Package ‘asymmetry.measures’

July 22, 2020

Encoding UTF-8
Type Package
Title Asymmetry Measures for Probability Density Functions
Version 0.2
URL
Date 2020-07-22
Maintainer Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>
Depends R (>= 3.5.0)
Imports stats, sn, skewt, gamlss.dist
Description Provides functions and examples for the weak and strong density asymmetry measures in the articles: "A measure of asymmetry", Patil, Patil and Bagkavos (2012) <doi:10.1007/s00362-011-0401-6> and "A measure of asymmetry based on a new necessary and sufficient condition for symmetry", Patil, Bagkavos and Wood (2014) <doi:10.1007/s13171-013-0034-z>. The measures provided here are useful for quantifying the asymmetry of the shape of a density of a random variable. The package facilitates implementation of the measures which are applicable in a variety of fields including e.g. probability theory, statistics and economics.
License GPL (>= 2)
NeedsCompilation no
RoxygenNote 6.1.0
LazyData true
Author Dimitrios Bagkavos [aut, cre],
Lucia Gamez [aut]
Repository CRAN
Date/Publication 2020-07-22 05:00:03 UTC

R topics documented:

d.sample ................................................................. 2
edf ................................................................. 5
Switch between a range of probability density functions.

**Description**

Returns the user-specified probability density function out of a range of available options evaluated at selected grid points.

**Usage**

```r
d.sample(s, dist, p1, p2)
```

**Arguments**

- **s**: A scalar or vector: the x-axis grid points where the probability density function will be evaluated.
- **dist**: Character string, used as a switch to the user selected distribution function (see details below).
- **p1**: A scalar. Parameter 1 (vector or object) of the selected density.
- **p2**: A scalar. Parameter 2 (vector or object) of the selected density.
Details

Based on user-specified argument `dist`, the function returns the value of the probability density function at `s`.

Supported distributions (along with the corresponding `dist` values) are:

- **weib**: The weibull distribution is implemented as
  \[
  f(s; p_1, p_2) = \frac{p_1}{p_2} \left( \frac{s}{p_2} \right)^{p_1-1} \exp \left\{ - \left( \frac{s}{p_2} \right)^{p_1} \right\}
  \]
  with \( s \geq 0 \) where \( p_1 \) is the shape parameter and \( p_2 \) the scale parameter.

- **lognorm**: The lognormal distribution is implemented as
  \[
  f(s) = \frac{1}{p_2 s \sqrt{2\pi}} e^{-\frac{(\log s - p_1)^2}{2p_2^2}}
  \]
  where \( p_1 \) is the mean and \( p_2 \) is the standard deviation of the distribution.

- **norm**: The normal distribution is implemented as
  \[
  f(s) = \frac{1}{p_2 \sqrt{2\pi}} e^{-\frac{(s - p_1)^2}{2p_2^2}}
  \]
  where \( p_1 \) is the mean and the \( p_2 \) is the standard deviation of the distribution.

- **uni**: The uniform distribution is implemented as
  \[
  f(s) = \frac{1}{p_2 - p_1}
  \]
  for \( p_1 \leq s \leq p_2 \).

- **cauchy**: The cauchy distribution is implemented as
  \[
  f(s) = \frac{1}{\pi p_2} \left\{ 1 + \left( \frac{s - p_1}{p_2} \right)^2 \right\}^{-1}
  \]
  where \( p_1 \) is the location parameter and \( p_2 \) the scale parameter.

- **fnorm**: The half normal distribution is implemented as
  \[
  f(s) = 2f(s) - 1
  \]
  where
  \[
  f(s) = \frac{1}{sd \sqrt{2\pi}} e^{-\frac{s^2}{2sd^2}},
  \]
  and \( sd = \sqrt{\pi/2}/p_1 \).

- **normmixt**: The normal mixture distribution is implemented as
  \[
  f(s) = p_1 \frac{1}{p_2 \sqrt{2\pi}} e^{-\frac{(s - p_2[1])^2}{2p_2[2]^2}} + (1 - p_1) \frac{1}{p_2[4] \sqrt{2\pi}} e^{-\frac{(s - p_2[3])^2}{2p_2[4]^2}}
  \]
  where \( p_1 \) is a mixture component(scalar) and \( p_2 \) a vector of parameters for the mean and variance of the two mixture components \( p_2 = c(mean1, sd1, mean2, sd2) \).
• **skewnorm**: The skew normal distribution with parameter $p_1$ is implemented as

$$f(s) = 2\phi(s)\Phi(p_1 s)$$

• **fas**: The Fernandez and Steel distribution is implemented as

$$f(s; p_1, p_2) = \frac{2}{p_1 + 1} \left\{ f_t(s/p_1; p_2)I_{s\geq0} + f_t(p_1 s; p_2)I_{s<0} \right\}$$

where $f_t(x; \nu)$ is the p.d.f. of the $t$ distribution with $\nu = 5$ degrees of freedom. $p_1$ controls the skewness of the distribution with values between $(0, +\infty)$ and $p_2$ denotes the degrees of freedom.

• **shash**: The Sinh-Arcsinh distribution is implemented as

$$f(s; \mu, p_1, p_2, \tau) = c e^{-r^2/2} \frac{1}{p_2^2} \sqrt{1 + z^2}$$

where $r = \sinh(\sinh(z) - (-p_1)), c = \cosh(\sinh(z) - (-p_1))$ and $z = ((s - \mu)/p_2)$. $p_1$ is the vector of skewness, $p_2$ is the scale parameter, $\mu = 0$ is the location parameter and $\tau = 1$ the kurtosis parameter.

**Value**

A vector containing the user selected density values at the user specified points $s$.

**Author(s)**

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

**References**


**See Also**

*r.sample*, *q.sample*, *p.sample*

**Examples**

```r
sellected.dens <- "weib" #select Weibull as the density
shape <- 2 # specify shape parameter
scale <- 1 # specify scale parameter
xout <- seq(0.1,5,length=50) #design point where the density is evaluated
d.sample(xout,selected.dens,shape,scale) # calculate density at xout
```
Empirical cumulative distribution function

Description

Empirical (nonparametric) cumulative distribution function for given a random sample.

Usage

edf(xin, xout)

Arguments

- **xin**: A vector of data points - the available sample.
- **xout**: A vector of design points where the distribution function will be estimated.

Details

The empirical distribution function estimator at \( x \) is defined as the number of observations up to \( x \), divided by \( n \), i.e.

\[
F_n(x) = \frac{\#\{X_1, ..., X_n\} \leq x}{n}.
\]

Value

A vector with the estimated distribution function at \( x_{out} \).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation:

Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallordolucia@gmail.com>

References


Examples

```r
x.in <- rexp(200)
x.out <- seq(0.1, 5, length=60)
dist.est <- edf(x.in, x.out)
plot(x.out, dist.est, col="blue", main="Empirical c.d.f.", xlab="x", ylab="probability")
```
Description

Implementation of the Epanechnikov kernel.

Usage

Epanechnikov(x)

Arguments

x  A vector of data points between $-\sqrt{5}$ and $\sqrt{5}$ where the kernel will be evaluated.

Details

Implements:

$$K(u) = \frac{3}{4\sqrt{5}} \left( 1 - \frac{x^2}{5} \right)$$

for $|x| \leq \sqrt{5}$

Value

The value of the kernel at $x$

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References

Kernel Statistics

See Also

IntEpanechnikov
eta.s

**Description**

Returns the strong asymmetry measure $\eta(X)$ of Patil, Bagkavos and Wood (2014).

**Usage**

```r
eta.s(xin, dist, GridLength, p1, p2)
```

**Arguments**

- **xin**: A vector of data points - the available sample.
- **dist**: Character string, specifies selected distribution function.
- **GridLength**: A non-negative number, which will be rounded up if fractional. Desired length of the sequence.
- **p1**: A scalar. Parameter 1 (vector or object) of the selected distribution.
- **p2**: A scalar. Parameter 2 (vector or object) of the selected distribution.

**Details**

Implements

$$\eta(X) = -0.5 \text{sign}(\rho_1) \max |\rho_p + \rho_p^*|$$

with $1/2 \leq p \leq 1$.

Uses maximum likelihood estimates for the unknown functionals in the definition of the measure.

**Value**

Returns a scalar, the value of the strong asymmetry measure $\eta(X)$.

**Author(s)**

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

**References**

eta.s.exact

See Also

eta.w.hat.bc, eta.w.hat, eta.w.breve, eta.w.breve.bc, eta.w.tilde, eta.w.tilde.bc

Examples

```r
selected.dist <- "norm" # select norm as the distribution
m.use <- mean(GDP.Per.head.dist.2005)
sd.use <- sd(GDP.Per.head.dist.2005)
grid <- 50
s.use <- GDP.Per.head.dist.1995
eta.s(GDP.Per.head.dist.2005, selected.dist, grid, m.use, sd.use)
```

eta.s.exact  Strong asymmetry measure \eta(X).

Description

Returns the strong asymmetry measure \eta(X) of Patil, Bagkavos and Wood (2014).

Usage

```r
eta.s.exact(xin, dist, GridLength, p1, p2)
```

Arguments

- `xin`: A vector of data points - the available sample.
- `dist`: Character string, specifies selected distribution function.
- `GridLength`: A non-negative number, which will be rounded up if fractional. Desired length of the sequence.
- `p1`: A scalar. Parameter 1 (vector or object) of the selected distribution.
- `p2`: A scalar. Parameter 2 (vector or object) of the selected distribution.

Details

Implements

\[
\eta(X) = -0.5 \text{sign}(\rho_1) \max |\rho_p + \rho_p^*|
\]

with \(1/2 \leq p \leq 1\) This version uses exact p.d.f. and c.d.f. evaluation and not estimates of the unknown functionals.

Value

Returns a scalar, the value of the strong asymmetry measure \eta(X).
Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

eta.w.hat.bc, eta.w.hat, eta.w.breve, eta.w.breve.bc, eta.w.tilde, eta.w.tilde.bc, eta.s

Examples

selected.dist <- "norm" #select norm as the distribution
m.use <- 2
sd.use<- 2
grid <- 50
s.use<- rnorm(100)
eta.s.exact(s.use,selected.dist,grid,m.use,sd.use) # calculate eta.s at xout

---

eta.w.breve

Asymmetry coefficient ˘η

Description

Implements the asymmetry coefficient ˘η of Patil, Patil and Bagkavos (2012).

Usage

eta.w.breve(xin, kfun)

Arguments

xin A vector of data points - the available sample.
kfun The kernel to use in the density estimate.
Given a sample $X_1, X_2, \ldots, X_n$ from a continuous density function $f(x)$ and distribution function $F(x)$, $\hat{\eta}$ is defined by

$$\hat{\eta} = -\frac{\sum_{i=1}^{n} U_i W_i - n\bar{U}\bar{W}}{\sqrt{(\sum_{i=1}^{n} U_i^2 - n\bar{U}^2)(\sum_{i=1}^{n} W_i^2 - n\bar{W}^2)}}$$

where

$$U_i = \hat{f}(X_i), \ W_i = F_n(X_i), \ \bar{U} = n^{-1} \sum_{i=1}^{n} U_i, \ \bar{W} = n^{-1} \sum_{i=1}^{n} W_i.$$ 

Value

Returns a scalar, the estimate of $\hat{\eta}$.

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

eta.w.hat.bc, eta.w.hat, eta.w.breve.bc, eta.w.tilde.bc, eta.w.tilde.bc

Examples

eta.w.breve(GDP.Per.head.dist.1995, Epanechnikov)

0.329707 #estimate of etabreve
eta.w.breve.bc

Asymmetry coefficient \( \hat{\eta} \) using boundary correction

Description

Implements the asymmetry coefficient \( \hat{\eta} \) of Patil, Patil and Bagkavos (2012).

Usage

\[
\text{eta.w.breve.bc}(\text{xin}, \text{kfun})
\]

Arguments

- \text{xin} A vector of data points - the available sample.
- \text{kfun} The kernel to use in the density estimate.

Details

Given a sample \( X_1, X_2, \ldots, X_n \) from a continuous density function \( f(x) \) and distribution function \( F(x) \). \( \hat{\eta} \) is defined by

\[
\hat{\eta} = -\frac{\sum_{i=1}^{n} U_i W_i - n\bar{U}\bar{W}}{\sqrt{\left(\sum_{i=1}^{n} U_i^2 - n\bar{U}^2\right)\left(\sum_{i=1}^{n} W_i^2 - n\bar{W}^2\right)}}
\]

where

\[
U_i = \hat{f}(X_i), \quad W_i = F_n(X_i), \quad \bar{U} = n^{-1} \sum_{i=1}^{n} U_i, \quad \bar{W} = n^{-1} \sum_{i=1}^{n} W_i.
\]

\text{eta.w.breve.bc} uses reflection to correct the boundary bias of the kernel density estimate \( \text{kde} \)

Value

Returns a scalar, the estimate of \( \hat{\eta} \).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References

See Also

\texttt{eta.w.hat.bc, eta.w.hat, eta.w.breve, eta.w.tilde, eta.w.tilde.bc}

Examples

\texttt{eta.w.breve.bc(GDP.Per.head.dist.1995,Epanechnikov)}
\texttt{0.329707} #estimate of etabreve

\texttt{eta.w.hat} \hspace{1cm} \textit{Asymmetry coefficient } \hat{\eta}

Description

Implements the asymmetry coefficient \( \hat{\eta} \) of Patil, Patil and Bagkavos (2012).

Usage

\texttt{eta.w.hat(xin, kfun)}

Arguments

\begin{itemize}
  \item \texttt{xin} \hspace{1cm} A vector of data points - the available sample.
  \item \texttt{kfun} \hspace{1cm} The kernel to use in the density estimate.
\end{itemize}

Details

Given a sample \( X_1, X_2, \ldots, X_n \) from a continuous density function \( f(x) \) and distribution function \( F(x) \), \( \hat{\eta} \) is defined by

\[
\hat{\eta} = -\frac{\sum_{i=1}^{n} U_i V_i - n \bar{U} \bar{V}}{\sqrt{\left(\sum_{i=1}^{n} U_i^2 - n \bar{U}^2\right)\left(\sum_{i=1}^{n} V_i^2 - n \bar{V}^2\right)}}
\]

where

\[
U_i = \hat{f}(X_i), \quad V_i = \hat{F}(X_i), \quad \bar{U} = n^{-1} \sum_{i=1}^{n} U_i, \quad \bar{V} = n^{-1} \sum_{i=1}^{n} V_i.
\]

Value

Returns a scalar, the estimate of \( \hat{\eta} \).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>
\texttt{eta.w.hat.bc}

**References**


**See Also**

\texttt{eta.w.hat.bc,eta.w.breve,eta.w.breve.bc,eta.w.tilde,eta.w.tilde.bc}

**Examples**

\begin{verbatim}
eta.w.hat(GDP.Per.head.dist.1995,Epanechnikov)
0.3463025 #estimate of etahat
\end{verbatim}

\begin{verbatim}
eta.w.hat.bc
Asymmetry coefficient \( \hat{\eta} \) using boundary correction
\end{verbatim}

**Description**

Implements the asymmetry coefficient \( \hat{\eta} \) of Patil, Patil and Bagkavos (2012)

**Usage**

\texttt{eta.w.hat.bc(xin, kfun)}

**Arguments**

- \texttt{xin} A vector of data points - the available sample.
- \texttt{kfun} The kernel to use in the density estimate.

**Details**

Given a sample \( X_1, X_2, \ldots, X_n \) from a continuous density function \( f(x) \) and distribution function \( F(x) \), \( \hat{\eta} \) is defined by

\[
\hat{\eta} = -\frac{\sum_{i=1}^{n} U_i V_i - n \bar{U} \bar{V}}{\sqrt{\left(\sum_{i=1}^{n} U_i^2 - n \bar{U}^2\right)\left(\sum_{i=1}^{n} V_i^2 - n \bar{V}^2\right)}}
\]

where

\[
U_i = \hat{f}(X_i), \quad V_i = \hat{F}(X_i), \quad \bar{U} = n^{-1} \sum_{i=1}^{n} U_i, \quad \bar{V} = n^{-1} \sum_{i=1}^{n} V_i.
\]

\texttt{eta.w.hat.bc} uses reflection to correct the boundary bias issue of the kernel estimate \texttt{kde}.
**Value**

Returns a scalar, the estimate of $\tilde{\eta}$.

**Author(s)**

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

**References**


**See Also**

`eta.w.hat, eta.w.breve, eta.w.breve.bc, eta.w.tilde, eta.w.tilde.bc`

**Examples**

```r
eta.w.hat.bc(GDP.Per.head.dist.1995, Epanechnikov)
0.3463025  # estimate of etahat.bc
```

---

**eta.w.tilde**  
**Asymmetry coefficient $\tilde{\eta}$**

**Description**

Implements the asymmetry coefficient $\tilde{\eta}$ of Patil, Patil and Bagkavos (2012).

**Usage**

```r
eta.w.tilde(xin, kfun)
```

**Arguments**

- `xin` A vector of data points - the available sample.
- `kfun` The kernel to use in the density estimate.

**Details**

Given a sample $X_1, X_2, \ldots, X_n$ from a continuous density function $f(x)$ and distribution function $F(x)$. $\tilde{\eta}$ is defined by

$$
\tilde{\eta} = -\frac{\sum_{i=1}^{n} U_i V_i - (n/2)\bar{U}}{\sqrt{(n/12)(\sum_{i=1}^{n} U_i^2 - n\bar{U}^2)}}
$$

where

$$
U_i = \hat{f}(X_i), \quad V_i = F(X_i), \quad \bar{U} = n^{-1} \sum_{i=1}^{n} U_i, \quad \bar{V} = n^{-1} \sum_{i=1}^{n} V_i.
$$
**Value**

Returns a scalar, the estimate of \( \tilde{\eta} \).

**Author(s)**

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

**References**


**See Also**

`eta.w.hat.bc`, `eta.w.hat`, `eta.w.breve.bc`, `eta.w.breve`, `eta.w.tilde.bc`

**Examples**

```r
eta.w.tilde(GDP.Per.head.dist.1995, Epanechnikov)
0.3333485  # estimate of etatile
```

**Description**

Implements the asymmetry coefficient \( \tilde{\eta} \) of Patil, Patil and Bagkavos (2012).

**Usage**

```r
eta.w.tilde.bc(xin, kfun)
```

**Arguments**

- `xin`: A vector of data points - the available sample.
- `kfun`: The kernel to use in the density estimate.
Details

Given a sample \( X_1, X_2, \ldots, X_n \) from a continuous density function \( f(x) \) and distribution function \( F(x) \), \( \tilde{\eta} \) is defined by

\[
\tilde{\eta} = -\frac{\sum_{i=1}^{n} U_i V_i - (n/2)\bar{U}}{\sqrt{(n/12)\sum_{i=1}^{n} U_i^2 - n\bar{U}^2}}
\]

where

\[
U_i = \hat{f}(X_i), \quad V_i = F(X_i), \quad \bar{U} = n^{-1} \sum_{i=1}^{n} U_i, \quad \bar{V} = n^{-1} \sum_{i=1}^{n} V_i.
\]

\( \text{eta.w.tilde.bc} \) uses reflection to correct the boundary bias of \( \text{kde} \).

Value

Returns a scalar, the estimate of \( \tilde{\eta} \).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

\texttt{eta.w.hat.bc,eta.w.hat,eta.w.breve.bc,eta.w.breve,eta.w.tilde}

Examples

\texttt{eta.w.tilde.bc(GDP.Per.head.dist.1995,Epanechnikov)}
0.3333485 #estimate of etatile.bc

---

\texttt{GDP.Per.head.dist.1995}

\textit{annual Gross Domestic Product (GDP) per head across 15 European Union (EU) countries}

---

Description

Contains values of the GDP/head distribution of 216 EU regions (the so called NUTS-2 level of the Eurostat categorization of territories within the EU for the year 1995.)
GDP.Per.head.dist.2005

Usage
GDP.Per.head.dist.1995

Format
A vector with 184 values of the GDP/head distribution for 1995.

Source

References

See Also
GDP.Per.head.dist.2005

GDP.Per.head.dist.2005
annual Gross Domestic Product (GDP) per head across 15 European Union (EU) countries

Description
Contains values of the GDP/head distribution of 216 EU regions (the so called NUTS-2 level of the Eurostat categorization of territories within the EU for the year 2005.

Usage
GDP.Per.head.dist.1995

Format
A vector with 184 values of the GDP/head distribution for 2005.

Source

References
See Also

GDP.Per.head.dist.1995

---

**IntEpanechnikov**  
*Integrated Epanechnikov function*

**Description**

Implements the Integrated Epanechnikov kernel.

**Usage**

`IntEpanechnikov(x)`

**Arguments**

- `x`  
  A vector of design points with values from $-\sqrt{5}$ to $\sqrt{5}$.

**Details**

Implements:

\[
K(u) = \int_{-\infty}^{u} \frac{3}{4\sqrt{5}} \left(1 - \frac{x^2}{5}\right) \, dx
\]

for $|x| \leq \sqrt{5}$

**Value**

The value of the integrated kernel function at the user designated points.

**Author(s)**

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

**References**

Kernel Statistics

**See Also**

Epanechnikov
Description

Classical univariate integrated kernel density estimator

Usage

IntKde(xin, xout, h, kfun)

Arguments

xin A vector of data points - the available sample size.
xout grid points where the distribution function will be estimated.
h The bandwidth parameter. Defaults to $3.572 \times \sigma \times n^{-1/3}$ according to Bowman et al.(1998).
kfun The kernel to use in the distribution function estimate.

Details

It implements the classical density integrated kernel estimator.

Let $X_1, X_2, \ldots, X_n$ be a univariate independent and identically distributed sample drawn from some unknown distribution function $F$. Its kernel density estimator is

$$
\hat{F}(x) = n^{-1} \sum_{i=1}^{n} K \{(x - X_i)h^{-1}\}
$$

where $K$ is an integrated kernel, and $h > 0$ is a smoothing parameter called the bandwidth.

Value

Returns a vector with the estimate of the distribution function at the user specified grid points.

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References

See Also

bw.nrd, bw.nrd0, bw.ucv, bw.bcv

Examples

x.in <- rnorm(100)
x.out <- seq(-3.4,3.4,length=60)
kernl <- IntEpanechnikov
dist.est <- IntKde(xin=x.in,xout=x.out,kfun=kernl)
plot(x.out,dist.est, type="l", col="red", main="Kernel c.d.f. estimator")

kde

Kernel density estimator.

Description

Classical univariate kernel density estimator.

Usage

kde(xin, xout, h, kfun)

Arguments

xin A vector of data points. Missing values not allowed.
xout A vector of grid points at which the estimate will be calculated.
h A scalar, the bandwidth to use in the estimate, e.g. bw.nrd(xin).
kfun Kernel function to use.

Details

Implements the classical density kernel estimator based on a sample $X_1, X_2, ..., X_n$ of i.i.d observations from a distribution $F$ with density $h$. The estimator is defined by

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

where $h$ is determined by a bandwidth selector such as Silverman’s default plug-in rule and $K$, the kernel, is a non-negative probability density function.

Value

A vector with the density estimates at the designated points xout.
**p.sample**

Switch between a range of available cumulative distribution functions.

**Description**

Returns the value of the selected cumulative distribution function at user supplied grid points.

**Usage**

```r
p.sample(s, dist, p1, p2)
```

**Arguments**

- `s`: A scalar or vector: the x-axis grid points where the cumulative distribution function is be evaluated.
- `dist`: Character string, used as a switch to the user selected distribution function (see details below).
- `p1`: A scalar. Parameter 1 (vector or object) of the selected distribution.
- `p2`: A scalar. Parameter 2 (vector or object) of the selected distribution.
Details

Based on the user-specified argument \texttt{dist}, the function returns the value of the cumulative distribution function at \( s \).

Supported distributions (along with the corresponding \texttt{dist} values) are:

- \texttt{weib}: The Weibull distribution is implemented as
  \[ F(s) = 1 - \exp \left\{ - \left( \frac{s}{p_2} \right)^{p_1} \right\} \]
  with \( s > 0 \) where \( p_1 \) is the shape parameter and \( p_2 \) the scale parameter.
- \texttt{lognorm}: The lognormal distribution is implemented as
  \[ F(s) = \Phi \left( \frac{\ln s - p_1}{p_2} \right) \]
  where \( p_1 \) is the mean, \( p_2 \) is the standard deviation and \( \Phi \) is the cumulative distribution function of the standard normal distribution.
- \texttt{norm}: The normal distribution is implemented as
  \[
  \Phi(s) = \frac{1}{\sqrt{2\pi}p_2} \int_{-\infty}^{s} e^{-\frac{(t-p_1)^2}{2p_2^2}} dt
  \]
  where \( p_1 \) is the mean and the \( p_2 \) is the standard deviation.
- \texttt{uni}: The uniform distribution is implemented as
  \[ F(s) = \frac{s - p_1}{p_2 - p_1} \]
  for \( p_1 \leq s \leq p_2 \).
- \texttt{cauchy}: The cauchy distribution is implemented as
  \[ F(s; p_1, p_2) = \frac{1}{\pi} \arctan \left( \frac{s - p_1}{p_2} \right) + \frac{1}{2} \]
  where \( p_1 \) is the location parameter and \( p_2 \) the scale parameter.
- \texttt{fnorm}: The half normal distribution is implemented as
  \[
  F_S(s; \sigma) = \int_{0}^{s} \frac{\sqrt{2/\pi}}{\sigma} \exp \left\{ -\frac{x^2}{2\sigma^2} \right\} dx
  \]
  where \( mean = 0 \) and \( sd = \sqrt{\pi/2}/p_1 \).
- \texttt{normmixt}: The normal mixture distribution is implemented as
  \[
  F(s) = p_1 \frac{1}{p_2[2] \sqrt{2\pi}} \int_{-\infty}^{s} e^{-\frac{(t-p_2[1])^2}{2p_2[2]^2}} dt + (1 - p_1) \frac{1}{p_2[4] \sqrt{2\pi}} \int_{-\infty}^{s} e^{-\frac{(t-p_2[3])^2}{2p_2[4]^2}} dt
  \]
  where \( p_1 \) is a mixture component(scalar) and \( p_2 \) a vector of parameters for the mean and variance of the two mixture components \( p_2 = c(mean1, sd1, mean2, sd2) \).
• skernorm: The skew normal distribution is implemented as

\[ F(y; p_1) = \Phi\left(\frac{y - \xi}{\omega}\right) - 2T\left(\frac{y - \xi}{\omega}, p_1\right) \]

where location = \(\xi = 0\), scale = \(\omega = 1\), parameter = \(p_1\) and \(T(h, a)\) is the Owens T function, defined by

\[ T(h, a) = \frac{1}{2\pi} \int_{-\infty}^{a} \exp\left\{\frac{-0.5h^2(1 + x^2)}{1 + x^2}\right\} dx, -\infty \leq h, a \leq \infty \]

• fas: The Fernandez and Steel distribution is implemented as

\[ F(s; p_1, p_2) = \frac{2}{p_1 + 1}\left\{\int_{-\infty}^{s} f_t(x/p_1; p_2)I_{x \geq 0} dx + \int_{-\infty}^{s} f_t(p_1 x; p_2)I_{x < 0} dx\right\} \]

where \(f_t(x; \nu)\) is the p.d.f. of the t distribution with \(\nu = 5\) degrees of freedom, \(p_1\) controls the skewness of the distribution with values between \((0, +\infty)\) and \(p_2\) is the degrees of freedom.

• shash: The Sinh-Arcsinh distribution is implemented as

\[ F(s; \mu, p_2, p_1, \tau) = \int_{-\infty}^{s} \frac{ce^{-r^2/2}}{\sqrt{2\pi}} \frac{1}{p_2^2} \frac{1}{\sqrt{1 + z^2}} dz \]

where \(r = \sinh(\sinh(z) - p_1), c = \cosh(\sinh(z) - p_1)\) and \(z = (s - \mu)/p_2\). \(p_1\) is the vector of skewness, \(p_2\) is the scale parameter, \(\mu = 0\) is the location parameter and \(\tau = 1\) the kurtosis parameter.

Value

A vector containing the cumulative distribution function values at the user specified points \(s\).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

r.sample, q.sample, d.sample
Examples

```r
selected.d <- "weib"  # select Weibull as the CDF
shape <- 2  # specify shape parameter
scale <- 1  # specify scale parameter
xout <- seq(0.1, 5, length=50)  # design point where the CDF is evaluated
p.sample(xout, selected.d, shape, scale)  # calculate CDF at xout
```

---

**pdfsq**

*Calculate $f^2(x)$*

---

**Description**

Calculates the square of a density.

**Usage**

```r
pdfsq(s, dist, p1, p2)
```

**Arguments**

- `s`: A scalar or vector: the x-axis grid points where the probability density function will be evaluated.
- `dist`: Character string, used as a switch to the user selected distribution function (see details below).
- `p1`: A scalar. Parameter 1 (vector or object) of the selected density.
- `p2`: A scalar. Parameter 2 (vector or object) of the selected density.

**Details**

Based on user-specified argument `dist`, the function returns the value of $f^2(x)dx$, used in the definitions of $\rho^*_p$, $\rho_p$ and their exact versions.

Supported distributions (along with the corresponding `dist` values) are:

- **weib**: The weibull distribution is implemented as

  $$f(s; p_1, p_2) = \frac{p_1}{p_2} \left( \frac{s}{p_2} \right)^{p_1-1} \exp \left\{ - \left( \frac{s}{p_2} \right)^{p_1} \right\}$$

  with $s \geq 0$ where $p_1$ is the shape parameter and $p_2$ the scale parameter.

- **lognorm**: The lognormal distribution is implemented as

  $$f(s) = \frac{1}{p_2 s \sqrt{2\pi}} e^{-\frac{(\log s - p_1)^2}{2p_2^2}}$$

  where $p_1$ is the mean and $p_2$ is the standard deviation of the distribution.
• norm: The normal distribution is implemented as
\[ f(s) = \frac{1}{p_2\sqrt{2\pi}} e^{-\frac{(s-p_1)^2}{2p_2^2}} \]
where \( p_1 \) is the mean and the \( p_2 \) is the standard deviation of the distribution.

• uni: The uniform distribution is implemented as
\[ f(s) = \frac{1}{p_2 - p_1} \]
for \( p_1 \leq s \leq p_2 \).

• cauchy: The cauchy distribution is implemented as
\[ f(s) = \frac{1}{\pi p_2} \left\{ 1 + \frac{(s-p_1)^2}{p_2^2} \right\} \]
where \( p_1 \) is the location parameter and \( p_2 \) the scale parameter.

• fnorm: The half normal distribution is implemented as
\[ 2f(s) - 1 \]
where
\[ f(s) = \frac{1}{sd\sqrt{2\pi}} e^{-\frac{s^2}{2sd^2}} \]
and \( sd = \sqrt{\pi/2}/p_1 \).

• normmixt: The normal mixture distribution is implemented as
\[ f(s) = p_1 \frac{1}{p_2[2]\sqrt{2\pi}} e^{-\frac{(s-p_2[1])^2}{2p_2[2]^2}} + (1 - p_1) \frac{1}{p_2[4]\sqrt{2\pi}} e^{-\frac{(s-p_2[3])^2}{2p_2[4]^2}} \]
where \( p_1 \) is a mixture component (scalar) and \( p_2 \) a vector of parameters for the mean and variance of the two mixture components \( p_2 = (mean1, sd1, mean2, sd2) \).

• skewnorm: The skew normal distribution with parameter \( p_1 \) is implemented as
\[ f(s) = 2\phi(s)\Phi(p_1 s) \]

• fas: The Fernandez and Steel distribution is implemented as
\[ f(s; p_1, p_2) = \frac{2}{p_1 + \frac{1}{p_1}} \left\{ f_t(s/p_1; p_2) I_{s \geq 0} + f_t(p_1 s; p_2) I_{s < 0} \right\} \]
where \( f_t(x; \nu) \) is the p.d.f. of the t distribution with \( \nu = 5 \) degrees of freedom. \( p_1 \) controls the skewness of the distribution with values between \((0, +\infty)\) and \( p_2 \) denotes the degrees of freedom.

• shash: The Sinh-Arcsinh distribution is implemented as
\[ f(s; \mu, p_1, p_2, \tau) = \frac{ce^{-r^2/2}}{\sqrt{2\pi} p_2} \frac{1}{2} \sqrt{1 + z^2} \]
where \( r = \sinh(\sinh(z) - (-p_1)), c = \cosh(\sinh(z) - (-p_1)) \) and \( z = ((s - \mu)/p2). p_1 \) is the vector of skewness, \( p_2 \) is the scale parameter, \( \mu = 0 \) is the location parameter and \( \tau = 1 \) the kurtosis parameter.
Value
A vector containing the user selected density values at the user specified points \( s \).

Author(s)
Dimitrios Bagkavos and Lucia Gamez Gallardo
R implementation and documentation: Dimitrios Bagkavos \(<\text{dimitrios.bagkavos@gmail.com}>\), Lucia Gamez Gallardo \(<\text{gamezgallardolucia@gmail.com}>\)

References

See Also
\( r\text{-sample}, q\text{-sample}, p\text{-sample} \)

Examples
```r
selected.dens <- "weib" # select Weibull
shape <- 2 # specify shape parameter
scale <- 1 # specify scale parameter
xout <- seq(0.1,5,length=50) # design point
pdfsq(xout,selected.dens,shape,scale) # calculate the square density at xout
```

Description
Calculate \( f^2(x)F(x) \)

Usage
\[ \text{pdfsqcdf}(s, \text{dist}, p1, p2) \]

Arguments

\( s \)  
A scalar or vector: the x-axis grid points where the probability density function will be evaluated.

\( \text{dist} \)  
Character string, used as a switch to the user selected distribution function (see details below).

\( p1 \)  
A scalar. Parameter 1 (vector or object) of the selected density.

\( p2 \)  
A scalar. Parameter 2 (vector or object) of the selected density.
Details

Based on user-specified argument dist, the function returns the value of \( f^2(x)F(x) \)dx, used in the definitions of \( \rho_p^*, \rho_p \) and their exact versions.

Supported distributions (along with the corresponding dist values) are:

- **weib**: The Weibull distribution is implemented as

  \[
  f(s; p_1, p_2) = \frac{p_1}{p_2} \left( \frac{s}{p_2} \right)^{p_1-1} \exp \left\{ - \left( \frac{s}{p_2} \right)^{p_1} \right\}
  \]

  with \( s \geq 0 \) where \( p_1 \) is the shape parameter and \( p_2 \) the scale parameter.

- **lognorm**: The lognormal distribution is implemented as

  \[
  f(s) = \frac{1}{p_2 s \sqrt{2\pi}} e^{-\frac{(\log s - p_1)^2}{2p_2^2}}
  \]

  where \( p_1 \) is the mean and \( p_2 \) is the standard deviation of the distribution.

- **norm**: The normal distribution is implemented as

  \[
  f(s) = \frac{1}{p_2 \sqrt{2\pi}} e^{-\frac{(s - p_1)^2}{2p_2^2}}
  \]

  where \( p_1 \) is the mean and \( p_2 \) is the standard deviation of the distribution.

- **uni**: The uniform distribution is implemented as

  \[
  f(s) = \frac{1}{p_2 - p_1}
  \]

  for \( p_1 \leq s \leq p_2 \).

- **cauchy**: The Cauchy distribution is implemented as

  \[
  f(s) = \frac{1}{\pi p_2 \left\{ 1 + \left( \frac{s - p_1}{p_2} \right)^2 \right\}}
  \]

  where \( p_1 \) is the location parameter and \( p_2 \) the scale parameter.

- **fnorm**: The half normal distribution is implemented as

  \[
  2f(s) - 1
  \]

  where

  \[
  f(s) = \frac{1}{sd \sqrt{2\pi}} e^{-\frac{s^2}{2sd^2}},
  \]

  and \( sd = \sqrt{\pi/2}/p_1 \).

- **normmixt**: The normal mixture distribution is implemented as

  \[
  f(s) = p_1 \frac{1}{p_2[2] \sqrt{2\pi}} e^{-\frac{(s - p_2[1])^2}{2p_2[2]}} + (1 - p_1) \frac{1}{p_2[4] \sqrt{2\pi}} e^{-\frac{(s - p_2[3])^2}{2p_2[4]}}
  \]

  where \( p_1 \) is a mixture component (scalar) and \( p_2 \) a vector of parameters for the mean and variance of the two mixture components \( p_2 = c(mean1, sd1, mean2, sd2) \).
• skernorm: The skew normal distribution with parameter $p_1$ is implemented as

$$f(s) = 2\phi(s)\Phi(p_1 s)$$

• fas: The Fernandez and Steel distribution is implemented as

$$f(s; p_1, p_2) = \frac{2}{p_1 + \frac{1}{p_1}} \left\{ f_t(s/p_1; p_2)I_{s \geq 0} + f_t(p_1 s; p_2)I_{s < 0} \right\}$$

where $f_t(x; \nu)$ is the p.d.f. of the $t$ distribution with $\nu = 5$ degrees of freedom. $p_1$ controls the skewness of the distribution with values between $(0, +\infty)$ and $p_2$ denotes the degrees of freedom.

• shash: The Sinh-Arcsinh distribution is implemented as

$$f(s; \mu, p_1, p_2, \tau) = ce^{-r^2/2} \sqrt{2\pi} p_2 1/2 \sqrt{1 + z^2}$$

where $r = \sinh(\sinh(z) - (-p_1))$, $c = \cosh(\sinh(z) - (-p_1))$ and $z = ((s - \mu)/p_2)$. $p_1$ is the vector of skewness, $p_2$ is the scale parameter, $\mu = 0$ is the location parameter and $\tau = 1$ the kurtosis parameter.

Value

A vector containing the user selected density values at the user specified points $s$.

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

r.sample, q.sample, p.sample

Examples

```r
selected.dens <- "weib" # select Weibull
shape <- 2 # specify shape parameter
scale <- 1 # specify scale parameter
xout <- seq(0.1, 5, length=50) # design point
pdfsqcdf(xout, selected.dens, shape, scale) # calculate pdfsqcdf function at xout
```
pdfsqcdfstar  Calculate \( f^2(x)(1 - F(x)) \).

Description

Return the product \( f^2(x)(1 - F(x)) \).

Usage

\texttt{pdfsqcdfstar(s,dist, p1,p2)}

Arguments

\begin{itemize}
  \item \texttt{s} \quad A scalar or vector: the x-axis grid points where the probability density function will be evaluated.
  \item \texttt{dist} \quad Character string, used as a switch to the user selected distribution function (see details below).
  \item \texttt{p1} \quad A scalar. Parameter 1 (vector or object) of the selected density.
  \item \texttt{p2} \quad A scalar. Parameter 2 (vector or object) of the selected density.
\end{itemize}

Details

Based on user-specified argument \texttt{dist}, the function returns the value of \( f^2(x)(1 - F(x))dx \), used in the definitions of \( \rho^*_p, \rho_p \) and their exact versions.

Supported distributions (along with the corresponding \texttt{dist} values) are:

- \texttt{weib}: The weibull distribution is implemented as
  \[
  f(s; p_1, p_2) = \left( \frac{p_1}{p_2} \right)^{p_1-1} s^{p_1-1} \exp \left\{ - \left( \frac{s}{p_2} \right)^{p_1} \right\}
  \]
  with \( s \geq 0 \) where \( p_1 \) is the shape parameter and \( p_2 \) the scale parameter.

- \texttt{lognorm}: The lognormal distribution is implemented as
  \[
  f(s) = \frac{1}{p_2 s \sqrt{2\pi}} e^{-\frac{(\log s - p_1)^2}{2p_2^2}}
  \]
  where \( p_1 \) is the mean and \( p_2 \) is the standard deviation of the distribution.

- \texttt{norm}: The normal distribution is implemented as
  \[
  f(s) = \frac{1}{p_2 \sqrt{2\pi}} e^{-\frac{(s - p_1)^2}{2p_2^2}}
  \]
  where \( p_1 \) is the mean and the \( p_2 \) is the standard deviation of the distribution.
• uni: The uniform distribution is implemented as

\[ f(s) = \frac{1}{p_2 - p_1} \]

for \( p_1 \leq s \leq p_2 \).

• cauchy: The cauchy distribution is implemented as

\[ f(s) = \frac{1}{\pi p_2} \left\{ 1 + \left( \frac{s-p_1}{p_2} \right)^2 \right\} \]

where \( p_1 \) is the location parameter and \( p_2 \) the scale parameter.

• fnorm: The half normal distribution is implemented as

\[ 2f(s) - 1 \]

where

\[ f(s) = \frac{1}{sd\sqrt{2\pi}} e^{-\frac{s^2}{2sd^2}} \]

and \( sd = \sqrt{\pi/2}/p_1 \).

• normmixt: The normal mixture distribution is implemented as

\[ f(s) = p_1 \frac{1}{p_2[2]\sqrt{2\pi}} e^{-\frac{(s-p_1[1])^2}{2p_2[2]^2}} + \left( 1 - p_1 \right) \frac{1}{p_2[4]\sqrt{2\pi}} e^{-\frac{(s-p_2[3])^2}{2p_2[4]^2}} \]

where \( p_1 \) is a mixture component (scalar) and \( p_2 \) a vector of parameters for the mean and variance of the two mixture components \( p_2 = c(mean1, sd1, mean2, sd2) \).

• skewnorm: The skew normal distribution with parameter \( p_1 \) is implemented as

\[ f(s) = 2\phi(s)\Phi(p_1s) \]

• fas: The Fernandez and Steel distribution is implemented as

\[ f(s; p_1, p_2) = \frac{2}{p_1 + \frac{1}{p_1}} \left\{ f_t(s/p_1; p_2)I_{s \geq 0} + f_t(p_1s; p_2)I_{s < 0} \right\} \]

where \( f_t(x; \nu) \) is the p.d.f. of the \( t \) distribution with \( \nu = 5 \) degrees of freedom. \( p_1 \) controls the skewness of the distribution with values between \((0, +\infty)\) and \( p_2 \) denotes the degrees of freedom.

• shash: The Sinh-Arcsinh distribution is implemented as

\[ f(s; \mu, p_1, p_2, \tau) = \frac{ce^{-r^2/2}}{\sqrt{2\pi}} \frac{1}{p_2} \frac{1}{2} \sqrt{1 + z^2} \]

where \( r = \sinh(\sinh(z) - (-p_1)) \), \( c = \cosh(\sinh(z) - (-p_1)) \) and \( z = ((s - \mu)/p_2) \). \( p_1 \) is the vector of skewness, \( p_2 \) is the scale parameter, \( \mu = 0 \) is the location parameter and \( \tau = 1 \) the kurtosis parameter.
Value
A vector containing the user selected density values at the user specified points s.

Author(s)
Dimitrios Bagkavos and Lucia Gamez Gallardo
R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References

See Also
r.sample, q.sample, p.sample

Examples
```r
calculated.dens <- "weib" # select Weibull
density <- 2 # specify shape parameter
scale <- 1 # specify scale parameter
xout <- seq(0.1, 5, length=50) # design point
pdfsqcdfstar(xout, calculated.dens, density, scale) # return \( f^2(x)F(x) \)
cdf <- pdfthird(xout, calculated.dens, density, scale) # return \( f^3(x) \)
```

Description
Return the value of \( f^3(x) \).

Usage
```r
pdfthird(s, dist, p1, p2)
```

Arguments
- **s**: A scalar or vector: the x-axis grid points where the probability density function will be evaluated.
- **dist**: Character string, used as a switch to the user selected distribution function (see details below).
- **p1**: A scalar. Parameter 1 (vector or object) of the selected density.
- **p2**: A scalar. Parameter 2 (vector or object) of the selected density.
Details

Based on user-specified argument dist, the function returns the value of \( f^3(x)dx \), used in the definitions of \( \rho^*_p, \rho_p \) and their exact versions.

Supported distributions (along with the corresponding dist values) are:

- **weib**: The weibull distribution is implemented as
  \[
  f(s; p_1, p_2) = \frac{p_1}{p_2} \left( \frac{s}{p_2} \right)^{p_1 - 1} \exp \left\{ - \left( \frac{s}{p_2} \right)^{p_1} \right\}
  \]
  with \( s \geq 0 \) where \( p_1 \) is the shape parameter and \( p_2 \) the scale parameter.

- **lognorm**: The lognormal distribution is implemented as
  \[
  f(s) = \frac{1}{p_2 \sqrt{2\pi}} e^{-\frac{(\log s - p_1)^2}{2p_2^2}}
  \]
  where \( p_1 \) is the mean and \( p_2 \) is the standard deviation of the distribution.

- **norm**: The normal distribution is implemented as
  \[
  f(s) = \frac{1}{p_2 \sqrt{2\pi}} e^{-\frac{(s - p_1)^2}{2p_2^2}}
  \]
  where \( p_1 \) is the mean and \( p_2 \) is the standard deviation of the distribution.

- **uni**: The uniform distribution is implemented as
  \[f(s) = \frac{1}{p_2 - p_1}\]
  for \( p_1 \leq s \leq p_2 \).

- **cauchy**: The cauchy distribution is implemented as
  \[
  f(s) = \frac{1}{\pi p_2} \left\{ 1 + \left( \frac{s - p_1}{p_2} \right)^2 \right\}
  \]
  where \( p_1 \) is the location parameter and \( p_2 \) the scale parameter.

- **fnorm**: The half normal distribution is implemented as
  \[
  2f(s) - 1
  \]
  where
  \[
  f(s) = \frac{1}{sd \sqrt{2\pi}} e^{-\frac{s^2}{2sd^2}},
  \]
  and \( sd = \sqrt{\pi/2}/p_1 \).

- **normmixt**: The normal mixture distribution is implemented as
  \[
  f(s) = p_1 \frac{1}{p_2[2]\sqrt{2\pi}} e^{-\frac{(s - p_2[1])^2}{2p_2[2]^2}} + (1 - p_1) \frac{1}{p_2[4]\sqrt{2\pi}} e^{-\frac{(s - p_2[3])^2}{2p_2[4]^2}}
  \]
  where \( p_1 \) is a mixture component(scalar) and \( p_2 \) a vector of parameters for the mean and variance of the two mixture components \( p_2 = (mean1, sd1, mean2, sd2) \).
• skewnorm: The skew normal distribution with parameter \( p_1 \) is implemented as

\[
f(s) = 2\phi(s)\Phi(p_1s)
\]

• fas: The Fernandez and Steel distribution is implemented as

\[
f(s; p_1, p_2) = \frac{2}{p_1 + \frac{1}{p_1}} \left\{ f_t(s/p_1; p_2)I_{s\geq0} + f_t(p_1s; p_2)I_{s<0} \right\}
\]

where \( f_t(x; \nu) \) is the p.d.f. of the \( t \) distribution with \( \nu = 5 \) degrees of freedom. \( p_1 \) controls the skewness of the distribution with values between \((0, +\infty)\) and \( p_2 \) denotes the degrees of freedom.

• shash: The Sinh-Arcsinh distribution is implemented as

\[
f(s; \mu, p_1, p_2, \tau) = c e^{-r^2/2} \sqrt{\frac{1}{2\pi}} \frac{1}{p_2} \frac{1}{\sqrt{1+z^2}}
\]

where \( r = \sinh(\sinh(z) - (\mu)) \), \( c = \cosh(\sinh(z) - (\mu)) \) and \( z = ((s - \mu)/p_2) \). \( p_1 \) is the vector of skewness, \( p_2 \) is the scale parameter, \( \mu = 0 \) is the location parameter and \( \tau = 1 \) the kurtosis parameter.

Value

A vector containing the user selected density values at the user specified points \( s \).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

pdfsq, pdfsqcdf, pdfsqcdfstan

Examples

```r
selected.dens <- "weib" #select Weibull
shape <- 2 # specify shape parameter
scale <- 1 # specify scale parameter
xout <- seq(0.1,5,length=50) # design point
pdfthird(xout,selected.dens,shape,scale) # calculate density to the cube at xout
```
q.sample

\textit{Switch between a range of available quantile functions.}

\textbf{Description}

Returns the quantiles of selected distributions at user specified locations.

\textbf{Usage}

\texttt{q.sample(s, dist, p1=0, p2=1)}

\textbf{Arguments}

- \texttt{s}: A scalar or vector: the probabilities where the quantile function will be evaluated.
- \texttt{dist}: Character string, used as a switch to the user selected distribution function (see details below).
- \texttt{p1}: A scalar. Parameter 1 (vector or object) of the selected distribution.
- \texttt{p2}: A scalar. Parameter 2 (vector or object) of the selected distribution.

\textbf{Details}

Based on user-specified argument \texttt{dist}, the function returns the value of the quantile function at \texttt{s}.

Supported distributions (along with the corresponding \texttt{dist} values) are:

- \texttt{weib}: The quantile function for the weibull distribution is implemented as
  \[ Q(s) = p_1 (\log(1-s))^{1/p_2} \]
  where \( p_1 \) is the shape parameter and \( p_2 \) the scale parameter.
- \texttt{lognorm}: The lognormal distribution has quantile function implemented as
  \[ Q(s) = \exp\left\{ p_1 + \sqrt{2p_2^2} \text{erf}^{-1}(2s-1) \right\} \]
  where \( p_1 \) is the mean, \( p_2 \) is the standard deviation and \( \text{erf} \) is the Gauss error function.
- \texttt{norm}: The normal distribution has quantile function implemented as
  \[ Q(p) = \Phi^{-1}(s; p_1, p_2) \]
  where \( p_1 \) is the mean and the \( p_2 \) is the standard deviation.
- \texttt{uni}: The uniform distribution has quantile function implemented as
  \[ Q(s; p_1, p_2) = s(p_2 - p_1) + p_1 \]
  for \( p_1 < s < p_2 \).
q.sample

- cauchy: The cauchy distribution has quantile function implemented as
  \[ Q(s) = p_1 + p_2 \tan \left\{ \pi \left( s - \frac{1}{2} \right) \right\} \]
  where \( p_1 \) is the location parameter and \( p_2 \) the scale parameter.
- fnorm: The half normal distribution has quantile function implemented as
  \[ Q(s) = p_1 \sqrt{2} \text{erf}^{-1}(s) \]
  where and \( p_1 \) is the standard deviation of the distribution.
- normmix: The quantile function normal mixture distribution is estimated numerically, based on the built in quantile function.
- skewnorm: There is no closed form expression for the quantile function of the skew normal distribution. For this reason, the quantiles are calculated through the qsn function of the sn package.
- fas: There is no closed form expression for the quantile function of the Fernandez and Steel distribution. For this reason, the quantiles are calculated through the qskt function of the skewt package.
- shash: There is no closed form expression for the quantile function of the Sinh-Arcsinh distribution. For this reason, the quantiles are calculated through the qSHASHO function of the gamlss package.

Value

A vector containing the quantile values at the user specified points \( s \).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com> , Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

r.sample, d.sample, p.sample

Examples

```r
selected.q <- "norm" # select Normal as the distribution
shape <- 2 # specify shape parameter
scale <- 2 # specify scale parameter
xout <- seq(0.1,1,length=50) # design point where the quantile function is evaluated
q.sample(xout, selected.q, shape, scale) # calculate quantiles at xout
```
Description
Generate a random sample of size \( n \) out of a range of available distributions.

Usage
\[
\text{r.sample}(s, \text{dist}, p1=0, p2=1)
\]

Arguments
- \( s \): A scalar which specifies the size of the random sample drawn.
- \( \text{dist} \): Character string, used as a switch to the user selected distribution function (see details below).
- \( p1 \): A scalar. Parameter 1 (vector or object) of the selected distribution.
- \( p2 \): A scalar. Parameter 2 (vector or object) of the selected distribution.

Details
Based on user-specified argument \( \text{dist} \), the function returns a random sample of size \( s \) from the corresponding distribution.

Supported distributions (along with the corresponding \( \text{dist} \) values) are:

- **weib**: The Weibull distribution is implemented as
  \[
  f(s; p_1, p_2) = \frac{p_1}{p_2} \left( \frac{s}{p_2} \right)^{p_1 - 1} \exp \left\{ - \left( \frac{s}{p_2} \right)^{p_1} \right\}
  \]
  with \( s \geq 0 \) where \( p_1 \) is the shape parameter and \( p_2 \) the scale parameter.

- **lognorm**: The lognormal distribution is implemented as
  \[
  f(s; p_1, p_2) = \frac{1}{p_2 \sqrt{2\pi}} e^{- \left( \log s - p_1 \right)^2 / 2p_2^2}
  \]
  where \( p_1 \) is the mean and \( p_2 \) is the standard deviation of the distribution.

- **norm**: The normal distribution is implemented as
  \[
  f(s; p_1, p_2) = \frac{1}{p_2 \sqrt{2\pi}} e^{- \left( s - p_1 \right)^2 / 2p_2^2}
  \]
  where \( p_1 \) is the mean and the \( p_2 \) is the standard deviation of the distribution.

- **uni**: The uniform distribution is implemented as
  \[
  f(s; p_1, p_2) = \frac{1}{p_2 - p_1}
  \]
  for \( p_1 \leq s \leq p_2 \).
• cauchy: The cauchy distribution is implemented as
\[
f(s) = \frac{1}{\pi p_2 \left\{1 + \left(\frac{s - p_1}{p_2}\right)^2\right\}}
\]
where \(p_1\) is the location parameter and \(p_2\) the scale parameter.

• fnorm: The half normal distribution is implemented as
\[
2f(s) - 1
\]
where
\[
f(s) = \frac{1}{sd\sqrt{2\pi}}e^{-\frac{s^2}{2sd^2}},
\]
and \(sd = \sqrt{\pi/2}/p_1\).

• normmixt: The normal mixture distribution is implemented as
\[
f(s) = p_1 \frac{1}{p_2[2]\sqrt{2\pi}}e^{-\frac{(s-p_2[1])^2}{2p_2[2]^2}} + (1 - p_1) \frac{1}{p_2[4]\sqrt{2\pi}}e^{-\frac{(s-p_2[3])^2}{2p_2[4]^2}}
\]
where \(p_1\) is a mixture component (scalar) and \(p_2\) a vector of parameters for the mean and variance of the two mixture components \(p_2 = c(mean1, sd1, mean2, sd2)\).

• skewnorm: The skew normal distribution with parameter \(p_1\) is implemented as
\[
f(s) = 2\phi(s)\Phi(p_1s)
\]

• fas: The Fernandez and Steel distribution is implemented as
\[
f(s; p_1, p_2) = \frac{2}{p_1 + \frac{1}{p_1}} \left\{f_t(s/p_1; p_2)I(s \geq 0) + f_t(p_1s; p_2)I(s < 0)\right\}
\]
where \(f_t(x; \nu)\) is the p.d.f. of the \(t\) distribution with \(\nu = 5\) degrees of freedom. \(p_1\) controls the skewness of the distribution with values between \((0, +\infty)\) and \(p_2\) denotes the degrees of freedom.

• shash: The Sinh-Arcsinh distribution is implemented as
\[
f(s; \mu, p_1, p_2, \tau) = \frac{ce^{-r^2/2}}{\sqrt{2\pi} p_2} \frac{1}{2} \frac{1}{\sqrt{1 + z^2}}
\]
where \(r = \sinh(\sinh(z) - (-p_1)), c = \cosh(\sinh(z) - (-p_1))\) and \(z = ((s - \mu)/p_2). p_1\) is the vector of skewness, \(p_2\) is the scale parameter, \(\mu = 0\) is the location parameter and \(\tau = 1\) the kurtosis parameter.

**Value**

A vector of random values at the user specified points \(s\).
Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also
d.sample,q.sample,p.sample

Examples

```r
selected.r <- "norm" #select Normal as the distribution
shape <- 2  # specify shape parameter
scale <- 1  # specify scale parameter
n <- 100    # specify sample size
r.sample(n,selected.r,shape,scale)  # calculate CDF at the designated point
```

---

**Rho.p**

Calculates \( \rho_p \), used in the implementation of the strong asymmetry measure \( \eta(X) \).

Description

Estimates \( \rho_p \), used in the calculation of the strong asymmetry measure \( \eta(X) \).

Usage

```r
Rho.p(xin, p.param, dist, p1=0, p2=1)
```

Arguments

- **xin**: A vector of data points - the available sample.
- **p.param**: A parameter with the value greater than or equal to 1/2 and less than 1.
- **dist**: Character string, specifies selected distribution function.
- **p1**: A scalar. Parameter 1 (vector or object) of the selected distribution.
- **p2**: A scalar. Parameter 2 (vector or object) of the selected distribution.
Rho.p

Details

Implements the quantity:

\[
\frac{2\sqrt{3} - \int_{-\infty}^{\xi_p} f^2(x)F(x) \, dx - \frac{p}{2} \int_{-\infty}^{\xi_p} f^2(x) \, dx}{p \left( p \int_{-\infty}^{\xi_p} f^3(x) \, dx - (\int_{-\infty}^{\xi_p} f^2(x) \, dx)^2 \right)^{1/2}}
\]

defined on page 6 Patil, Bagkavos and Wood, see also (4) in Bagkavos, Patil and Wood. Estimation of the p.d.f. and c.d.f. functions is currently performed by maximum likelihood as e.g. kernel estimates inherit large amount of variance to \( \rho_p \).

Value

Returns a scalar, the value of \( \rho_p \).

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

Rho.p.exact, Rhostar.p, Rhostar.p.exact

Examples

```r
set.seed(1234)

selected.r <- "weib" # select Weibull as the distribution
shape <- 1 # specify shape parameter
scale <- 1 # specify scale parameter
n <- 100 # specify sample size
param <- 0.9 # specify parameter
xout<-r.sample(n,selected.r,shape,scale) # specify sample
Rho.p(xout,param,selected.r,shape,scale) # calculate Rho.p
#-0.06665222 # returns the result

selected.r2 <- "norm" # select Normal as the distribution
n <- 100 # specify sample size
mean <- 0 # specify the mean
```
sd <- 1 # specify the variance
param <- 0.9 # specify parameter
xout <- r.sample(n, selected.r2, mean, sd) # specify sample
Rho.p(xout, param, selected.r2, mean, sd) # calculate Rho.p
# -0.1005591 # returns the result

selected.r3 <- "cauchy" # select Cauchy as the distribution
n <- 100 # specify sample size
location <- 0 # specify the location parameter
scale <- 1 # specify the scale parameter
param <- 0.9 # specify parameter
xout <- r.sample(n, selected.r3, location, scale) # specify sample
Rho.p(xout, param, selected.r3, location, scale) # calculate Rho.p
# -0.0580943 # returns the result

Rho.p.exact

Calculates the exact value \( \rho_p \), used in the implementation of the strong asymmetry measure \( \eta(X) \).

Description

Returns \( \rho_p \), used in the calculation of the strong asymmetry measure \( \eta(X) \).

Usage

Rho.p.exact(xin, p.param, dist, p1=0, p2=1)

Arguments

- **xin**: A vector of data points - the available sample.
- **p.param**: A parameter with the value greater than or equal to 1/2 and less than 1.
- **dist**: Character string, specifies selected distribution function.
- **p1**: A scalar. Parameter 1 (vector or object) of the selected distribution.
- **p2**: A scalar. Parameter 2 (vector or object) of the selected distribution.

Details

Implements the quantity:

\[
\frac{2\sqrt{3}}{p} \left\{ p \int_{-\infty}^{\xi_p} f_3(x) \, dx - \left( \int_{-\infty}^{\xi_p} f_2(x) \, dx \right)^2 \right\}^{1/2}
\]

defined on page 6 Patil, Bagkavos and Wood, see also (4) in Bagkavos, Patil and Wood. This implementation uses exact calculation of the functionals in the definition of \( \rho_p \).
Value

Returns a scalar, the exact value of $\rho_p$.

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

Rho.p, Rhostar.p, Rhostar.p.exact

Examples

```r
set.seed(1234)

selected.r <- "weib" #select Weibull as the distribution
shape <- 1 # specify shape parameter
scale <- 1 # specify scale parameter
n <- 100 # specify sample size
param <- 0.9 # specify parameter
xout<-r.sample(n,selected.r,shape,scale) # specify sample
Rho.p.exact(xout,param,selected.r,shape,scale) # calculate Rho.p.exact
#-0.06665222 # returns the result

selected.r2 <- "norm" #select Normal as the distribution
n <- 100 # specify sample size
mean <- 0 # specify the mean
sd <- 1 # specify the variance
param <- 0.9 # specify parameter
xout <- r.sample(n,selected.r2,mean,sd) # specify sample
Rho.p.exact(xout,param,selected.r2,mean,sd) # calculate Rho.p.exact
#-0.2384271 # returns the result

selected.r3 <- "cauchy" #select Cauchy as the distribution
n <- 100 # specify sample size
location <- 0 # specify the location parameter
scale <- 1 # specify the scale parameter
param <- 0.9 # specify parameter
xout <- r.sample(n,selected.r3,location,scale) # specify sample
```
\texttt{Rho.p.exact(xout,param,selected.r3,location,scale)} # calculate Rho.p.exact
\#-0.82340374 # returns the result

---

\textbf{Rhostar.p} \hspace{1cm} \textit{Calculates $\rho^*_p$, used in the implementation of the strong asymmetry measure $\eta(X)$.}

---

\textbf{Description}
Estimates $\rho^*_p$, used in the calculation of the strong asymmetry measure $\eta(X)$.

\textbf{Usage}
\texttt{Rhostar.p(xin, p.param, dist, p1, p2)}

\textbf{Arguments}
- \texttt{xin} A vector of data points - the available sample.
- \texttt{p.param} A parameter with the value greater than or equal to 1/2 and less than 1.
- \texttt{dist} Character string, specifies selected distribution function.
- \texttt{p1} A scalar. Parameter 1 (vector or object) of the selected distribution.
- \texttt{p2} A scalar. Parameter 2 (vector or object) of the selected distribution.

\textbf{Details}
Implements the quantity
\[
\frac{2\sqrt{3}}{p} \left\{ p \int_{\xi_1 - p}^{\infty} f^3(x) \, dx - \left( \int_{\xi_1 - p}^{\infty} f^2(x) \, dx \right)^2 \right\}^{1/2}
\]
defined on page 6 Patil, Bagkavos and Wood (2014), see also (5) in Bagkavos, Patil and Wood (2016). Estimation of the p.d.f. and c.d.f. functions is currently performed by maximum likelihood as e.g. kernel estimates inherit large amount of variance to $\rho^*_p$.

\textbf{Value}
Returns a scalar, the value of $\rho^*_p$.

\textbf{Author(s)}
Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>
References


See Also


Examples

```r
set.seed(1234)

selected.r <- "weib" # select Weibull as the distribution
shape <- 1 # specify shape parameter
scale <- 1 # specify scale parameter
n <- 100 # specify sample size
param <- 0.9 # specify parameter
xout<-r.sample(n,selected.r,shape,scale) # specify sample
Rhostar.p(xout,param,selected.r,shape,scale) # calculate Rhostar.p
#-0.08936363 # returns the result

selected.r2 <- "norm" # select Normal as the distribution
n <- 100 # specify sample size
mean <- 0 # specify the mean
sd <- 1 # specify the variance
param <- 0.9 # specify parameter
xout <- r.sample(n,selected.r2,mean,sd) # specify sample
Rhostar.p(xout,param,selected.r2,mean,sd) # calculate Rhostar.p
#-0.02302223 # returns the result

selected.r3 <- "cauchy" # select Cauchy as the distribution
n <- 100 # specify sample size
location <- 0 # specify the location parameter
scale <- 1 # specify the scale parameter
param <- 0.9 # specify parameter
xout<-r.sample(n,selected.r3,location,scale) # specify sample
Rhostar.p(xout,param,selected.r3,location,scale) # calculate Rhostar.p
#0.02043852 # returns the result
```

Rhostar.p.exact

Calculates the exact value of $\rho \_p^\ast$, used in the implementation of the strong asymmetry measure $\eta(X)$. 
Description

Returns $\rho^*_p$, used in the calculation of the strong asymmetry measure $\eta(X)$.

Usage

Rhostar.p.exact(xin, p.param, dist, p1, p2)

Arguments

- **xin**: A vector of data points - the available sample.
- **p.param**: A parameter with the value greater than or equal to 1/2 and less than 1.
- **dist**: Character string, specifies selected distribution function.
- **p1**: A scalar. Parameter 1 (vector or object) of the selected distribution.
- **p2**: A scalar. Parameter 2 (vector or object) of the selected distribution.

Details

Implements the quantity

$$\frac{2\sqrt{3}}{p} \left\{ p \int_{\xi_{1-p}}^{\infty} f^3(x) \, dx - \left( \int_{\xi_{1-p}}^{\infty} f^2(x) \, dx \right)^2 \right\}^{1/2}$$

defined on page 6 Patil, Bagkavos and Wood (2014), see also (5) in Bagkavos, Patil and Wood (2016). This implementation uses exact calculation of the functionals in the definition of $\rho^*_p$.

Value

Returns a scalar, the exact value of $\rho^*_p$.

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References


See Also

Rho.p, Rhostar.p, Rho.p.exact
Examples

```r
set.seed(1234)

selected.r <- "weib" #select Weibull as the distribution
shape <- 1 # specify shape parameter
scale <- 1 # specify scale parameter
n <- 100 # specify sample size
param <- 0.9 # specify parameter
xout<-r.sample(n,selected.r,shape,scale) # specify sample
Rhostar.p.exact(xout,param,selected.r,shape,scale) # calculate Rhostar.p.exact
#-0.05206678 # returns the result

selected.r2 <- "norm" #select Normal as the distribution
n <- 100 # specify sample size
mean <- 0 # specify the mean
sd <- 1 # specify the variance
param <- 0.9 # specify parameter
xout <- r.sample(n,selected.r2,mean,sd) # specify sample
Rhostar.p.exact(xout,param,selected.r2,mean,sd) # calculate Rhostar.p.exact
#-0.008687447 # returns the result

selected.r3 <- "cauchy" #select Cauchy as the distribution
n <- 100 # specify sample size
location <- 0 # specify the location parameter
scale <- 1 # specify the scale parameter
param <- 0.9 # specify parameter
xout <- r.sample(n,selected.r3,location,scale) # specify sample
Rhostar.p.exact(xout,param,selected.r3,location,scale) # calculate Rhostar.p.exact
#0.0280602 # returns the result
```

SimpsonInt

Simpson integration

Description

Implements Simpson's extended integration rule.

Usage

SimpsonInt(xin,h)

Arguments

- **xin**: A vector of design points where the integral will be evaluated.
- **h**: Assuming a<b and n is a positive integer. \( h = (b - a)/n \).
Details

Simpson’s extended numerical integration rule is implemented for \( n+1 \) equally spaced subdivisions (where \( n \) is even) of \([a, b]\) as

\[
\int_a^b f(x) \, dx = \frac{h}{3} \left\{ f(a) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \ldots + 4f(x_{n-1}) + f(b) \right\}
\]

where \( hx = (b - a)/n \) and \( x_i = a + ihx \). Simpson’s rule will return an exact result when the polynomial in question has a degree of three or less. For other functions, Simpson’s Rule only gives an approximation.

Value

A scalar, the approximate value of the integral.

Author(s)

Dimitrios Bagkavos and Lucia Gamez Gallardo

R implementation and documentation: Dimitrios Bagkavos <dimitrios.bagkavos@gmail.com>, Lucia Gamez Gallardo <gamezgallardolucia@gmail.com>

References

Simpson’s Rule

Examples

```r
x.in <- seq(0, pi/4, length=5)
h.out <- pi/8
SimpsonInt(x.in, h.out)
```
Index

bw.bcv, 20, 21
bw.nrd, 20, 21
bw.nrd0, 20, 21
bw.ucv, 20, 21

d.sample, 2, 23, 35, 38

edf, 5
Epanechnikov, 6, 18
eta.s, 7, 9
eta.s.exact, 8
eta.w.breve, 8, 9, 9, 12–16
eta.w.breve.bc, 8–10, 11, 13–16
eta.w.hat, 8–10, 12, 12, 14–16
eta.w.hat.bc, 8–10, 12, 13, 13, 15, 16
eta.w.tilde, 8–10, 12–14, 14, 16
eta.w.tilde.bc, 8–10, 12–15, 15

GDP.Per.head.dist.1995, 16, 18
GDP.Per.head.dist.2005, 17, 17

IntEpanechnikov, 6, 18
IntKde, 19

kde, 20

p.sample, 4, 21, 26, 28, 31, 35, 38
pdfsq, 24, 33
pdfsqcdf, 26, 33
pdfsqcdfstar, 29, 33
pdfthird, 31

q.sample, 4, 23, 26, 28, 31, 34, 38

r.sample, 4, 23, 26, 28, 31, 35, 36
Rho.p, 38, 41, 43, 44
Rho.p.exact, 39, 40, 43, 44
Rhostar.p, 39, 41, 42, 44
Rhostar.p.exact, 39, 41, 43, 43

SimpsonInt, 45