

Package: bde (via r-universe)

August 28, 2024

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

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

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

RemoteUrl <https://github.com/cran/bde>

RemoteRef HEAD

RemoteSha 675f7409d2701511af4830dc7c2d03d59a4b795f

Contents

bde	3
beta_0.75_0.65	5
beta_1_10	5
beta_5_10	5
BoundedDensity	6
boundedDensity	7
BrVitale	8
brVitale	9
Chen99Kernel	10
chen99Kernel	12
density	12
distribution	13
eruption	13
getb	14
getc	14
getDataPoints	15
getDataPointsCache	15
getDensityCache	16
getDensityEstimator	16
getDistributionCache	16
getGamma	17
getM	17
getm	18
getModified	18
getmu	19
getSubclasses	19
gplot	20
HirukawaJLNKernel	20
hirukawaJLNKernel	22
HirukawaTSKernel	23
hirukawaTSKernel	24
JonesCorrectionMuller91BoundaryKernel	25
jonesCorrectionMuller91BoundaryKernel	27
JonesCorrectionMuller94BoundaryKernel	28
jonesCorrectionMuller94BoundaryKernel	30
KakizawaB1	30
kakizawaB1	32
KakizawaB2	33
kakizawaB2	34
KakizawaB3	35
kakizawaB3	37
launchApp	37
lines	38
MacroBetaChen99Kernel	38
macroBetaChen99Kernel	40
MacroBetaHirukawaJLNKernel	41

macroBetaHirukawaJLNKernel	42
MacroBetaHirukawaTSKernel	43
macroBetaHirukawaTSKernel	45
MicroBetaChen99Kernel	46
microBetaChen99Kernel	48
mise	48
Muller91BoundaryKernel	49
muller91BoundaryKernel	51
Muller94BoundaryKernel	52
muller94BoundaryKernel	53
NoBoundaryKernel	54
noBoundaryKernel	56
NormalizedBoundaryKernel	56
normalizedBoundaryKernel	58
plot	59
quantile	59
rsample	60
suicide.r	60
tgaussian	61
tuna.r	61
Vitale	62
vitale	63
Index	65

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: <ul style="list-style-type: none"> • "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	<i>Synthetic dataset from a beta distribution</i>
----------------	---

Description

This is a synthetic generated dataset sampling a beta distribution with parameters $\text{shape1} = 0.75$ and $\text{shape2} = 0.65$

Usage

beta_0.75_0.65

Format

A vector containing 10000 observations.

beta_1_10	<i>Synthetic dataset from a beta distribution</i>
-----------	---

Description

This is a synthetic generated dataset sampling a beta distribution with parameters $\text{shape1} = 1$ and $\text{shape2} = 10$

Usage

beta_1_10

Format

A vector containing 10000 observations.

beta_5_10	<i>Synthetic dataset from a beta distribution</i>
-----------	---

Description

This is a synthetic generated dataset sampling a beta distribution with parameters $\text{shape1} = 5$ and $\text{shape2} = 10$

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 <i>generator method</i>
----------------	--

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. *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 <- 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 <code>[lower.limit,upper.limit]</code> interval. These points are used for convenience to cache density and distribution values. If <code>dataPointsCache=NULL</code> the values are initialized to a sequence of 101 equally spaced values from <code>lower.limit</code> to <code>upper.limit</code>
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 <code>[lower.limit,upper.limit]</code> interval

Details

See [BrVitale](#) class for more details.

Chen99Kernel	<i>Class "Chen99Kernel"</i>
--------------	-----------------------------

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 <i>generator method</i>
--------------	--------------------------------------

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	<i>Probability Density Function (pdf)</i>
---------	---

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 <code>getSubclasses("BoundedDensity")</code> . 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	<i>Cumulative Density Function (cdf)</i>
--------------	--

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 <code>getSubclasses("BoundedDensity")</code> .
x	Vector of points where the density function is evaluated. These points must be in the interval <code>[.Object@lower.limit, .Object@upper.limit]</code>
discreteApproximation	Logical; if TRUE the distribution function is computed using a discrete approximation using the values cached in <code>dataPointsCache</code> and <code>densityCache</code> . 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	<i>Eruption lengths of Old Faithful geyser</i>
----------	--

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	<i>Accessor method for b slot</i>
------	-----------------------------------

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	<i>Accessor method for c slot</i>
------	-----------------------------------

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	<i>Accesor method for dataPoints slot</i>
---------------	---

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	<i>Accesor method for DataPointsCache slot</i>
--------------------	--

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	<i>Accesor method for gamma slot</i>
----------	--------------------------------------

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	<i>Accesor method for M slot</i>
------	----------------------------------

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)

<code>getm</code>	<i>Accesor method for m slot</i>
-------------------	----------------------------------

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)`

<code>getmodified</code>	<i>Accesor method for modified slot</i>
--------------------------	---

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	<i>Accesor method for Mu slot</i>
-------	-----------------------------------

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 <code>getSubclasses("BoundaryKernel")</code> .
---------	--

Methods

`getmu(.Object)`

getSubclasses	<i>List of subclasses</i>
---------------	---------------------------

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 <code>getSubclasses("BoundedDensity")</code> .
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 <code>?par</code> 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

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

<code>dataPoints</code>	a numeric vector containing data samples within the <code>[lower.limit, upper.limit]</code> interval. These data samples are used to obtain the kernel estimator
<code>c</code>	a numeric value between 0 and 1. This parameter is used in the TS approximation as a smoothing parameter
<code>b</code>	the bandwidth of the kernel estimator
<code>dataPointsCache</code>	a numeric vector containing points within the <code>[lower.limit, upper.limit]</code> interval. These points are used for convenience to cache density and distribution values. If <code>dataPointsCache=NULL</code> the values are initialized to a sequence of 101 equally spaced values from <code>lower.limit</code> to <code>upper.limit</code>
<code>modified</code>	if TRUE, the modified version of the kernel estimator is used
<code>lower.limit</code>	a numeric value for the lower limit of the bounded interval for the data
<code>upper.limit</code>	a numeric value for the upper limit of the bounded interval for the data. That is, the data is with the <code>[lower.limit, upper.limit]</code> 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 function $-K(u)$ 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 [JonesCorrectionMuller94BoundaryKernel](#) 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 function $-K(u)$ 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 Bernstein 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 <i>generator method</i>
------------	------------------------------------

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

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

`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	<i>Class "KakizawaB2"</i>
------------	---------------------------

Description

This class deals with B2 approximation to kernel density estimation as described in Kakizawa (2004). This is a Bernstein 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

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

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 <code>getSubclasses("BoundedDensity")</code> . 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	<i>Shiny launch application</i>
-----------	---------------------------------

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.
 Gouriéroux, 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 <- ggplot(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

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

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 <i>generator method</i>
------------------	--

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	<i>Class "NormalizedBoundaryKernel"</i>
--------------------------	---

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	<i>Random sample</i>
---------	----------------------

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 <code>getSubclasses("BoundedDensity")</code> .
n	number of random observations to be generated

Methods

`rsample(.Object,n)`

suicide.r	<i>Scaled data from suicide risk data</i>
-----------	---

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	<i>Synthetic dataset from a truncated Gaussian distribution</i>
-----------	---

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	<i>Scaled tuna data</i>
--------	-------------------------

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. *Biometrika*, 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

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

`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

* **bde-methods**

launchApp, 37

* **classes**

BoundedDensity, 6

BrVitale, 8

Chen99Kernel, 10

HirukawaJLNKernel, 20

HirukawaTSKernel, 23

JonesCorrectionMuller91BoundaryKernel, 25

JonesCorrectionMuller94BoundaryKernel, 28

KakizawaB1, 30

KakizawaB2, 33

KakizawaB3, 35

MacroBetaChen99Kernel, 38

MacroBetaHirukawaJLNKernel, 41

MacroBetaHirukawaTSKernel, 43

MicroBetaChen99Kernel, 46

Muller91BoundaryKernel, 49

Muller94BoundaryKernel, 52

NoBoundaryKernel, 54

NormalizedBoundaryKernel, 56

Vitale, 62

* **constructor**

bde, 3

* **datasets**

beta_0.75_0.65, 5

beta_1_10, 5

beta_5_10, 5

eruption, 13

suicide.r, 60

tgaussian, 61

tuna.r, 61

* **generator-methods**

boundedDensity, 7

brVitale, 9

chen99Kernel, 12

hirukawaJLNKernel, 22

hirukawaTSKernel, 24

jonesCorrectionMuller91BoundaryKernel, 27

jonesCorrectionMuller94BoundaryKernel, 30

kakizawaB1, 32

kakizawaB2, 34

kakizawaB3, 37

macroBetaChen99Kernel, 40

macroBetaHirukawaJLNKernel, 42

macroBetaHirukawaTSKernel, 45

microBetaChen99Kernel, 48

muller91BoundaryKernel, 51

muller94BoundaryKernel, 53

noBoundaryKernel, 56

normalizedBoundaryKernel, 58

vitale, 63

* **methods**

density, 12

distribution, 13

getb, 14

getc, 14

getdataPoints, 15

getdataPointsCache, 15

getdensityCache, 16

getdensityEstimator, 16

getdistributionCache, 16

getgamma, 17

getM, 17

getm, 18

getmodified, 18

getmu, 19

gplot, 20

lines, 38

plot, 59

quantile, 59

rsample, 60

* **utils-methods**

getSubclasses, 19

- mise, 48
- bde, 3
- beta_0.75_0.65, 5
- beta_1_10, 5
- beta_5_10, 5
- BoundedDensity, 6, 7
- boundedDensity, 6, 7
- BoundedDensity-class (BoundedDensity), 6
- boundedDensity-generator (boundedDensity), 7
- BrVitale, 8, 10, 17
- brVitale, 8, 9
- BrVitale-class (BrVitale), 8
- brVitale-generator (brVitale), 9
- Chen99Kernel, 10, 12
- chen99Kernel, 10, 12
- Chen99Kernel-class (Chen99Kernel), 10
- chen99Kernel-generator (chen99Kernel), 12
- density, 6, 8, 11, 12, 21, 23, 26, 29, 31, 33, 36, 39, 41, 44, 46, 50, 52, 55, 57, 62
- density, BoundaryKernel-method (density), 12
- density, BoundedDensity-method (density), 12
- density, BrVitale-method (density), 12
- density, Chen99Kernel-method (density), 12
- density, HirukawaJLNKernel-method (density), 12
- density, HirukawaTSKernel-method (density), 12
- density, JonesCorrectionMuller91BoundaryKernel-method (density), 12
- density, JonesCorrectionMuller94BoundaryKernel-method (density), 12
- density, KakizawaB1-method (density), 12
- density, KakizawaB2-method (density), 12
- density, KakizawaB3-method (density), 12
- density, MacroBetaChen99Kernel-method (density), 12
- density, MacroBetaHirukawaJLNKernel-method (density), 12
- density, MacroBetaHirukawaTSKernel-method (density), 12
- density, MicroBetaChen99Kernel-method (density), 12
- density, Muller91BoundaryKernel-method (density), 12
- density, Muller94BoundaryKernel-method (density), 12
- density, NoBoundaryKernel-method (density), 12
- density, NormalizedBoundaryKernel-method (density), 12
- density, Vitale-method (density), 12
- density-method (density), 12
- distribution, 6, 8, 11, 13, 21, 23, 26, 29, 31, 33, 36, 39, 41, 44, 46, 50, 52, 55, 57, 62
- distribution, BernsteinPolynomials-method (distribution), 13
- distribution, BoundaryKernel-method (distribution), 13
- distribution, BoundedDensity-method (distribution), 13
- distribution, BrVitale-method (distribution), 13
- distribution, Chen99Kernel-method (distribution), 13
- distribution, HirukawaJLNKernel-method (distribution), 13
- distribution, HirukawaTSKernel-method (distribution), 13
- distribution, JonesCorrectionMuller91BoundaryKernel-method (distribution), 13
- distribution, JonesCorrectionMuller94BoundaryKernel-method (distribution), 13
- distribution, KakizawaB1-method (distribution), 13
- distribution, KakizawaB2-method (distribution), 13
- distribution, KakizawaB3-method (distribution), 13
- distribution, KernelDensity-method (distribution), 13
- distribution, MacroBetaChen99Kernel-method (distribution), 13
- distribution, MacroBetaHirukawaJLNKernel-method (distribution), 13
- distribution, MacroBetaHirukawaTSKernel-method (distribution), 13
- distribution, MicroBetaChen99Kernel-method

- (distribution), 13
- distribution,Muller91BoundaryKernel-method (distribution), 13
- distribution,Muller94BoundaryKernel-method (distribution), 13
- distribution,NoBoundaryKernel-method (distribution), 13
- distribution,NormalizedBoundaryKernel-method (distribution), 13
- distribution,Vitale-method (distribution), 13
- distribution-methods (distribution), 13
- eruption, 13
- getb, 11, 14, 21, 23, 26, 29, 39, 42, 44, 47, 50, 53, 55, 57
- getb,Chen99Kernel-method (getb), 14
- getb,HirukawaJLNKernel-method (getb), 14
- getb,HirukawaTSKernel-method (getb), 14
- getb,KernelDensity-method (getb), 14
- getb,MacroBetaChen99Kernel-method (getb), 14
- getb,MacroBetaHirukawaJLNKernel-method (getb), 14
- getb,MacroBetaHirukawaTSKernel-method (getb), 14
- getb,MicroBetaChen99Kernel-method (getb), 14
- getb,Muller91BoundaryKernel-method (getb), 14
- getb,Muller94BoundaryKernel-method (getb), 14
- getb,NoBoundaryKernel-method (getb), 14
- getb,NormalizedBoundaryKernel-method (getb), 14
- getb-methods (getb), 14
- getc, 14, 23
- getc,HirukawaTSKernel-method (getc), 14
- getc-methods (getc), 14
- getdataPoints, 8, 11, 15, 21, 23, 26, 29, 31, 34, 36, 39, 42, 44, 47, 50, 53, 55, 57, 62
- getdataPoints,BernsteinPolynomials-method (getdataPoints), 15
- getdataPoints,BrVitale-method (getdataPoints), 15
- getdataPoints,Chen99kernel-method (getdataPoints), 15
- getdataPoints,HirukawaJLNKernel-method (getdataPoints), 15
- getdataPoints,HirukawaTSKernel-method (getdataPoints), 15
- getdataPoints,KakizawaB1-method (getdataPoints), 15
- getdataPoints,KakizawaB2-method (getdataPoints), 15
- getdataPoints,KakizawaB3-method (getdataPoints), 15
- getdataPoints,KernelDensity-method (getdataPoints), 15
- getdataPoints,MacroBetaChen99Kernel-method (getdataPoints), 15
- getdataPoints,MacroBetaHirukawaJLNKernel-method (getdataPoints), 15
- getdataPoints,MacroBetaHirukawaTSKernel-method (getdataPoints), 15
- getdataPoints,MicroBetaChen99Kernel-method (getdataPoints), 15
- getdataPoints,Muller91BoundaryKernel-method (getdataPoints), 15
- getdataPoints,Muller94BoundaryKernel-method (getdataPoints), 15
- getdataPoints,NoBoundaryKernel-method (getdataPoints), 15
- getdataPoints,NormalizedBoundaryKernel-method (getdataPoints), 15
- getdataPoints,Vitale-method (getdataPoints), 15
- getdataPoints-methods (getdataPoints), 15
- getdataPointsCache, 6, 8, 11, 15, 21, 23, 26, 29, 31, 34, 36, 39, 41, 44, 46, 50, 53, 55, 57, 62
- getdataPointsCache,BoundedDensity-method (getdataPointsCache), 15
- getdataPointsCache,BrVitale-method (getdataPointsCache), 15
- getdataPointsCache,Chen99Kernel-method (getdataPointsCache), 15
- getdataPointsCache,HirukawaJLNKernel-method (getdataPointsCache), 15
- getdataPointsCache,HirukawaTSKernel-method (getdataPointsCache), 15
- getdataPointsCache,KakizawaB1-method (getdataPointsCache), 15
- getdataPointsCache,KakizawaB2-method

- (getdataPointsCache), 15
- getdataPointsCache,KakizawaB3-method (getdataPointsCache), 15
- getdataPointsCache,MacroBetaChen99Kernel-method (getdataPointsCache), 15
- getdataPointsCache,MacroBetaHirukawaJLNKernel-method (getdataPointsCache), 15
- getdataPointsCache,MacroBetaHirukawaTSKernel-method (getdataPointsCache), 15
- getdataPointsCache,MicroBetaChen99Kernel-method (getdataPointsCache), 15
- getdataPointsCache,Muller91BoundaryKernel-method (getdataPointsCache), 15
- getdataPointsCache,Muller94BoundaryKernel-method (getdataPointsCache), 15
- getdataPointsCache,NoBoundaryKernel-method (getdataPointsCache), 15
- getdataPointsCache,NormalizedBoundaryKernel-method (getdataPointsCache), 15
- getdataPointsCache,Vitale-method (getdataPointsCache), 15
- getdataPointsCache-methods (getdataPointsCache), 15
- getdensityCache, 6, 8, 11, 16, 21, 23, 26, 29, 31, 34, 36, 39, 41, 44, 47, 50, 53, 55, 57, 62
- getdensityCache,BoundedDensity-method (getdensityCache), 16
- getdensityCache,BrVitale-method (getdensityCache), 16
- getdensityCache,Chen99Kernel-method (getdensityCache), 16
- getdensityCache,HirukawaJLNKernel-method (getdensityCache), 16
- getdensityCache,HirukawaTSKernel-method (getdensityCache), 16
- getdensityCache,KakizawaB1-method (getdensityCache), 16
- getdensityCache,KakizawaB2-method (getdensityCache), 16
- getdensityCache,KakizawaB3-method (getdensityCache), 16
- getdensityCache,MacroBetaChen99Kernel-method (getdensityCache), 16
- getdensityCache,MacroBetaHirukawaJLNKernel-method (getdensityCache), 16
- getdensityCache,MacroBetaHirukawaTSKernel-method (getdensityCache), 16
- getdensityCache,MicroBetaChen99Kernel-method (getdensityCache), 16
- getdensityCache,Muller91BoundaryKernel-method (getdensityCache), 16
- getdensityCache,Muller94BoundaryKernel-method (getdensityCache), 16
- getdensityCache,NoBoundaryKernel-method (getdensityCache), 16
- getdensityCache,NormalizedBoundaryKernel-method (getdensityCache), 16
- getdensityCache,Vitale-method (getdensityCache), 16
- getdensityCache-methods (getdensityCache), 16
- getdensityEstimator, 16, 31, 34, 36
- getdensityEstimator,KakizawaB1-method (getdensityEstimator), 16
- getdensityEstimator,KakizawaB2-method (getdensityEstimator), 16
- getdensityEstimator,KakizawaB3-method (getdensityEstimator), 16
- getdensityEstimator-methods (getdensityEstimator), 16
- getdistributionCache, 6, 8, 11, 16, 21, 23, 26, 29, 31, 34, 36, 39, 41, 44, 47, 50, 53, 55, 57, 62
- getdistributionCache,BoundedDensity-method (getdistributionCache), 16
- getdistributionCache,BrVitale-method (getdistributionCache), 16
- getdistributionCache,Chen99Kernel-method (getdistributionCache), 16
- getdistributionCache,HirukawaJLNKernel-method (getdistributionCache), 16
- getdistributionCache,HirukawaTSKernel-method (getdistributionCache), 16
- getdistributionCache,KakizawaB1-method (getdistributionCache), 16
- getdistributionCache,KakizawaB2-method (getdistributionCache), 16
- getdistributionCache,KakizawaB3-method (getdistributionCache), 16
- getdistributionCache,MacroBetaChen99Kernel-method (getdistributionCache), 16
- getdistributionCache,MacroBetaHirukawaJLNKernel-method (getdistributionCache), 16
- getdistributionCache,MacroBetaHirukawaTSKernel-method (getdistributionCache), 16

- getdistributionCache, MicroBetaChen99Kernel-method (getdistributionCache), 16
- getdistributionCache, Muller91BoundaryKernel-method (getdistributionCache), 16
- getdistributionCache, Muller94BoundaryKernel-method (getdistributionCache), 16
- getdistributionCache, NoBoundaryKernel-method (getdistributionCache), 16
- getdistributionCache, NormalizedBoundaryKernel-method (getdistributionCache), 16
- getdistributionCache, Vitale-method (getdistributionCache), 16
- getdistributionCache-methods (getdistributionCache), 16
- getgamma, 17, 31
- getgamma, KakizawaB1-method (getgamma), 17
- getgamma-methods (getgamma), 17
- getM, 8, 17
- getm, 8, 18, 31, 34, 36, 62
- getM, BernsteinPolynomials-method (getM), 17
- getm, BernsteinPolynomials-method (getm), 18
- getM, BrVitale-method (getM), 17
- getm, Vitale-method (getm), 18
- getM-methods (getM), 17
- getm-methods (getm), 18
- getmodified, 11, 18, 21, 23, 39, 42, 44, 47
- getmodified, Chen99Kernel-method (getmodified), 18
- getmodified, HirukawaJLNKernel-method (getmodified), 18
- getmodified, HirukawaTSKernel-method (getmodified), 18
- getmodified, MacroBetaChen99Kernel-method (getmodified), 18
- getmodified, MacroBetaHirukawaJLNKernel-method (getmodified), 18
- getmodified, MicroBetaChen99Kernel-method (getmodified), 18
- getmodified-methods (getmodified), 18
- getmu, 19, 26, 29, 50, 53, 55, 57
- getmu, BoundaryKernel-method (getmu), 19
- getmu, NoBoundaryKernel-method (getmu), 19
- getmu-methods (getmu), 19
- getSubclasses, 19
- gplot, 20
- gplot, BernsteinPolynomials-method (gplot), 20
- gplot, BoundaryKernel-method (gplot), 20
- gplot, BoundedDensity-method (gplot), 20
- gplot, BrVitale-method (gplot), 20
- gplot, Chen99Kernel-method (gplot), 20
- gplot, HirukawaJLNKernel-method (gplot), 20
- gplot, HirukawaTSKernel-method (gplot), 20
- gplot, JonesCorrectionMuller91BoundaryKernel-method (gplot), 20
- gplot, JonesCorrectionMuller94BoundaryKernel-method (gplot), 20
- gplot, KakizawaB1-method (gplot), 20
- gplot, KakizawaB2-method (gplot), 20
- gplot, KakizawaB3-method (gplot), 20
- gplot, KernelDensity-method (gplot), 20
- gplot, list-method (gplot), 20
- gplot, MacroBetaChen99Kernel-method (gplot), 20
- gplot, MacroBetaHirukawaJLNKernel-method (gplot), 20
- gplot, MacroBetaHirukawaTSKernel-method (gplot), 20
- gplot, MicroBetaChen99Kernel-method (gplot), 20
- gplot, Muller91BoundaryKernel-method (gplot), 20
- gplot, Muller94BoundaryKernel-method (gplot), 20
- gplot, NoBoundaryKernel-method (gplot), 20
- gplot, NormalizedBoundaryKernel-method (gplot), 20
- gplot, Vitale-method (gplot), 20
- gplot-methods (gplot), 20
- HirukawaJLNKernel, 20, 22
- hirukawaJLNKernel, 20, 22
- HirukawaJLNKernel-class (HirukawaJLNKernel), 20
- hirukawaJLNKernel-generator (hirukawaJLNKernel), 22
- HirukawaTSKernel, 14, 23, 25
- hirukawaTSKernel, 23, 24
- HirukawaTSKernel-class (HirukawaTSKernel), 23

- hirukawaTSKernel-generator
(hirukawaTSKernel), 24
- JonesCorrectionMuller91BoundaryKernel,
25, 28
- jonesCorrectionMuller91BoundaryKernel,
25, 27
- JonesCorrectionMuller91BoundaryKernel-class
(JonesCorrectionMuller91BoundaryKernel),
25
- jonesCorrectionMuller91BoundaryKernel-generator
(jonesCorrectionMuller91BoundaryKernel),
27
- JonesCorrectionMuller94BoundaryKernel,
28, 30
- jonesCorrectionMuller94BoundaryKernel,
28, 30
- JonesCorrectionMuller94BoundaryKernel-class
(JonesCorrectionMuller94BoundaryKernel),
28
- jonesCorrectionMuller94BoundaryKernel-generator
(jonesCorrectionMuller94BoundaryKernel),
30
- KakizawaB1, 16, 17, 30, 33
- kakizawaB1, 31, 32
- kakizawaB1, KakizawaB1-method
(kakizawaB1), 32
- KakizawaB1-class (KakizawaB1), 30
- kakizawaB1-methods (kakizawaB1), 32
- KakizawaB2, 16, 33, 35
- kakizawaB2, 33, 34
- kakizawaB2, KakizawaB2-method
(kakizawaB2), 34
- KakizawaB2-class (KakizawaB2), 33
- kakizawaB2-methods (kakizawaB2), 34
- KakizawaB3, 16, 35, 37
- kakizawaB3, 35, 37
- kakizawaB3, KakizawaB3-method
(kakizawaB3), 37
- KakizawaB3-class (KakizawaB3), 35
- kakizawaB3-methods (kakizawaB3), 37
- launchApp, 37
- lines, 38
- lines, BernsteinPolynomials-method
(lines), 38
- lines, BoundaryKernel-method (lines), 38
- lines, BoundedDensity-method (lines), 38
- lines, BrVitale-method (lines), 38
- lines, Chen99Kernel-method (lines), 38
- lines, HirukawaJLNKernel-method (lines),
38
- lines, HirukawaTSKernel-method (lines),
38
- lines, JonesCorrectionMuller91BoundaryKernel-method
(lines), 38
- lines, JonesCorrectionMuller94BoundaryKernel-method
(lines), 38
- lines, KakizawaB1-method (lines), 38
- lines, KakizawaB2-method (lines), 38
- lines, KakizawaB3-method (lines), 38
- lines, KernelDensity-method (lines), 38
- lines, MacroBetaChen99Kernel-method
(lines), 38
- lines, MacroBetaHirukawaJLNKernel-method
(lines), 38
- lines, MacroBetaHirukawaTSKernel-method
(lines), 38
- lines, MicroBetaChen99Kernel-method
(lines), 38
- lines, Muller91BoundaryKernel-method
(lines), 38
- lines, Muller94BoundaryKernel-method
(lines), 38
- lines, NoBoundaryKernel-method (lines),
38
- lines, NormalizedBoundaryKernel-method
(lines), 38
- lines, Vitale-method (lines), 38
- lines-methods (lines), 38
- MacroBetaChen99Kernel, 38, 40
- macroBetaChen99Kernel, 38, 40
- MacroBetaChen99Kernel-class
(MacroBetaChen99Kernel), 38
- macroBetaChen99Kernel-generator
(macroBetaChen99Kernel), 40
- MacroBetaHirukawaJLNKernel, 41, 43
- macroBetaHirukawaJLNKernel, 41, 42
- MacroBetaHirukawaJLNKernel-class
(MacroBetaHirukawaJLNKernel),
41
- macroBetaHirukawaJLNKernel-generator
(macroBetaHirukawaJLNKernel),
42
- MacroBetaHirukawaTSKernel, 14, 43, 46
- macroBetaHirukawaTSKernel, 43, 45

- MacroBetaHirukawaTSKernel-class
(MacroBetaHirukawaTSKernel), 43
- macroBetaHirukawaTSKernel-generator
(macroBetaHirukawaTSKernel), 45
- MicroBetaChen99Kernel, 46, 48
- microBetaChen99Kernel, 46, 48
- MicroBetaChen99Kernel-class
(MicroBetaChen99Kernel), 46
- microBetaChen99Kernel-generator
(microBetaChen99Kernel), 48
- mise, 48
- Muller91BoundaryKernel, 49, 52
- muller91BoundaryKernel, 25, 49, 51
- Muller91BoundaryKernel-class
(Muller91BoundaryKernel), 49
- muller91BoundaryKernel-generator
(muller91BoundaryKernel), 51
- Muller94BoundaryKernel, 52, 54
- muller94BoundaryKernel, 28, 52, 53
- Muller94BoundaryKernel-class
(Muller94BoundaryKernel), 52
- muller94BoundaryKernel-generator
(muller94BoundaryKernel), 53

- NoBoundaryKernel, 54, 56
- noBoundaryKernel, 54, 56
- NoBoundaryKernel-class
(NoBoundaryKernel), 54
- noBoundaryKernel-generator
(noBoundaryKernel), 56
- NormalizedBoundaryKernel, 56, 59
- normalizedBoundaryKernel, 57, 58
- NormalizedBoundaryKernel-class
(NormalizedBoundaryKernel), 56
- normalizedBoundaryKernel-generator
(normalizedBoundaryKernel), 58

- par, 38, 59
- plot, 6, 8, 11, 21, 23, 26, 29, 31, 33, 36, 39,
41, 44, 46, 50, 52, 55, 57, 59, 62
- plot, BernsteinPolynomials-method
(plot), 59
- plot, BoundaryKernel-method (plot), 59
- plot, BoundedDensity-method (plot), 59
- plot, BrVitale-method (plot), 59
- plot, Chen99Kernel-method (plot), 59
- plot, HirukawaJLNKernel-method (plot), 59
- plot, HirukawaTSKernel-method (plot), 59
- plot, JonesCorrectionMuller91BoundaryKernel-method
(plot), 59
- plot, JonesCorrectionMuller94BoundaryKernel-method
(plot), 59
- plot, KakizawaB1-method (plot), 59
- plot, KakizawaB2-method (plot), 59
- plot, KakizawaB3-method (plot), 59
- plot, KernelDensity-method (plot), 59
- plot, MacroBetaChen99Kernel-method
(plot), 59
- plot, MacroBetaHirukawaJLNKernel-method
(plot), 59
- plot, MacroBetaHirukawaTSKernel-method
(plot), 59
- plot, MicroBetaChen99Kernel-method
(plot), 59
- plot, Muller91BoundaryKernel-method
(plot), 59
- plot, Muller94BoundaryKernel-method
(plot), 59
- plot, NoBoundaryKernel-method (plot), 59
- plot, NormalizedBoundaryKernel-method
(plot), 59
- plot, Vitale-method (plot), 59
- plot-methods (plot), 59

- quantile, 6, 8, 11, 21, 23, 26, 29, 31, 33, 36,
39, 41, 44, 46, 50, 52, 55, 57, 59, 62
- quantile, BoundaryKernel-method
(quantile), 59
- quantile, BoundedDensity-method
(quantile), 59
- quantile, BrVitale-method (quantile), 59
- quantile, Chen99Kernel-method
(quantile), 59
- quantile, HirukawaJLNKernel-method
(quantile), 59
- quantile, HirukawaTSKernel-method
(quantile), 59
- quantile, JonesCorrectionMuller91BoundaryKernel-method
(quantile), 59
- quantile, JonesCorrectionMuller94BoundaryKernel-method
(quantile), 59
- quantile, KakizawaB1-method (quantile),
59
- quantile, KakizawaB2-method (quantile),
59
- quantile, KakizawaB3-method (quantile),
59

- quantile, MacroBetaChen99Kernel-method (quantile), [59](#)
- quantile, MacroBetaHirukawaJLNKernel-method (quantile), [59](#)
- quantile, MacroBetaHirukawaTSKernel-method (quantile), [59](#)
- quantile, MicroBetaChen99Kernel-method (quantile), [59](#)
- quantile, Muller91BoundaryKernel-method (quantile), [59](#)
- quantile, Muller94BoundaryKernel-method (quantile), [59](#)
- quantile, NoBoundaryKernel-method (quantile), [59](#)
- quantile, NormalizedBoundaryKernel-method (quantile), [59](#)
- quantile, Vitale-method (quantile), [59](#)
- quantile-methods (quantile), [59](#)

- rsample, [6](#), [8](#), [11](#), [21](#), [23](#), [26](#), [29](#), [31](#), [33](#), [36](#), [39](#), [41](#), [44](#), [46](#), [50](#), [52](#), [55](#), [57](#), [60](#), [62](#)
- rsample, BoundaryKernel-method (rsample), [60](#)
- rsample, BoundedDensity-method (rsample), [60](#)
- rsample, BrVitale-method (rsample), [60](#)
- rsample, Chen99Kernel-method (rsample), [60](#)
- rsample, HirukawaJLNKernel-method (rsample), [60](#)
- rsample, HirukawaTSKernel-method (rsample), [60](#)
- rsample, JonesCorrectionMuller91BoundaryKernel-method (rsample), [60](#)
- rsample, JonesCorrectionMuller94BoundaryKernel-method (rsample), [60](#)
- rsample, KakizawaB1-method (rsample), [60](#)
- rsample, KakizawaB2-method (rsample), [60](#)
- rsample, KakizawaB3-method (rsample), [60](#)
- rsample, MacroBetaChen99Kernel-method (rsample), [60](#)
- rsample, MacroBetaHirukawaJLNKernel-method (rsample), [60](#)
- rsample, MacroBetaHirukawaTSKernel-method (rsample), [60](#)
- rsample, MicroBetaChen99Kernel-method (rsample), [60](#)
- rsample, Muller91BoundaryKernel-method (rsample), [60](#)
- rsample, Muller94BoundaryKernel-method (rsample), [60](#)
- rsample, NoBoundaryKernel-method (rsample), [60](#)
- rsample, NormalizedBoundaryKernel-method (rsample), [60](#)
- rsample, Vitale-method (rsample), [60](#)
- rsample-methods (rsample), [60](#)

- suicide.r, [60](#)

- tgaussian, [61](#)
- tuna, [61](#)
- tuna.r, [61](#)

- Vitale, [62](#), [64](#)
- vitale, [62](#), [63](#)
- Vitale-class (Vitale), [62](#)
- vitale-generator (vitale), [63](#)