# Package ‘bde’

October 12, 2022

**Type** Package

**Title** Bounded Density Estimation

**Version** 1.0.1.1

**Date** 2013-06-19

**Author** Guzman Santafe, Borja Calvo, Aritz Perez and Jose A. Lozano

**Maintainer** Guzman Santafe <guzman.santafe@unavarra.es>

**Depends** R (>= 2.10), shiny, ggplot2

**Imports** methods

**Description**
A collection of S4 classes which implements different methods to estimate and deal with densities in bounded domains. That is, densities defined within the interval [lower.limit, upper.limit], where lower.limit and upper.limit are values that can be set by the user.

**License** GPL-2

**LazyData** TRUE

**Collate** BoundedDensity.R KernelDensity.R Chen99Kernel.R
MicroBetaChen99Kernel.R MacroBetaChen99Kernel.R
BoundaryKernel.R NoBoundaryKernel.R NormalizedBoundaryKernel.R
Muller91BoundaryKernel.R
JonesCorrectionMuller91BoundaryKernel.R
Muller94BoundaryKernel.R
JonesCorrectionMuller94BoundaryKernel.R BernsteinPolynomials.R
Vitale.R BrVitale.R KakizawaB1.R KakizawaB2.R KakizawaB3.R
HirukawaJLNKernel.R HirukawaTSKernel.R
MacroBetaHirukawaJLNKernel.R MacroBetaHirukawaTSKernel.R
utils.R bde.R

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-06-10 14:39:25 UTC

# R topics documented:

---

bde *Generic bounded density constructor*

---

## Description

Function to access all the methods

## Usage

```
bde(dataPoints,dataPointsCache=NULL,estimator,b=length(sample)^{-2/5},
    lower.limit=0, upper.limit=1,options=NULL)
```

## Arguments

`dataPoints` Vector containing the points to be used to estimate the density.

`dataPointsCache` Points where the density has to be estimated. If omitted, 101 points equally distributed in the [lower.limit,upper.limit] interval are used

`estimator` Density estimator to be used. This has to be one of the following:

- `"betakernel"`: Chen's beta kernel density estimator
- `"vitale"`: Vitale's Bernstein polynomial based estimator

- "boundarykernel": Boundary kernel based density estimators, as proposed by Muller et al.
- "kakizawa": Kakizawa's density estimators

b
: Bandwidth to be used. Note that in the case of Vitale's estimator the m parameter is set at 1/b

lower.limit
: a numeric value for the lower limit of the bounded interval for the data

upper.limit
: a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the [lower.limit,upper.limit] interval

options
: A list containing the different options available for the estimators:

- betakernel:
  - "modified": a logical value indicating whether the modified kernel has to be used or not. False by default
  - "normalization": a string: "none", to use the original kernels, "densitywise" to use the macrobeta kernels and "kernelwise" to use the microbeta kernels. If not specified, no normalization is used
  - "mbc": a string indicating the multiplicative bias correction to be used: "none", no correction is used, "jnl" Hirukawa's JNL approach, "ts" Hirukawa's TS approach. If not specified, no correction is used
  - "c": a numeric value between 0 and 1 corresponding to the c parameter in the TS correction (it is only taken into consideration if TS correction is selected). Default value is set to 0.5
- vitale:
  - "biasreduced": a logical value. If true, Leblanc's bias reduced estimator is used; otherwise the original estimator is used. False by default
- boundarykernel:
  - "mu": numeric parameter to indicate the kind of kernel. Options are 0, for the rectangular function, 1 for Epanechnikov's kernel, 2 for the quadratic and 3 for the biquadratic. Default value is set at 1
  - "method": a string indicating the functions to be used: "Muller94" (default value), "Muller91", "Normalize" or "None"
  - "corrected": a logical value indicating whether Jones' non-negativity correction should be used. By default it is set to false
- kakizawa:
  - "method": a string indicating the function to be used "b1", "b2" or "b3" (default value).
  - "estimator": a Bounded Density estimator. See all accepted classes here with getSubclasses("BoundedDensity"). If no estimator is provided, a Muller94BoundaryKernel estimator with default parameters and the same dataPoints as those give for the Kakizawa estimator is used.
  - "gamma": in case that b1 function is used the gamma parameter is required. This parameter takes 0.5 as default value.

`beta_0.75_0.65` *Synthetic dataset from a beta distribution*

### Description

This is a synthetic generated dataset sampling a beta distribution with parameters `shape1 = 0.75` and `shape2 = 0.65`

### Usage

```
beta_0.75_0.65
```

### Format

A vector containing 10000 observations.

`beta_1_10` *Synthetic dataset from a beta distribution*

### Description

This is a synthetic generated dataset sampling a beta distribution with parameters `shape1 = 1` and `shape2 = 10`

### Usage

```
beta_1_10
```

### Format

A vector containing 10000 observations.

`beta_5_10` *Synthetic dataset from a beta distribution*

### Description

This is a synthetic generated dataset sampling a beta distribution with parameters `shape1 = 5` and `shape2 = 10`

### Usage

```
beta_5_10
```

### Format

A vector containing 10000 observations.

BoundedDensity *Class* "BoundedDensity"

## Description

This class deals with generic estimations of a bounded densities. The probability density function is approximated by providing a set of data points in a lower and upper bounded interval and their associated densities. Using this information, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

## Objects from the Class

Objects can be created by using the generator function `boundedDensity`.

## Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## Examples

```
# data points and its densities
a <- seq(0,1,0.01)
b <- dbeta(a,5,10)

# create the density model
model <- boundedDensity(x=a,densities=b)

# examples of usual functions
density(model,0.5)

distribution(model,0.2,discreteApproximation=FALSE)
distribution(model,0.2,discreteApproximation=TRUE)

# graphical representation
hist(b,freq=FALSE)
lines(model, col="red",lwd=2)
```

---

`boundedDensity` — `BoundedDensity` *generator method*

---

## Description

User friendly constructor method for `BoundedDensity` objects.

## Usage

```
boundedDensity(x,densities,lower.limit=0,upper.limit=1)
```

## Arguments

| | |
|---|---|
| `x` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. |
| `densities` | a numeric vector containing the density for each point in `x` |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See `BoundedDensity` class for more details.

BrVitale *Class* "BrVitale"

### Description

This class deals with bias reduced version of Vitale (1975) Bernstein Polynomial approximation as described in Leblanc (2009). The polynomial estimator is computed using the provided data samples. Using this polynomial estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

### Objects from the Class

Objects can be created by using the generator function `brVitale`.

### Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`m`: the order of the polynomial approximation

`M`: a numeric parameter for bias reduction. Usually this parameter is set to `m/2` since it leads to optimal MISE (mean integrated squared error) properties

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

### Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getm** See "`getm`" for details

**getM** See "`getM`" for details

### Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

### References

Vitale, R. A. (1975). A Bernstein polynomial approach to density function estimation. *tatistical Inference and Related Topics*, 2, 87-99.

Leblanc, A. (2010). A bias-reduced approach to density estimation using Bernstein polynomials. *Journal of Nonparametric Statistics*, 22(4), 459-475.

### Examples

```
# create the model
model <- brVitale(dataPoints = tuna.r, m = 25, M = 25/2)


# examples of usual functions
density(model,0.5)

distribution(model,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(model, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(model,show=TRUE,includePoints=TRUE)
```

---

## brVitale — `BrVitale` *generator method*

---

### Description

User friendly constructor method for `BrVitale` objects.

### Usage

```
brVitale(dataPoints, m=round(length(dataPoints)^(2/5)), M=NULL, dataPointsCache=NULL,
          lower.limit = 0, upper.limit = 1)
```

### Arguments

`dataPoints` a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`m` a integer value indicating the order of the polynomial approximation. `m` must take values greater than 0

M
: a numeric value indicating the parameter for bias reduction, with m > M. If M=NULL, the value m/2, which leads to optimal MISE (mean integrated squared error) properties, is taken as default

dataPointsCache
: a numeric vector containing points within the [lower.limit,upper.limit] interval. These points are used for convenience to cache density and distribution values. If dataPointsCache=NULL the values are initialized to a sequence of 101 equally spaced values from lower.limit to upper.limit

lower.limit
: a numeric value for the lower limit of the bounded interval for the data

upper.limit
: a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the [lower.limit,upper.limit] interval

## Details

See BrVitale class for more details.

---

Chen99Kernel *Class* "Chen99Kernel"

---

## Description

This class deals with Kernel estimators for bounded densities as described in Chen's 99 paper. The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

## Objects from the Class

Objects can be created by using the generator function chen99Kernel.

## Slots

**dataPointsCache:** a numeric vector containing points within the [lower.limit,upper.limit] interval

**densityCache:** a numeric vector containing the density for each point in dataPointsCache

**distributionCache:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

**dataPoints:** a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator

**b:** the bandwidth of the kernel estimator

**modified:** if TRUE, the modified version of the kernel estimator is used

**lower.limit:** a numeric value for the lower limit of the bounded interval for the data

**upper.limit:** a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "density" for details

**distribution** See "distribution" for details

**quantile** See "quantile" for details

**rsample** See "rsample" for details

**plot** See "plot" for details

**getdataPointsCache** See "getdataPointsCache" for details

**getdensityCache** See "getdensityCache" for details

**getdistributionCache** See "getdistributionCache" for details

**getdataPoints** See "getdataPoints" for details

**getb** See "getb" for details

**getmodified** See "getmodified" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Chen, S. X. (1999). Beta kernel estimators for density functions. *Computational Statistics & Data Analysis*, 31, 131-145.

## Examples

```
# create the model
kernel.noModified <- chen99Kernel(dataPoints = tuna.r, b = 0.01, modified = FALSE)
kernel.Modified <- chen99Kernel(dataPoints = tuna.r, b = 0.01, modified = TRUE)

# examples of usual functions
density(kernel.noModified,0.5)
density(kernel.Modified,0.5)

distribution(kernel.noModified,1,discreteApproximation=FALSE)
distribution(kernel.noModified,1,discreteApproximation=TRUE)

distribution(kernel.Modified,1,discreteApproximation=FALSE)
distribution(kernel.Modified,1,discreteApproximation=TRUE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Chen99 Kernels Tuna Data")
lines(kernel.noModified,col="red",lwd=2)
lines(kernel.Modified,col="blue",lwd=2)

# graphical representation using ggplot2
graph <- gplot(list("KernelNoModified"=kernel.noModified,
                "KernelModified"=kernel.Modified),show=TRUE)
```

chen99Kernel — chen99Kernel *generator method*

## Description

User friendly constructor method for Chen99Kernel objects.

## Usage

```
chen99Kernel(dataPoints, b=length(dataPoints)^(-2/5), dataPointsCache=NULL,
                modified = FALSE, lower.limit = 0, upper.limit = 1)
```

## Arguments

| | |
|---|---|
| dataPoints | a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator |
| b | the bandwidth of the kernel estimator |
| dataPointsCache | a numeric vector containing points within the [lower.limit,upper.limit] interval. These points are used for convenience to cache density and distribution values. If dataPointsCache=NULL the values are initialized to a sequence of 101 equally spaced values from lower.limit to upper.limit |
| modified | if TRUE, the modified version of the kernel estimator is used |
| lower.limit | a numeric value for the lower limit of the bounded interval for the data |
| upper.limit | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the [lower.limit,upper.limit] interval |

## Details

See Chen99Kernel class for more details.

density — *Probability Density Function (pdf)*

## Description

Density function for the given bounded density object.

## Arguments

| | |
|---|---|
| x | A bounded density estimator. See all the accepted classes here by running the command getSubclasses("BoundedDensity"). This parameter is named x instead of .Object to agree with other already defined density methods |
| values | Vector of points where the density function is evaluated. These points must be in the interval [x@lower.limit,x@upper.limit]. This parameter is named values instead of x to agree with other already defined density methods |

### Methods

`density(x,values)`

---

`distribution` *Cumulative Density Function (cdf)*

---

### Description

Distribution function for the given bounded density object

### Arguments

| | |
|---|---|
| `.Object` | A bounded density estimator. See all the accepted classes here by running the command `getSubclasses("BoundedDensity")`. |
| `x` | Vector of points where the density function is evaluated. These points must be in the interval `[.Object@lower.limit,.Object@upper.limit]` |
| `discreteApproximation` | Logical; if `TRUE` the distribution function is computed using a discrete approximation using the values cached in `dataPointsCache` and `densityCache`. Otherwise, the integral of the density function is evaluated. |

### Details

If `discreteApproximation` is not specified it assumes the default value `TRUE`. When the distribution function is used with a `BoundedDensity` object, `discreteApproximation` value is and a discrete approximation is always obtained.

### Methods

`distribution(.Object,x,discreteApproximation=TRUE)`

---

`eruption` *Eruption lengths of Old Faithful geyser*

---

### Description

The dataset comprises lengths (in minutes) of eruptions of Old Faithful geyser in Yellowstone National Park, USA. The data are within the interval [1.67,4.93].

### Usage

`eruption`

### Format

A vector containing 107 observations.

### Source

The data were obtained from Silverman (1996) Table 2.2

### References

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall

Weisberg, S. (1980). *Applied linear regression*. John Wiley & Sons, Canada

---

## getb — *Accesor method for* `b` *slot*

---

### Description

This method obtains the values stored in the `b` slot of a bounded density object. This slot contains the bandwidth parameter for the kernel estimator.

### Arguments

| | |
|---|---|
| `.Object` | A kernel density estimator. See all the accepted classes here by running the command `getSubclasses("KernelDensity")`. |

### Methods

`getb(.Object)`

---

## getc — *Accesor method for* `c` *slot*

---

### Description

This method obtains the values stored in the `c` slot of a HirukawaTSKernel object. This parameter is used in the kernel estimation as a smoothing parameter.

### Arguments

| | |
|---|---|
| `.Object` | A `HirukawaTSKernel` or a `MacroBetaHirukawaTSKernel` object. |

### Methods

`getc(.Object)`

## getdataPoints — *Accesor method for* `dataPoints` *slot*

### Description

This method obtains the values stored in the `DataPoints` slot of a bounded density object. This slot contains the data sample used to estimate the density model.

### Arguments

| | |
|---|---|
| `.Object` | A bounded density estimator. See all the accepted classes by running the commands `getSubclasses("KernelDensity")` and `getSubclasses("BernsteinPolynomials")`. |

### Methods

```
getdataPoints(.Object)
```

## getdataPointsCache — *Accesor method for* `DataPointsCache` *slot*

### Description

This method obtains the values stored in the `dataPointsCache` slot of a bounded density object.

### Arguments

| | |
|---|---|
| `.Object` | A bounded density estimator. See all the accepted classes here by running the command `getSubclasses("BoundedDensity")`. |

### Methods

```
getdataPointsCache(.Object)
```

## getdensityCache *Accesor method for* `densityCache` *slot*

### Description

This method obtains the values stored in the `DensityCache` slot of a bounded density object

### Arguments

| | |
|---|---|
| `.Object` | A bounded density estimator. See all the accepted classes here by running the command `getSubclasses("BoundedDensity")`. |

### Methods

`getdensityCache(.Object)`

## getdensityEstimator *Accesor method for* `gamma` *slot*

### Description

This method obtains the class name of the object stored in the `densityEstimator` slot of a KakizawaB1, KakizawaB2 or KakizawaB3 object.

### Arguments

| | |
|---|---|
| `.Object` | A `KakizawaB1`, `KakizawaB2` or `KakizawaB3` object. |

### Methods

`getdensityEstimator(.Object)`

## getdistributionCache *Accesor method for* `distributionCache` *slot*

### Description

This method obtains the values stored in the `DistributionCache` slot of a bounded density object.

### Arguments

| | |
|---|---|
| `.Object` | A bounded density estimator. See all the accepted classes here by running the command `getSubclasses("BoundedDensity")`. |

### Methods

`getdistributionCache(.Object)`

## `getgamma` *Accesor method for* `gamma` *slot*

### Description

This method obtains the values stored in the `gamma` slot of a KakizawaB1 object. This slot contains a parameter used in the B1 approximation using Bernstein polynomials.

### Arguments

| | |
|---|---|
| `.Object` | A `KakizawaB1` object. |

### Methods

```
getgamma(.Object)
```

## `getM` *Accesor method for* `M` *slot*

### Description

This method obtains the values stored in the `M` slot of a BrVitale object. This slot contains parameter for bias reduction.

### Arguments

| | |
|---|---|
| `.Object` | A `BrVitale` Object. |

### Methods

```
getM(.Object)
```

## getm — *Accesor method for* `m` *slot*

### Description

This method obtains the values stored in the `m` slot of a BernsteinPolynomials object. This slot contains the order of the polynomial expansion.

### Arguments

| | |
|---|---|
| `.Object` | A boundary kernel density estimator. See all the accepted classes here with `getSubclasses("BernsteinPolynomials")`. |

### Methods

`getm(.Object)`

## getmodified — *Accesor method for* `modified` *slot*

### Description

This method obtains the values stored in the `modified` slot of a Kernel density object. The value of this slot is `TRUE` if a modified version of the kernel estimator is used and `FALSE` otherwise.

### Arguments

| | |
|---|---|
| `.Object` | A kernel density estimator. See all the accepted classes here by running the command `getSubclasses("KernelDensity")`. |

### Methods

`getgetmodified(.Object)`

getmu *Accesor method for* `Mu` *slot*

## Description

This method obtains the values stored in the `mu` slot of a Boundary Kernel object. This slot contains the degree of smoothing for the boundary kernel estimator. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (bicuadratic kernel) or 3 (tricuadratic kernel).

## Arguments

| | |
|---|---|
| `.Object` | A boundary kernel density estimator. See all the accepted classes here with `getSubclasses("BoundaryKernel")`. |

## Methods

`getmu(.Object)`

getSubclasses *List of subclasses*

## Description

This method returns a list containing the name of the class given as parameter and all the subclasses. Virtual classes are excluded from the list.

## Usage

```
getSubclasses(className)
```

## Arguments

| | |
|---|---|
| `className` | a string with the name of a S4 class |

## Examples

```
# show the names of the class BoundedDensity and all its subclasses
getSubclasses("BoundedDensity")

# show the names of the class Chen99Kernel and all its subclasses
getSubclasses("Chen99Kernel")
```

## gplot — *Bounded Density Plotting based on ggplot2*

### Description

Function to plot bounded density probability density functions.

### Arguments

| | |
|---|---|
| `.Object` | A bounded density estimator or a list of bounded density estimators. See all the accepted classes here by running the command `getSubclasses("BoundedDensity")`. |
| `show` | Logical value. If `FALSE` the density of the `BoundedDensity` object in `.Object` is not plotted but only the `ggplot2` graphical object is returned. This object can be used for further modifications and plots. If `TRUE` `ggplot2` graphical object is returned and also the density is plotted. |
| `includePoints` | Logical value. It determines whether or not the point used to estimate the density (`dataPoints`) are included in the plot. Note that, in order to improve the visualization, the points are jittered in the Y axis. When the amount of points is very high, jittering is not enough; in that case, the `alpha` parameter can be used to control the transparency of the points. |
| `lwd` | Usual line width graphical parameter. See `?par` for more information |
| `alpha` | A value between 0 and 1 indicating the transparency of the points when they are included in the plot |

### Methods

`gplot(.Object,show=FALSE,includePoints=FALSE,lwd=1,alpha=1)`

### References

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer.

## HirukawaJLNKernel — *Class* "HirukawaJLNKernel"

### Description

This class deals with the JLN Kernel estimator as described in Hirukawa (2010). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

### Objects from the Class

Objects can be created by using the generator function `hirukawaJLNKernel`.

## Slots

`dataPointsCache`**:** a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`**:** a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`**:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`**:** a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`**:** the bandwidth of the kernel estimator

`modified`**:** if `TRUE`, the modified version of the kernel estimator is used

`lower.limit`**:** a numeric value for the lower limit of the bounded interval for the data

`upper.limit`**:** a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getb** See "`getb`" for details

**getmodified** See "`getmodified`" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Hirukawa, M. (2010). Nonparametric multiplicative bias correction for kernel-type density estimation on the unit interval. *Computational Statistics & Data Analysis*, 54(2), 473-495.

## Examples

```
# create the model
kernel.noModified <- hirukawaJLNKernel(dataPoints = tuna.r, b = 0.01, modified = FALSE)
kernel.Modified <- hirukawaJLNKernel(dataPoints = tuna.r, b = 0.01, modified = TRUE)

# examples of usual functions
```

```
density(kernel.noModified,0.5)
density(kernel.Modified,0.5)

distribution(kernel.noModified,1,discreteApproximation=FALSE)
distribution(kernel.noModified,1,discreteApproximation=TRUE)

distribution(kernel.Modified,1,discreteApproximation=FALSE)
distribution(kernel.Modified,1,discreteApproximation=TRUE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Chen99 Kernels Tuna Data")
lines(kernel.noModified, col="red",lwd=2)
lines(kernel.Modified,col="blue",lwd=2)

# graphical representation using ggplot2
graph <- gplot(list("noModified"=kernel.noModified,
          "modified"=kernel.Modified), show=TRUE)
```

---

`hirukawaJLNKernel` `HirukawaJLNKernel` *generator method*

---

## Description

User friendly constructor method for `HirukawaJLNKernel` objects.

## Usage

```
hirukawaJLNKernel(dataPoints, b, dataPointsCache=NULL, modified = FALSE,
                    lower.limit = 0, upper.limit = 1)
```

## Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `modified` | if `TRUE`, the modified version of the kernel estimator is used |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See `HirukawaJLNKernel` class for more details.

HirukawaTSKernel *Class* "HirukawaTSKernel"

## Description

This class deals with the TS Kernel estimator as described in Hirukawa (2010). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

## Objects from the Class

Objects can be created by using the generator function `hirukawaTSKernel`.

## Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`: the bandwidth of the kernel estimator

`modified`: if `TRUE`, the modified version of the kernel estimator is used

`c`: a numeric value between 0 and 1. This parameter is used in the TS approximation as a smoothing parameter

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getb** See "`getb`" for details

**getmodified** See "`getmodified`" for details

**getc** See "`getc`" for details

### Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

### References

Hirukawa, M. (2010). Nonparametric multiplicative bias correction for kernel-type density estimation on the unit interval. *Computational Statistics & Data Analysis*, 54(2), 473-495.

### Examples

```
# create the model
kernel.noModified <- hirukawaTSKernel(dataPoints = tuna.r, b = 0.01,
                      modified = FALSE, c = 0.5)
kernel.Modified <- hirukawaTSKernel(dataPoints = tuna.r, b = 0.01,
                      modified = TRUE, c = 0.5)

# examples of usual functions
density(kernel.noModified,0.5)
density(kernel.Modified,0.5)

distribution(kernel.noModified,1,discreteApproximation=FALSE)
distribution(kernel.noModified,1,discreteApproximation=TRUE)

distribution(kernel.Modified,1,discreteApproximation=FALSE)
distribution(kernel.Modified,1,discreteApproximation=TRUE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Chen99 Kernels Tuna Data")
lines(kernel.noModified,col="red",lwd=2)
lines(kernel.Modified,col="blue",lwd=2)

# graphical representation using ggplot2
graph <- gplot(list("noModified"=kernel.noModified,
          "modified"=kernel.Modified), show=TRUE)
```

---

## `hirukawaTSKernel` `HirukawaTSKernel` *generator method*

---

### Description

User friendly constructor method for `HirukawaTSKernel` objects.

### Usage

```
hirukawaTSKernel(dataPoints, c, b=length(dataPoints)^(-2/5), dataPointsCache=NULL,
                    modified = FALSE, lower.limit = 0, upper.limit = 1)
```

### Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `c` | a numeric value between 0 and 1. This parameter is used in the TS approximation as a smoothing parameter |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `modified` | if `TRUE`, the modified version of the kernel estimator is used |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

### Details

See `HirukawaTSKernel` class for more details.

---

`JonesCorrectionMuller91BoundaryKernel`
*Class* `"JonesCorrectionMuller91BoundaryKernel"`

---

### Description

This class deals with nonnegative boundary correction of the `muller91BoundaryKernel` estimators for bounded densities. In this normalization, two kernel functions are needed. The first kernel funciton `-K(u)-` is the kernel function used in `muller91BoundaryKernel` (using left boundary, interior or right boundary kernel functions as needed). For the second kernel function, the popular choice `L(u) = u * K(u)` is taken. The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations. Note that the renormalization of this kernel estimator guarantees nonnegative values for the density function but the cumulative density function may takes values greater than 1.

### Objects from the Class

Objects can be created by using the generator function `jonesCorrectionMuller91BoundaryKernel`.

## Slots

**dataPointsCache:** a numeric vector containing points within the [lower.limit,upper.limit] interval

**densityCache:** a numeric vector containing the density for each point in dataPointsCache

**distributionCache:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

**dataPoints:** a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator

**b:** the bandwidth of the kernel estimator

**mu:** a integer value indicating the degree of smoothness for the boundary kernel. mu can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel)

**normalizedKernel:** this slot is used to save a NormalizedBoundaryKernel object used in the normalization. It is only for internal use

**lower.limit:** a numeric value for the lower limit of the bounded interval for the data

**upper.limit:** a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "density" for details

**distribution** See "distribution" for details

**quantile** See "quantile" for details

**rsample** See "rsample" for details

**plot** See "plot" for details

**getdataPointsCache** See "getdataPointsCache" for details

**getdensityCache** See "getdensityCache" for details

**getdistributionCache** See "getdistributionCache" for details

**getdataPoints** See "getdataPoints" for details

**getb** See "getb" for details

**getmu** See "getmu" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Jones, M. C. and Foster, P. J. (1996). A simple nonnegative boundary correction method for kernel density estimation. *Statistica Sinica*, 6, 1005-1013.

Muller, H. (1991). Smooth optimum kernel estimators near endpoints. *Biometrika*, 78(3), 521-530.

### Examples

```
# create the model
kernel <-jonesCorrectionMuller91BoundaryKernel(dataPoints = tuna.r, b = 0.01, mu = 2)


# examples of usual functions
density(kernel,0.5)

distribution(kernel,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(kernel, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(kernel, show=TRUE, includePoints=TRUE)
```

---

## jonesCorrectionMuller91BoundaryKernel *JonesCorrectionMuller91BoundaryKernel generator method*

---

### Description

User friendly constructor method for `JonesCorrectionMuller91BoundaryKernel` objects.

### Usage

```
jonesCorrectionMuller91BoundaryKernel(dataPoints, mu=1, b=length(dataPoints)^(-2/5),
                                      dataPointsCache=NULL, lower.limit = 0,
                                      upper.limit = 1)
```

### Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `mu` | a integer value indicating the degree of smoothness for the boundary kernel. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel) |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See JonesCorrectionMuller91BoundaryKernel class for more details.

---

JonesCorrectionMuller94BoundaryKernel
*Class* "JonesCorrectionMuller94BoundaryKernel"

---

## Description

This class deals with nonnegative boundary correction of the muller94BoundaryKernel estimators for bounded densities. In this normalization, two kernel functions are needed. The first kernel funciton -K(u)- is the kernel function used in muller94BoundaryKernel (using left boundary, interior or right boundary kernel functions as needed). For the second kernel function, the popular choice L(u) = u * K(u) is taken. The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations. Note that the renormalization of this kernel estimator guarantees nonnegative values for the density function but the cumulative density function may takes values greater than 1.

## Objects from the Class

Objects can be created by using the generator function jonesCorrectionMuller94BoundaryKernel.

## Slots

**dataPointsCache:** a numeric vector containing points within the [lower.limit,upper.limit] interval

**densityCache:** a numeric vector containing the density for each point in dataPointsCache

**distributionCache:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

**dataPoints:** a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator

**b:** the bandwidth of the kernel estimator

**mu:** a integer value indicating the degree of smoothness for the boundary kernel. mu can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel)

**normalizedKernel:** this slot is used to save a NormalizedBoundaryKernel object used in the normalization. It is only for internal use

**lower.limit:** a numeric value for the lower limit of the bounded interval for the data

**upper.limit:** a numeric value for the upper limit of the bounded interval for the data

### Methods

**density** See "density" for details

**distribution** See "distribution" for details

**quantile** See "quantile" for details

**rsample** See "rsample" for details

**plot** See "plot" for details

**getdataPointsCache** See "getdataPointsCache" for details

**getdensityCache** See "getdensityCache" for details

**getdistributionCache** See "getdistributionCache" for details

**getdataPoints** See "getdataPoints" for details

**getb** See "getb" for details

**getmu** See "getmu" for details

### Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

### References

Jones, M. C. and Foster, P. J. (1996). A simple nonnegative boundary correction method for kernel density estimation. *Statistica Sinica*, 6, 1005-1013.

Muller, H. and Wang, J. (1994). Hazard rate estimation under random censoring with varying kernels and bandwidths. *Biometrics*, 50(1), 61-76.

### Examples

```
# data points to cache densities and distribution
cache <- seq(0,1,0.01)

# create the model
kernel <-jonesCorrectionMuller94BoundaryKernel(dataPoints = tuna.r, b = 0.01, mu = 2,
                                                dataPointsCache = cache)


# examples of usual functions
density(kernel,0.5)

distribution(kernel,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(kernel, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(kernel, show=TRUE, includePoints = TRUE)
```

## jonesCorrectionMuller94BoundaryKernel — JonesCorrectionMuller94BoundaryKernel *generator method*

### Description

User friendly constructor method for JonesCorrectionMuller94BoundaryKernel objects.

### Usage

```
jonesCorrectionMuller94BoundaryKernel(dataPoints, mu=1, b=length(dataPoints)^(-2/5),
                                      dataPointsCache=NULL, lower.limit = 0,
                                      upper.limit = 1)
```

### Arguments

| | |
|---|---|
| dataPoints | a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator |
| mu | a integer value indicating the degree of smoothness for the boundary kernel. mu can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel) |
| b | the bandwidth of the kernel estimator |
| dataPointsCache | a numeric vector containing points within the [lower.limit,upper.limit] interval. These points are used for convenience to cache density and distribution values. If dataPointsCache=NULL the values are initialized to a sequence of 101 equally spaced values from lower.limit to upper.limit |
| lower.limit | a numeric value for the lower limit of the bounded interval for the data |
| upper.limit | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the [lower.limit,upper.limit] interval |

### Details

See JonesCorrectionMuller94BoundaryKernel class for more details.

## KakizawaB1 — *Class* "KakizawaB1"

### Description

This class deals with B1 approximation to kernel density estimation as described in Kakizawa (2004). This is a Berstein polynomial approximation of the density function which uses BoundedDensity objects instead of a polynomial function. By contrast to the original Kakizawa's approach where only boundary kernels are used, here, any BoundedDensity object is allowed. Using this estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

### Objects from the Class

Objects can be created by using the generator function `kakizawaB1`.

### Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`gamma`: a numeric value between 0 and 1. This parameter is used in the B1 approximation using Bernstein polynomials

`densityEstimator`: a BoundedDensity object used to estimate the density

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

### Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getm** See "`getm`" for details

**getgamma** See "`getgamma`" for details

**getdensityEstimator** See "`getdensityEstimator`" for details

### Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

### References

Kakizawa, Y. (2004). Bernstein polynomial probability density estimation. *Journal of Nonparametric Statistics*, 16(5), 709-729.

## Examples

```
# create the model
# we use a MicroBetaChen99Kernel is used as estimator y KakizawaB1 approximation
est <- microBetaChen99Kernel(dataPoints = tuna.r, b = 0.01, modified = FALSE)
model <- kakizawaB1(dataPoints = tuna.r, m = 25, gamma = 0.25)


# examples of usual functions
density(model,0.5)

distribution(model,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(model, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(model, show=TRUE, includePoints=TRUE)
```

---

kakizawaB1 KakizawaB1 *generator method*

---

## Description

User friendly constructor method for KakizawaB1 objects.

## Usage

```
kakizawaB1(dataPoints,estimator=NULL,m=round(length(dataPoints)^(2/5)),gamma=0.5,
           dataPointsCache=NULL, lower.limit = 0, upper.limit = 1)
```

## Arguments

| | |
|---|---|
| dataPoints | a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator |
| estimator | A bounded density estimator. See all the accepted classes here with getSubclasses("BoundedDensity"). If no estimator is provided here (default value = NULL), a Muller94BoundaryKernel estimator with default parameters and the same dataPoints as those give for the Kakizawa estimator is used. |
| m | a integer value indicating the order of the polynomial approximation. m must take values greater than 0 |
| gamma | a numeric value between 0 and 1. This parameter is used in the B1 approximation using Bernstein polynomials |
| dataPointsCache | a numeric vector containing points within the [lower.limit,upper.limit] interval. These points are used for convenience to cache density and distribution values. If dataPointsCache=NULL the values are initialized to a sequence of 101 equally spaced values from lower.limit to upper.limit |

`lower.limit` a numeric value for the lower limit of the bounded interval for the data

`upper.limit` a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval

## Details

See `KakizawaB1` class for more details.

---

`KakizawaB2` *Class* `"KakizawaB2"`

---

## Description

This class deals with B2 approximation to kernel density estimation as described in Kakizawa (2004). This is a Berstein polynomial approximation of the density function which uses BoundedDensity objects instead of a polynomial function. By contrast to the original Kakizawa's approach where only boundary kernels are used, here, any BoundedDensity object is allowed. Using this estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

## Objects from the Class

Objects can be created by using the generator function `kakizawaB2`.

## Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`densityEstimator`: a BoundedDensity object used to estimate the density

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "getdataPointsCache" for details

**getdensityCache** See "getdensityCache" for details

**getdistributionCache** See "getdistributionCache" for details

**getdataPoints** See "getdataPoints" for details

**getm** See "getm" for details

**getdensityEstimator** See "getdensityEstimator" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Kakizawa, Y. (2004). Bernstein polynomial probability density estimation. *Journal of Nonparametric Statistics*, 16(5), 709-729.

## Examples

```
# create the model
# we use a MicroBetaChen99Kernel is used as estimator y KakizawaB1 approximation
est <- microBetaChen99Kernel(dataPoints = tuna.r, b = 0.01, modified = FALSE)
model <- kakizawaB2(dataPoints = tuna.r, m = 25, estimator = est)


# examples of usual functions
density(model,0.5)

distribution(model,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(model, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(model, show=TRUE, includePoints=TRUE)
```

---

kakizawaB2 KakizawaB2 *generator method*

---

## Description

User friendly constructor method for KakizawaB2 objects.

## Usage

```
kakizawaB2(dataPoints, estimator=NULL,m=round(length(dataPoints)^(2/5)),
              dataPointsCache=NULL, lower.limit = 0, upper.limit = 1)
```

### Arguments

- `dataPoints` a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator
- `estimator` A bounded density estimator. See all the accepted classes here with getSubclasses("BoundedDensity"). If no estimator is provided here (default value = NULL), a Muller94BoundaryKernel estimator with default parameters and the same dataPoints as those give for the Kakizawa estimator is used.
- `m` a integer value indicating the order of the polynomial approximation. `m` must take values greater than 0
- `dataPointsCache` a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit`
- `lower.limit` a numeric value for the lower limit of the bounded interval for the data
- `upper.limit` a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval

### Details

See `KakizawaB2` class for more details.

---

## KakizawaB3 *Class* "KakizawaB3"

---

### Description

This class deals with B3 approximation to kernel density estimation as described in Kakizawa (2004). This is a Berstein polynomial approximation of the density function which uses Bounded-Density objects instead of a polynomial function. By contrast to the original Kakizawa's approach where only boundary kernels are used, here, any BoundedDensity object is allowed. Using this estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

### Objects from the Class

Objects can be created by using the generator function `kakizawaB3`.

### Slots

- `dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval
- `densityCache`: a numeric vector containing the density for each point in `dataPointsCache`
- `distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`densityEstimator`: a BoundedDensity object used to estimate the density

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getm** See "`getm`" for details

**getdensityEstimator** See "`getdensityEstimator`" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Kakizawa, Y. (2004). Bernstein polynomial probability density estimation. *Journal of Nonparametric Statistics*, 16(5), 709-729.

## Examples

```
# create the model
# we use a MicroBetaChen99Kernel is used as estimator y KakizawaB1 approximation
est <- microBetaChen99Kernel(dataPoints = tuna.r, b = 0.01, modified = FALSE)
model <- kakizawaB3(dataPoints = tuna.r, m = 25, estimator = est)


# examples of usual functions
density(model,0.5)

distribution(model,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(model, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(model, show=TRUE, includePoints=TRUE)
```

---

`kakizawaB3` — `KakizawaB3` *generator method*

---

## Description

User friendly constructor method for `KakizawaB3` objects.

## Usage

```
kakizawaB3(dataPoints, estimator=NULL,m=round(length(dataPoints)^(2/5)),
           dataPointsCache=NULL, lower.limit = 0, upper.limit = 1)
```

## Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `estimator` | A bounded density estimator. See all the accepted classes here with getSubclasses("BoundedDensity"). If no estimator is provided here (default value = NULL), a Muller94BoundaryKernel estimator with default parameters and the same dataPoints as those give for the Kakizawa estimator is used. |
| `m` | a integer value indicating the order of the polynomial approximation. `m` must take values greater than 0 |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See `KakizawaB3` class for more details.

---

`launchApp` — *Shiny launch application*

---

## Description

Runs the shiny service for the bde package.

## Usage

```
launchApp(...)
```

### Arguments

| | |
|---|---|
| ... | no parameters are needed |

## lines *Add a Bounded Density pdf to a Plot*

### Description

Function to draw a bounded density probability density functions in the current plot.

### Arguments

| | |
|---|---|
| x | A bounded density estimator.See all the accepted classes here by running the command `getSubclasses("BoundedDensity")`. |
| ... | Arguments to be passed to methods, such as graphical parameters (see `par`). |

### Methods

`lines(x,...)`

## MacroBetaChen99Kernel *Class* "MacroBetaChen99Kernel"

### Description

This class deals with the density-wise normalization (macro beta) of the Chen's 99 Kernel estimator (as described in Gourierous and Monfort, 2006). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

### Objects from the Class

Objects can be created by using the generator function `macroBetaChen99Kernel`.

### Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`: the bandwidth of the kernel estimator

`modified`: if `TRUE`, the modified version of the kernel estimator is used

`normalizationConst`: this slot is used to save the density-wise normalization constant. It is only for internal use

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getb** See "`getb`" for details

**getmodified** See "`getmodified`" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Chen, S. X. (1999). Beta kernel estimators for density functions. *Computational Statistics & Data Analysis*, 31, 131-145.

Gourieroux, C. and Monfort, A. (2006). (Non) consistency of the Beta Kernel Estimator for Recovery Rate Distribution. *Working Paper 2006-31*, Centre de Recherche en Economie et Statistique.

## Examples

```
# create the model
kernel.noModified <- macroBetaChen99Kernel(dataPoints = tuna.r, b = 0.01,
                        modified = FALSE)
kernel.Modified <- macroBetaChen99Kernel(dataPoints = tuna.r, b = 0.01,
                        modified = TRUE)

# examples of usual functions
density(kernel.noModified,0.5)
density(kernel.Modified,0.5)

distribution(kernel.noModified,1,discreteApproximation=FALSE)
distribution(kernel.noModified,1,discreteApproximation=TRUE)
```

```
distribution(kernel.Modified,1,discreteApproximation=FALSE)
distribution(kernel.Modified,1,discreteApproximation=TRUE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Chen99 Kernels Tuna Data")
lines(kernel.noModified, col="red",lwd=2)
lines(kernel.Modified,col="blue",lwd=2)

# graphical representation using ggplot2
graph <- gplot(list("noModified"=kernel.noModified,
          "modified"=kernel.Modified), show=TRUE)
```

## macroBetaChen99Kernel MacroBetaChen99Kernel *generator method*

### Description

User friendly constructor method for `MacroBetaChen99Kernel` objects.

### Usage

```
macroBetaChen99Kernel(dataPoints, b=length(dataPoints)^(-2/5), dataPointsCache=NULL,
                        modified = FALSE, lower.limit = 0, upper.limit = 1)
```

### Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `modified` | if `TRUE`, the modified version of the kernel estimator is used |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

### Details

See `MacroBetaChen99Kernel` class for more details.

---

`MacroBetaHirukawaJLNKernel`
*Class* "MacroBetaHirukawaJLNKernel"

---

### Description

This class deals with the density-wise normalization (macro beta) of the JLN Kernel estimator as described in Hirukawa (2010). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

### Objects from the Class

Objects can be created by using the generator function `macroBetaHirukawaJLNKernel`.

### Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`: the bandwidth of the kernel estimator

`modified`: if `TRUE`, the modified version of the kernel estimator is used

`normalizationConst`: this slot is used to save the density-wise normalization constant. It is only for internal use

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

### Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "getdataPoints" for details

**getb** See "getb" for details

**getmodified** See "getmodified" for details

### Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

### References

Hirukawa, M. (2010). Nonparametric multiplicative bias correction for kernel-type density estimation on the unit interval. *Computational Statistics & Data Analysis*, 54(2), 473-495.

### Examples

```
# create the model
kernel.noModified <- macroBetaHirukawaJLNKernel(dataPoints = tuna.r, b = 0.01,
                        modified = FALSE)
kernel.Modified <- macroBetaHirukawaJLNKernel(dataPoints = tuna.r, b = 0.01,
                        modified = TRUE)

# examples of usual functions
density(kernel.noModified,0.5)
density(kernel.Modified,0.5)

distribution(kernel.noModified,1,discreteApproximation=FALSE)
distribution(kernel.noModified,1,discreteApproximation=TRUE)

distribution(kernel.Modified,1,discreteApproximation=FALSE)
distribution(kernel.Modified,1,discreteApproximation=TRUE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Chen99 Kernels Tuna Data")
lines(kernel.noModified, col="red",lwd=2)
lines(kernel.Modified,col="blue",lwd=2)

# graphical representation using ggplot2
graph <- gplot(list("noModified"=kernel.noModified,
          "modified"=kernel.Modified), show=TRUE)
```

---

## macroBetaHirukawaJLNKernel

`MacroBetaHirukawaJLNKernel` *generator method*

---

### Description

User friendly constructor method for `MacroBetaHirukawaJLNKernel` objects.

### Usage

```
macroBetaHirukawaJLNKernel(dataPoints, b=length(dataPoints)^(-2/5), dataPointsCache=NULL,
                            modified = FALSE, lower.limit = 0, upper.limit = 1)
```

### Arguments

- `dataPoints` a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator
- `b` the bandwidth of the kernel estimator
- `dataPointsCache` a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit`
- `modified` if `TRUE`, the modified version of the kernel estimator is used
- `lower.limit` a numeric value for the lower limit of the bounded interval for the data
- `upper.limit` a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval

### Details

See `MacroBetaHirukawaJLNKernel` class for more details.

---

`MacroBetaHirukawaTSKernel`

## *Class* "MacroBetaHirukawaTSKernel"

---

### Description

This class deals with the density-wise normalization (macro beta) of the TS Kernel estimator as described in Hirukawa (2010). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

### Objects from the Class

Objects can be created by using the generator function `macroBetaHirukawaTSKernel`.

### Slots

- `dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval
- `densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`: the bandwidth of the kernel estimator

`modified`: if `TRUE`, the modified version of the kernel estimator is used

`c`: a numeric value between 0 and 1. This parameter is used in the TS approximation as a smoothing parameter

`normalizationConst`: this slot is used to save the density-wise normalization constant. It is only for internal use

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getb** See "`getb`" for details

**getmodified** See "`getmodified`" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Hirukawa, M. (2010). Nonparametric multiplicative bias correction for kernel-type density estimation on the unit interval. *Computational Statistics & Data Analysis*, 54(2), 473-495.

## Examples

```
# create the model
kernel.noModified <- macroBetaHirukawaTSKernel(dataPoints = tuna.r, b = 0.01,
                      modified = FALSE, c = 0.5)
kernel.Modified <- macroBetaHirukawaTSKernel(dataPoints = tuna.r, b = 0.01,
                      modified = TRUE, c = 0.5)
```

```
# examples of usual functions
density(kernel.noModified,0.5)
density(kernel.Modified,0.5)

distribution(kernel.noModified,1,discreteApproximation=FALSE)
distribution(kernel.noModified,1,discreteApproximation=TRUE)

distribution(kernel.Modified,1,discreteApproximation=FALSE)
distribution(kernel.Modified,1,discreteApproximation=TRUE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Chen99 Kernels Tuna Data")
lines(kernel.noModified,col="red",lwd=2)
lines(kernel.Modified,col="blue",lwd=2)

# graphical representation using ggplot2
graph <- gplot(list("noModified"=kernel.noModified,
          "modified"=kernel.Modified), show=TRUE)
```

---

macroBetaHirukawaTSKernel

`MacroBetaHirukawaTSKernel` *generator method*

---

## Description

User friendly constructor method for `MacroBetaHirukawaTSKernel` objects.

## Usage

```
macroBetaHirukawaTSKernel(dataPoints, c, b=length(dataPoints)^(-2/5),
                          dataPointsCache=NULL, modified = FALSE, lower.limit = 0,
                            upper.limit = 1)
```

## Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `c` | a numeric value between 0 and 1. This parameter is used in the TS approximation as a smoothing parameter |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `modified` | if `TRUE`, the modified version of the kernel estimator is used |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See MacroBetaHirukawaTSKernel class for more details.

---

MicroBetaChen99Kernel *Class* "MicroBetaChen99Kernel"

---

## Description

This class deals with the kernel-wise normalization of the Chen's 99 Kernel estimator (as described in Gourierous and Monfort, 2006). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

## Objects from the Class

Objects can be created by using the generator function microBetaChen99Kernel.

## Slots

dataPointsCache**:** a numeric vector containing points within the [lower.limit,upper.limit] interval

densityCache**:** a numeric vector containing the density for each point in dataPointsCache

distributionCache**:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

dataPoints**:** a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator

b**:** the bandwidth of the kernel estimator

modified**:** if TRUE, the modified version of the kernel estimator is used

normalizationConstants**:** this slot is used to save the kernel-wise normalization constants. It is only for internal use

lower.limit**:** a numeric value for the lower limit of the bounded interval for the data

upper.limit**:** a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "density" for details

**distribution** See "distribution" for details

**quantile** See "quantile" for details

**rsample** See "rsample" for details

**plot** See "plot" for details

**getdataPointsCache** See "getdataPointsCache" for details

**getdensityCache** See "getdensityCache" for details

**getdistributionCache** See "getdistributionCache" for details

**getdataPoints** See "getdataPoints" for details

**getb** See "getb" for details

**getmodified** See "getmodified" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References


Chen, S. X. (1999). Beta kernel estimators for density functions. *Computational Statistics & Data Analysis*, 31, 131-145.

Gourieroux, C. and Monfort, A. (2006). (Non) consistency of the Beta Kernel Estimator for Recovery Rate Distribution. *Working Paper 2006-31*, Centre de Recherche en Economie et Statistique.


## Examples

```
# create the model
kernel.noModified <- microBetaChen99Kernel(dataPoints = tuna.r, b = 0.01,
                      modified = FALSE)
kernel.Modified <- microBetaChen99Kernel(dataPoints = tuna.r, b = 0.01,
                      modified = TRUE)

# examples of usual functions
density(kernel.noModified,0.5)
density(kernel.Modified,0.5)

distribution(kernel.noModified,1,discreteApproximation=FALSE)
distribution(kernel.noModified,1,discreteApproximation=TRUE)

distribution(kernel.Modified,1,discreteApproximation=FALSE)
distribution(kernel.Modified,1,discreteApproximation=TRUE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Chen99 Kernels Tuna Data")
lines(kernel.noModified, col="red",lwd=2)
lines(kernel.Modified,col="blue",lwd=2)

# graphical representation using ggplot2
graph <- gplot(list("noModified"=kernel.noModified,
          "modified"=kernel.Modified), show=TRUE)
```

## microBetaChen99Kernel MicroBetaChen99Kernel *generator method*

### Description

User friendly constructor method for MicroBetaChen99Kernel objects.

### Usage

```
microBetaChen99Kernel(dataPoints, b=length(dataPoints)^(-2/5), dataPointsCache=NULL,
                        modified = FALSE, lower.limit = 0, upper.limit = 1)
```

### Arguments

| | |
|---|---|
| dataPoints | a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator |
| b | the bandwidth of the kernel estimator |
| dataPointsCache | a numeric vector containing points within the [lower.limit,upper.limit] interval. These points are used for convenience to cache density and distribution values. If dataPointsCache=NULL the values are initialized to a sequence of 101 equally spaced values from lower.limit to upper.limit |
| modified | if TRUE, the modified version of the kernel estimator is used |
| lower.limit | a numeric value for the lower limit of the bounded interval for the data |
| upper.limit | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the [lower.limit,upper.limit] interval |

### Details

See MicroBetaChen99Kernel class for more details.

## mise *Mean Integrated Squared Error*

### Description

Computes the mean integrated squared error (MISE) for two given Bounded density objects.

### Usage

```
mise(model1,model2,discreteApproximation = TRUE)
```

### Arguments

model1 — a bounded density object. See getSubclasses("BoundedDensity") to see all the allowed class objects

model2 — a bounded density object. See getSubclasses("BoundedDensity") to see all the allowed class objects

discreteApproximation — If TRUE, the mise is calculated using the data stored in the cache. Otherwise the integral is computed.

### Examples

```
# a general approximation to a Beta(1,10) distribution using BoundedDensity objects
cache <- seq(0,1,0.01)
dens  <- dbeta(cache,1,10)
bd    <- boundedDensity(x=cache,densities=dens)

# a BrVitale approximation to the Beta(1,10) distribution using a random data sample to
# learn the model
dataSample <- rbeta(100,1,10)
kernel     <- hirukawaTSKernel(dataPoints=dataSample, b=0.1, c=0.3,
                                dataPointsCache=cache, modified=FALSE)

# compute the mise
mise(bd,kernel,discreteApproximation=TRUE)
mise(bd,kernel,discreteApproximation=FALSE)
```

---

Muller91BoundaryKernel

## *Class* "Muller91BoundaryKernel"

---

### Description

This class deals with Kernel estimators for bounded densities using boundary kernel described in Muller (1991). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations. Note that this kernel estimator is not normalized and therefore it is not a probability distribution (the cumulative density function may return values greater than 1).

### Objects from the Class

Objects can be created by using the generator function muller91BoundaryKernel.

## Slots

**dataPointsCache:** a numeric vector containing points within the [lower.limit,upper.limit] interval

**densityCache:** a numeric vector containing the density for each point in dataPointsCache

**distributionCache:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

**dataPoints:** a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator

**b:** the bandwidth of the kernel estimator

**mu:** a integer value indicating the degree of smoothness for the boundary kernel. mu can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel)

**lower.limit:** a numeric value for the lower limit of the bounded interval for the data

**upper.limit:** a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "density" for details

**distribution** See "distribution" for details

**quantile** See "quantile" for details

**rsample** See "rsample" for details

**plot** See "plot" for details

**getdataPointsCache** See "getdataPointsCache" for details

**getdensityCache** See "getdensityCache" for details

**getdistributionCache** See "getdistributionCache" for details

**getdataPoints** See "getdataPoints" for details

**getb** See "getb" for details

**getmu** See "getmu" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Muller, H. (1991). Smooth optimum kernel estimators near endpoints. *Biometrika*, 78(3), 521-530.

## Examples

```
# create the model
kernel <- muller91BoundaryKernel(dataPoints = tuna.r, b = 0.01, mu = 2)


# examples of usual functions
density(kernel,0.5)

distribution(kernel,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(kernel, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(kernel, show=TRUE, includePoints=TRUE)
```

---

`muller91BoundaryKernel`

# `Muller91BoundaryKernel` *generator method*

---

## Description

User friendly constructor method for `Muller91BoundaryKernel` objects.

## Usage

```
muller91BoundaryKernel(dataPoints,  mu=1, b=length(dataPoints)^(-2/5),
                           dataPointsCache=NULL, lower.limit = 0,
                           upper.limit = 1)
```

## Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `mu` | a integer value indicating the degree of smoothness for the boundary kernel. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel) |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See `Muller91BoundaryKernel` class for more details.

---

`Muller94BoundaryKernel` *Class* `"Muller94BoundaryKernel"`

---

## Description

This class deals with Kernel estimators for bounded densities using boundary kernel described in Muller and Wang (1994). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations. Note that this kernel estimator is not normalized and therefore it is not a probability distribution (the cumulative density function may return values greater than 1).

## Objects from the Class

Objects can be created by using the generator function `muller94BoundaryKernel`.

## Slots

`dataPointsCache`**:** a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`**:** a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`**:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`**:** a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`**:** the bandwidth of the kernel estimator

`mu`**:** a integer value indicating the degree of smoothness for the boundary kernel. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel)

`lower.limit`**:** a numeric value for the lower limit of the bounded interval for the data

`upper.limit`**:** a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "getdataPointsCache" for details

**getdensityCache** See "getdensityCache" for details

**getdistributionCache** See "getdistributionCache" for details

**getdataPoints** See "getdataPoints" for details

**getb** See "getb" for details

**getmu** See "getmu" for details

### Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

### References

Muller, H. and Wang, J. (1994). Hazard rate estimation under random censoring with varying kernels and bandwidths. *Biometrics*, 50(1), 61-76.

### Examples

```
# create the model
kernel <- muller94BoundaryKernel(dataPoints = tuna.r, b = 0.01, mu = 2)


# examples of usual functions
density(kernel,0.5)

distribution(kernel,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(kernel, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(kernel, show=TRUE, includePoints=TRUE)
```

---

## muller94BoundaryKernel — `Muller94BoundaryKernel` *generator method*

### Description

User friendly constructor method for `Muller94BoundaryKernel` objects.

### Usage

```
muller94BoundaryKernel(dataPoints, mu=1, b=length(dataPoints)^(-2/5),
                       dataPointsCache=NULL, lower.limit = 0,
                       upper.limit = 1)
```

### Arguments

dataPoints — a numeric vector containing data samples within the [lower.limit,upper.limit] interval. These data samples are used to obtain the kernel estimator

mu — a integer value indicating the degree of smoothness for the boundary kernel. mu can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel)

b — the bandwidth of the kernel estimator

dataPointsCache — a numeric vector containing points within the [lower.limit,upper.limit] interval. These points are used for convenience to cache density and distribution values. If dataPointsCache=NULL the values are initialized to a sequence of 101 equally spaced values from lower.limit to upper.limit

lower.limit — a numeric value for the lower limit of the bounded interval for the data

upper.limit — a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the [lower.limit,upper.limit] interval

### Details

See Muller94BoundaryKernel class for more details.

---

## NoBoundaryKernel — *Class* "NoBoundaryKernel"

---

### Description

This class deals with Kernel estimators for bounded densities using boundary kernels where the same kernel function is used for all regions: left boundary, interior and right boundary. The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations. Note that this kernel estimator is not normalized and therefore it is not a probability distribution (the cumulative density function may return values greater than 1).

### Objects from the Class

Objects can be created by using the generator function noBoundaryKernel.

### Slots

**dataPointsCache:** a numeric vector containing points within the [lower.limit,upper.limit] interval

**densityCache:** a numeric vector containing the density for each point in dataPointsCache

**distributionCache:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`: the bandwidth of the kernel estimator

`mu`: a integer value indicating the degree of smoothness for the boundary kernel. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel)

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getb** See "`getb`" for details

**getmu** See "`getmu`" for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## Examples

```
# create the model
kernel <- noBoundaryKernel(dataPoints = tuna.r, b = 0.01, mu = 2)


# examples of usual functions
density(kernel,0.5)

distribution(kernel,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(kernel, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(kernel, show=TRUE, includePoints=TRUE)
```

`noBoundaryKernel` `NoBoundaryKernel` *generator method*

## Description

User friendly constructor method for `NoBoundaryKernel` objects.

## Usage

```
noBoundaryKernel(dataPoints, mu=1, b=length(dataPoints)^(-2/5),
                 dataPointsCache=NULL, lower.limit = 0,
                 upper.limit = 1)
```

## Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `mu` | a integer value indicating the degree of smoothness for the boundary kernel. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel) |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See `NoBoundaryKernel` class for more details.

`NormalizedBoundaryKernel` *Class* `"NormalizedBoundaryKernel"`

## Description

This class deals with Kernel estimators for bounded densities using renormalized boundary kernel described in Kakizawa (2004). The kernel estimator is computed using the provided data samples. Using this kernel estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations. Note that, the renormalization of this kernel guarantees non-negative density values. However, despite its name, the normalized boundary kernel is not a probability distribution (the cumulative density function may return values greater than 1).

### Objects from the Class

Objects can be created by using the generator function `normalizedBoundaryKernel`.

### Slots

`dataPointsCache`: a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`: a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`: a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`: a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`b`: the bandwidth of the kernel estimator

`mu`: a integer value indicating the degree of smoothness for the boundary kernel. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel)

`lower.limit`: a numeric value for the lower limit of the bounded interval for the data

`upper.limit`: a numeric value for the upper limit of the bounded interval for the data

### Methods

**density** See "`density`" for details

**distribution** See "`distribution`" for details

**quantile** See "`quantile`" for details

**rsample** See "`rsample`" for details

**plot** See "`plot`" for details

**getdataPointsCache** See "`getdataPointsCache`" for details

**getdensityCache** See "`getdensityCache`" for details

**getdistributionCache** See "`getdistributionCache`" for details

**getdataPoints** See "`getdataPoints`" for details

**getb** See "`getb`" for details

**getmu** See "`getmu`" for details

### Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

### References

Kakizawa, Y. (2004). Bernstein polynomial probability density estimation. *Journal of Nonparametric Statistics*, 16(5), 709-729.

### Examples

```
# create the model
kernel <- normalizedBoundaryKernel(dataPoints = tuna.r, b = 0.01, mu = 2)


# examples of usual functions
density(kernel,0.5)

distribution(kernel,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(kernel,col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(kernel, show=TRUE, includePoints=TRUE)
```

---

## normalizedBoundaryKernel — `NormalizedBoundaryKernel` *generator method*

---

### Description

User friendly constructor method for `NormalizedBoundaryKernel` objects.

### Usage

```
normalizedBoundaryKernel(dataPoints, mu=1, b=length(dataPoints)^(-2/5),
                             dataPointsCache=NULL, lower.limit = 0,
                             upper.limit = 1)
```

### Arguments

| | |
|---|---|
| `dataPoints` | a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator |
| `mu` | a integer value indicating the degree of smoothness for the boundary kernel. `mu` can take the following values: 0 (uniform kernel), 1 (Epanechnikov kernel), 2 (biweight kernel) or 3 (triweight kernel) |
| `b` | the bandwidth of the kernel estimator |
| `dataPointsCache` | a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit` |
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See NormalizedBoundaryKernel class for more details.

---

plot *Bounded Density Plotting*

---

## Description

Function to plot bounded density probability density functions.

## Arguments

| | |
|---|---|
| x | A bounded density estimator. See all the accepted classes here by running the command getSubclasses("BoundedDensity"). |
| main, type, xlab, ylab | Graphical parameters with default values (see par). |
| ... | Arguments to be passed to methods, such as (other) graphical parameters (see par). |

## Methods

```
plot(x,main="Bounded density",type="l",xlab="X",ylab="Density",...)
```

---

quantile *Quantile*

---

## Description

Quantile function for the given bounded density object.

## Arguments

| | |
|---|---|
| x | A bounded density estimator. See all the accepted classes here by running the command getSubclasses("BoundedDensity").This parameter is named x instead of .Object to agree with other already defined density methods. |
| p | Vector of probabilities |

## Methods

```
quantile(x,p)
```

## `rsample` *Random sample*

### Description

Random generator function for the given bounded density object.

### Arguments

| | |
|---|---|
| `.Object` | A bounded density estimator. See all the accepted classes here by running the command `getSubclasses("BoundedDensity")`. |
| `n` | number of random observations to be generated |

### Methods

`rsample(.Object,n)`

## `suicide.r` *Scaled data from suicide risk data*

### Description

The dataset comprises lengths (in days) of psychiatric treatment spells for patients used as controls in a study of suicide risks. The data have been scaled to the interval [0,1] by dividing each data sample by the maximum value.

### Usage

```
suicide.r
```

### Format

A vector containing 86 observations.

### Source

The data were obtained from Silverman (1996) Table 2.1

### References

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall

Copas, J. B. and Fryer, M. J. (1980). Density estimation and suicide risks in psychiatric treatment. *Journal of the Royal Statistical Society. Series A*, 143(2), 167-176

## tgaussian — *Synthetic dataset from a truncated Gaussian distribution*

### Description

This is a synthetic generated dataset sampling a truncated Gaussian distribution on the interval [0,1] with `mean=0` and `sd=0.25`

### Usage

```
tgaussian
```

### Format

A vector containing 10000 observations.

## tuna.r — *Scaled* `tuna` *data*

### Description

The `tuna` data come from an aerial line transect survey of Southern Bluefin Tuna in the Great Australian Bight and it is included in the `boot` package. The `tuna.r` data is a scaled version of the `tuna` data within the [0,1] interval. This new data set is obtained as follows:

```
library(boot)
tuna.r <- tuna$y/17
```

### Usage

```
tuna.r
```

### Format

A vector containing 64 observations.

### Source

The data were obtained from

Chen, S.X. (1996). Empirical likelihood confidence intervals for nonparametric density estimation. *Biometrica*, 83, 329-341.

### See Also

`tuna`

`Vitale` *Class* `"Vitale"`

## Description

This class deals with Vitale (1975) Bernstein Polynomial approximation as described in Leblanc (2009). The polynomial estimator is computed using the provided data samples. Using this polynomial estimator, the methods implemented in the class can be used to compute densities, values of the distribution function, quantiles, sample the distribution and obtain graphical representations.

## Objects from the Class

Objects can be created by using the generator function `vitale`.

## Slots

`dataPointsCache`**:** a numeric vector containing points within the `[lower.limit,upper.limit]` interval

`densityCache`**:** a numeric vector containing the density for each point in `dataPointsCache`

`distributionCache`**:** a numeric vector used to cache the values of the distribution function. This slot is included to improve the performance of the methods when multiple calculations of the distribution function are used

`dataPoints`**:** a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator

`m`**:** the order of the polynomial approximation

`lower.limit`**:** a numeric value for the lower limit of the bounded interval for the data

`upper.limit`**:** a numeric value for the upper limit of the bounded interval for the data

## Methods

**density** See `"density"` for details

**distribution** See `"distribution"` for details

**quantile** See `"quantile"` for details

**rsample** See `"rsample"` for details

**plot** See `"plot"` for details

**getdataPointsCache** See `"getdataPointsCache"` for details

**getdensityCache** See `"getdensityCache"` for details

**getdistributionCache** See `"getdistributionCache"` for details

**getdataPoints** See `"getdataPoints"` for details

**getm** See `"getm"` for details

## Author(s)

Guzman Santafe, Borja Calvo and Aritz Perez

## References

Vitale, R. A. (1975). A Bernstein polynomial approach to density function estimation. *Statistical Inference and Related Topics*, 2, 87-99.

Leblanc, A. (2010). A bias-reduced approach to density estimation using Bernstein polynomials. *Journal of Nonparametric Statistics*, 22(4), 459-475.

## Examples

```
# create the model
model <- vitale(dataPoints = tuna.r, m = 25)


# examples of usual functions
density(model,0.5)

distribution(model,0.5,discreteApproximation=FALSE)

# graphical representation
hist(tuna.r,freq=FALSE,main="Tuna Data")
lines(model, col="red",lwd=2)

# graphical representation using ggplot2
graph <- gplot(model, show=TRUE, includePoints=TRUE)
```

---

# `vitale` — `Vitale` *generator method*

---

## Description

User friendly constructor method for `Vitale` objects.

## Usage

```
vitale(dataPoints, m=round(length(dataPoints)^(2/5)), dataPointsCache=NULL,
        lower.limit = 0, upper.limit = 1)
```

## Arguments

- `dataPoints` — a numeric vector containing data samples within the `[lower.limit,upper.limit]` interval. These data samples are used to obtain the kernel estimator
- `m` — a integer value indicating the order of the polynomial approximation. `m` must take values greater than 0
- `dataPointsCache` — a numeric vector containing points within the `[lower.limit,upper.limit]` interval. These points are used for convenience to cache density and distribution values. If `dataPointsCache=NULL` the values are initialized to a sequence of 101 equally spaced values from `lower.limit` to `upper.limit`

| | |
|---|---|
| `lower.limit` | a numeric value for the lower limit of the bounded interval for the data |
| `upper.limit` | a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the `[lower.limit,upper.limit]` interval |

## Details

See `Vitale` class for more details.

# Index