretized L2 distance*

---

### Description

Implements the density goodness of fit test statistic  $\hat{S}_n(h)$  of Bagkavos, Patil and Wood (2021), based on aggregation of local discrepancies between the fitted parametric density and a nonparametric empirical density estimator.

### Usage

```
S.n(xin, h, dist, p1, p2)
```

### Arguments

<code>xin</code>	A vector of data points - the available sample size.
<code>h</code>	The bandwidth to use, typically the output of <code>hopt.edgeworth</code> .
<code>dist</code>	The null distribution.
<code>p1</code>	Parameter 1 (vector or object) for the null distribution.
<code>p2</code>	Parameter 2 (vector or object) for the null distribution.

### Details

Implements the test statistic used for testing the hypothesis

$$H_0 : f(x) = f_0(x, p1, p2) \text{ vs } H_a : f(x) \neq f_0(x, p1, p2).$$

This density goodness-of-fit test is based on a discretized approximation of the L2 distance. Assuming that  $n$  is the number of observations and  $g = (\max(xin) - \min(xin))/n^{-drate}$  is the number of bins in which the range of the data is split, the test statistic is:

$$S_n(h) = n\Delta^2 h^{-1/2} \sum \sum_{i \neq j} K\{(X_i - X_j)h^{-1}\} \{Y_i - f_0(X_i)\} \{Y_j - f_0(X_j)\}$$

where  $K$  is the Epanechnikov kernel implemented in this package with the [Epanechnikov](#) function. The null model  $f_0$  is specified through the `dist` argument with parameters passed through the `p1` and `p2` arguments. The test is implemented either with bandwidth `hopt.edgeworth` or with bandwidth `hopt.be` which provide the value of  $h$  needed for calculation of  $S_n(h)$  and the critical value used to determine acceptance or rejection of the null hypothesis. See the example below for an application to a real world dataset.

### Value

A vector with the value of the test statistic as well as the Delta value used for its calculation

### Author(s)

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

## References

Bagkavos, Patil and Wood: Nonparametric goodness-of-fit testing for a continuous multivariate parametric model, (2021), under review.

## See Also

[S.n.Boot](#)

## Examples

```
library(fGarch)
library(boot)
## Not run: data(EuStockMarkets)
DAX <- as.ts(EuStockMarkets[, "DAX"])
dax <- diff(log(DAX))#[, "DAX"]

# Fit a GARCH(1,1) model to dax returns:
l11<-garchFit(~ garch(1,1), data = as.ts(dax), trace = FALSE, cond.dist = "std")
# define the model innovations, to be used as input to the test statistic
xin<-l11$residuals /l11@sigma.t
# exclude smallest value - only for uniform presentation of results
#(this step can be excluded):
xin = xin[xin!= min(xin)]

#inputs for the test statistic:
#kernel function to use in implementing the statistic
#and functional estimates for optimal h:
kfun<-"epanechnikov"
a.sig<-0.05 #define the significance level
#null hypothesis is that the innovations are normally distributed:
Nulldist<-"normal"

p1<-mean(xin)
p2<- sd(xin)
#Power optimal bandwidth:
h<-hopt.edgeworth(xin, Nulldist, kfun, p1, p2, a.sig )
h.be <- hopt.be(xin)
# Edgeworth cutoff point:
cutoff<-cutoff.edgeworth(xin, Nulldist, kfun, p1, p2, a.sig )
# Bootstrap cutoff point:
cutoff.boot<-cutoff.bootstrap(xin, 100, "permutation", Nulldist, h.be, kfun, p1, p2, a.sig)
# Asympt. Norm. cutoff point:
cutoff.asympt<-cutoff.asymptotic( Nulldist, p1, p2, a.sig )

TestStatistic<-S.n(xin, h, Nulldist, p1, p2)
TestStatistic.be<-S.n(xin, h.be, Nulldist, p1, p2)

cat("L2 test statistic value with power opt. band:", TestStatistic[1],
"\nL2 test statistic value Barry-Essen bandwidth:", TestStatistic.be[1],
"\ncritical value asymptotic:", round(cutoff.asympt,3), "critical value bootstrap:",
round(cutoff.boot,3), "critical value Edgeworth:", round(cutoff,3), "\n")
#L2 test statistic value Edgeworth: 7.257444
```

```
#L2 test statistic value Berry-Esseen bandwidth: 10.97069
# critical value Asymptotically Norm.: 1.801847
# critical value Edgeworth: 2.140446
# critical value bootstrap: 6.040048
# L2 test statistic > critical value on all occasions, hence normality is rejected
## End(Not run)
```

---

S.n.Boot

*Goodness-of-Fit test statistic based on discretized L2 distance*


---

### Description

Implements the bootstrapped version of the density goodness-of-fit test  $\hat{S}_n(h)$  defined in (6) Bagkavos, Patil and Wood (2021).

### Usage

```
S.n.Boot(xin1, indices, h, dist, kfun, p1, p2)
```

### Arguments

xin1	A vector of data points to perform bootstrap on.
indices	indices to use for the bootstrap process.
h	The bandwidth to use, typically the output of <code>hopt.be</code> .
dist	The null distribution.
kfun	The kernel to use in the density estimates used in the bandwidth expression.
p1	Argument 1 (vector or object) for the null distribution.
p2	Argument 2 (vector or object) for the null distribution.

### Details

Implements the bootstrap version of the test statistic `S.n` for use in the `cutoff.bootstrap` function. This function is typically not to be called directly by the user; it is rather meant to be called indirectly through the `cutoff.bootstrap` function.

### Value

A vector of values of the test statistic.

### Author(s)

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

### References

Bagkavos, Patil and Wood: Nonparametric goodness-of-fit testing for a continuous multivariate parametric model, (2021), under review.

**See Also**[S.n](#)

S.nd

*Density goodness-of-fit test statistic based on discretized L2 distance***Description**

Implements the multivariate ( $d \geq 2$ ) density goodness of fit test statistic  $\hat{S}_n(h)$  of Bagkavos, Patil and Wood (2021), based on aggregation of local discrepancies between the fitted parametric density and a nonparametric empirical density estimator.

**Usage**

```
S.nd(xin, h, dist, p1, p2)
```

**Arguments**

<code>xin</code>	A matrix ( $n \times d$ ) of data points - the available sample with $n$ rows and $d$ columns, each column corresponds to a different coordinate axis.
<code>h</code>	The bandwidth vector to use, typically the output of <code>hopt.be</code> in each coordinate direction.
<code>dist</code>	The null distribution.
<code>p1</code>	Parameter 1 (vector or object) for the null distribution.
<code>p2</code>	Parameter 2 (vector or object) for the null distribution.

**Details**

Implements the test statistic used for testing the hypothesis

$$H_0 : f(x) = f_0(x, p1, p2) \text{ vs } H_a : f(x) \neq f_0(x, p1, p2).$$

This density goodness-of-fit test is based on a discretized approximation of the L2 distance. Assuming that  $n$  is the number of observations and  $g = (\max(xin) - \min(xin))/n^{-drate}$  is the number of bins in which the range of the data is split, the test statistic is:

$$S_n(h) = n\Delta^2 \sum_{i \neq j} K\{(X_{i1} - X_{j1})h_1^{-1}, \dots, (X_{id} - X_{jd})h_d^{-1}\} \{Y_i - f_0(X_i)\} \{Y_j - f_0(X_j)\}$$

where  $K$  is the Epanechnikov kernel implemented in this package with the [Epanechnikov](#) function. The null model  $f_0$  is specified through the `dist` argument with parameters passed through the `p1` and `p2` arguments. The test is implemented either with bandwidth [hopt.edgeworth](#) or with bandwidth [hopt.be](#) which provide the value of  $h$  needed for calculation of  $S_n(h)$  and the critical value used to determine acceptance or rejection of the null hypothesis.

**Value**

A vector with the value of the test statistic as well as the Delta value used for its calculation

**Author(s)**

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>

**References**

Bagkavos, Patil and Wood: Nonparametric goodness-of-fit testing for a continuous multivariate parametric model, (2021), under review.

**See Also**

[S.n](#)

**Examples**

```
library(mvtnorm)
sigma <- matrix(c(4,2,2,3), ncol=2)

x <- rmvnorm(n=100, mean=c(1,2), sigma=sigma)
h.be1 <- hopt.be(x[,1])
h.be2 <- hopt.be(x[,2])
h<-c(h.be1, h.be2)
Nullldist<-"normal"

S.nd(x, h, Nullldist, c(1,2), sigma)
```

# Index

Biweight, [8](#)

Biweight (Kernels), [9](#)

cutoff.asymptotic, [2](#), [4](#), [6](#)

cutoff.bootstrap, [2](#), [3](#), [6](#), [13](#)

cutoff.edgeworth, [2](#), [4](#), [4](#)

Epanechnikov, [8](#), [11](#), [14](#)

Epanechnikov (Kernels), [9](#)

Epanechnikov2 (Kernels), [9](#)

Gaussian, [8](#)

Gaussian (Kernels), [9](#)

hopt.be, [3](#), [6](#), [8](#), [11](#), [14](#)

hopt.edgeworth, [7](#), [7](#), [11](#), [14](#)

kde, [8](#)

Kernels, [9](#)

NDistDens, [10](#)

Rectangular, [8](#)

Rectangular (Kernels), [9](#)

S.n, [2](#), [5](#), [7](#), [11](#), [14](#), [15](#)

S.n.Boot, [12](#), [13](#)

S.nd, [14](#)

Triangular, [8](#)

Triangular (Kernels), [9](#)