Package ‘AsyK’

May 6, 2026

Type Package

Title Kernel Density Estimation

Version 1.5.6

Maintainer Javaria Ahmad Khan <jakhan0@yahoo.com>

Description A collection of functions related to density estimation by using Chen's (2000) idea. Mean Squared Errors (MSE) are calculated for estimated curves. For this purpose, R functions allow the distribution to be Gamma, Exponential or Weibull. For details see Chen (2000), Scaillet (2004) <doi:10.1080/10485250310001624819> and Khan and Akbar.

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.2

URL https://CRAN.R-project.org/package=AsyK

NeedsCompilation no

Author Javaria Ahmad Khan [aut, cre],
Atif Akbar. [aut]

Repository CRAN

Date/Publication 2025-04-29 17:00:06 UTC

Contents

AsyK-package *AsyK*

Description

A collection of functions related to density estimation by using Chen's (2000) idea. For observing estimated values see `Laplace` and `RIG`. Plots by using these kernels can be drawn by `plot.Laplace` and `plot.RIG`. Mean squared errors (MSE) can be calculated by `mse`. Here we also present a normal scale rule bandwidth which is given by Silverman (1986) for non-normal data.

Details

Kernel Density Estimation

Author(s)

Javaria Ahmad Khan, Atif Akbar.

See Also

Useful links:

- https://CRAN.R-project.org/package=AsyK

Laplace *Estimate Density Values by Laplace kernel*

Description

Estimated Kernel density values by using Laplace Kernel.

Usage

```
Laplace(x = NULL, y, k = NULL, h = NULL)
```

Arguments

x	scheme for generating grid points
y	a numeric vector of positive values.
k	gird points.
h	the bandwidth

Details

Laplace kernel is developed by Khan and Akbar. Kernel is developed by using Chen's idea. Laplace kernel is;

$$K_{Laplace\left(x,h^{\frac{1}{2}}\right)}(u) = \frac{1}{2\sqrt{h}} exp\left(-\frac{|u-x|}{\sqrt{h}}\right)$$

Value

x	grid points
y	estimated values of density

Author(s)

Javaria Ahmad Khan, Atif Akbar.

References

Khan, J. A.; Akbar, A. Density Estimation by Laplace Kernel. *Working paper, Department of Statistics, Bahauddin Zakariya University, Multan, Pakistan.*

See Also

To examine Laplace density plot see plot.Laplace and for Mean Squared Error mse. Similarly, for RIG kernel RIG.

Examples

```
#Data can be simulated or real data
## Number of grid points "k" should be at least equal to the data size.
### If user define the generating scheme of gridpoints than number of gridpoints should
####be equal or greater than "k"
###### otherwise NA will be produced.
y <- rexp(100, 1)
xx <- seq(min(y) + 0.05, max(y), length = 100)
h <- 2
den <- Laplace(x = xx, y = y, k = 200, h = h)

##If scheme for generating gridpoints is unknown
y <- rexp(50, 1)
h <- 3
den <- Laplace(y = y, k = 90, h = h)

##If user do not mention the number of grid points
y <- rexp(23, 1)
xx <- seq(min(y) + 0.05, max(y), length = 90)

## Not run:
#any bandwidth can be used
require(KernSmooth)
h <- dpik(y)
den <- Laplace(x = xx, y = y, h = h)
```

```
## End(Not run)

#if bandwidth is missing
y <- rexp(100, 1)
xx <- seq(min(y) + 0.05, max(y), length = 100)
den <- Laplace(x = xx, y = y, k = 90)
```

---

mse *Calculate Mean Squared Error(MSE) by using different Kernels*

Description

This function calculates the mean squared error (MSE) by using user specified kernel.This function is same as provided in package "DELTD". For details see https://CRAN.R-project.org/package=DELTD.

Usage

```
mse(kernel, type)
```

Arguments

kernel	type of kernel which is to be used
type	mention distribution of vector.If exponential distribution then use "Exp". If use gamma distribution then use "Gamma".If Weibull distribution then use "Weibull".

Value

Mean Squared Error (MSE)

Author(s)

Javaria Ahmad Khan, Atif Akbar.

References

https://CRAN.R-project.org/package=DELTD

See Also

This is also available in **DELTD**

Examples

```
y <- rexp(100, 1)
xx <- seq(min(y) + 0.05, max(y), length = 500)
h <- 2
gr <- Laplace(x = xx, y = y, k = 200, h = h)
mse(kernel = gr, type = "Exp")
## if distribution is other than mentioned \code{type} is used then NaN will be produced.
## Not run:
mse(kernel = gr, type ="Beta")

## End(Not run)
```

NSR *Calculate Bandwidth proposed by Silverman for non-normal data.*

Description

Calculate Bandwidth proposed by Silverman for non-normal data.

Usage

```
NSR(y)
```

Arguments

y	a numeric vector of positive values.

Value

h

Author(s)

Javaria Ahmad Khan, Atif Akbar.

References

Silverman, B. W. 1986. *Density Estimation*. Chapman & Hall/ CRC, London.

Examples

```
y <- rexp(10, 1)
 NSR(y)
```

`plot.Laplace` *Density Plot by Laplace kernel*

Description

Plot density by using Laplace Kernel.

Usage

```
## S3 method for class 'Laplace'
plot(x, ...)
```

Arguments

`x`	an object of class "Laplace"
`...`	Not presently used in this implementation

Value

nothing

Author(s)

Javaria Ahmad Khan, Atif Akbar.

References

Khan, J. A.; Akbar, A. Density Estimation by Laplace Kernel. *Working paper, Department of Statistics, Bahauddin Zakariya University, Multan, Pakistan.*

See Also

To examine Laplace estimated values for density see `Laplace` and for Mean Squared Error `mse`. Similarly, for plot of Laplace kernel `plot.RIG`.

Examples

```
y <- rexp(100, 1)
h <- 0.79 * IQR(y) * length(y) ^ (-1/5)
xx <- seq(min(y) + 0.05, max(y), length = 100)
den <- Laplace(x = xx, y = y, k = 100, h = h)
plot(den, type = "l")


##other details can also be added
y <- rexp(100, 1)
h <- 0.79 * IQR(y) * length(y) ^ (-1/5)
den <- Laplace(x = xx, y = y, k = 100, h = h)
plot(den, type = "s", ylab = "Density Function", lty = 1, xlab = "Time")
```

```
## To add true density along with estimated
d1 <- density(y, bw = h)
lines(d1, type = "p", col = "red")
legend("topright", c("Real Density", "Density by RIG Kernel"),
col = c("red", "black"), lty = c(1, 2))
```

---

plot.RIG *Density Plot by Reciprocal Inverse Gaussian kernel*

Description

Plot density by using Reciprocal Inverse Gaussian Kernel.

Usage

```
## S3 method for class 'RIG'
plot(x, ...)
```

Arguments

x	an object of class "RIG"
...	Not presently used in this implementation

Value

nothing

Author(s)

Javaria Ahmad Khan, Atif Akbar.

References

Scaillet, O. 2004. Density estimation using inverse and reciprocal inverse Gaussian kernels. *Non-parametric Statistics*, **16**, 217-226.

See Also

To examine RIG estimated values for density see `RIG` and for Mean Squared Error `mse`. Similarly, for plot of Laplace kernel `plot.Laplace`.

Examples

```
y <- rexp(200, 1)
h <- 0.79 * IQR(y) * length(y) ^ (-1/5)
xx <- seq(min(y) + 0.05, max(y), length = 200)
den <- RIG(x = xx, y = y, k = 200, h = h)
plot(den, type = "l")


##other details can also be added
y <- rexp(200, 1)
h <- 0.79 * IQR(y) * length(y) ^ (-1/5)
den <- RIG(x = xx, y = y, k = 200, h = h)
plot(den, type = "s", ylab = "Density Function", lty = 1, xlab = "Time")

## To add true density along with estimated
d1 <- density(y, bw = h)
lines(d1, type = "p", col = "red")
legend("topright", c("Real Density", "Density by RIG Kernel"),
col = c("red", "black"), lty = c(1, 2))
```

RIG *Estimated Density Values by Reciprocal Inverse Gaussian kernel*

Description

Estimated Kernel density values by using Reciprocal Inverse Gaussian Kernel.

Usage

```
RIG(x = NULL, y, k = NULL, h = NULL)
```

Arguments

x	scheme for generating grid points
y	a numeric vector of positive values.
k	gird points.
h	the bandwidth

Details

Scaillet 2003. proposed Reciprocal Inverse Gaussian kerenl. He claimed that his proposed kernel share the same properties as those of gamma kernel estimator.

$$K_{RIG\left(\ln ax4\ln\left(\frac{1}{h}\right)\right)}(y) = \frac{1}{\sqrt{2\pi y}} exp\left[-\frac{x-h}{2h}\left(\frac{y}{x-h} - 2 + \frac{x-h}{y}\right)\right]$$

Value

x	grid points
y	estimated values of density

Author(s)

Javaria Ahmad Khan, Atif Akbar.

References

Scaillet, O. 2004. Density estimation using inverse and reciprocal inverse Gaussian kernels. *Nonparametric Statistics*, **16**, 217-226.

See Also

To examine RIG density plot see `plot.RIG` and for Mean Squared Error `mse`. Similarly, for Laplace kernel `Laplace`.

Examples

```
#Data can be simulated or real data
## Number of grid points "k" should be at least equal to the data size.
### If user define the generating scheme of gridpoints than number of gridpoints should
####be equal or greater than "k"
###### otherwise NA will be produced.
y <- rexp(100, 1)
xx <- seq(min(y) + 0.05, max(y), length = 100)
h <- 2
den <- RIG(x = xx, y = y, k = 200, h = h)

##If scheme for generating gridpoints is unknown
y <- rexp(50, 1)
h <- 3
den <- RIG(y = y, k = 90, h = h)

## Not run:
##If user do not mention the number of grid points
y <- rexp(23, 1)
xx <- seq(min(y) + 0.05, max(y), length = 90)
#any bandwidth can be used
require(KernSmooth)
h <- dpik(y)
den <- RIG(x = xx, y = y, h = h)

## End(Not run)
#if bandwidth is missing
y <- rexp(100, 1)
xx <- seq(min(y) + 0.05, max(y), length = 100)
den <- RIG(x = xx, y = y, k = 90)
```

Index