Package ‘HyperbolicDist’

February 19, 2015

Version 0.6-2

Date 2009-09-09

Title The hyperbolic distribution

Author David Scott <d.scott@auckland.ac.nz>

Maintainer David Scott <d.scott@auckland.ac.nz>

Depends R (>= 2.3.0)

Suggests VarianceGamma, actuar

Encoding latin1

Description This package provides functions for the hyperbolic and related distributions. Density, distribution and quantile functions and random number generation are provided for the hyperbolic distribution, the generalized hyperbolic distribution, the generalized inverse Gaussian distribution and the skew-Laplace distribution. Additional functionality is provided for the hyperbolic distribution, including fitting of the hyperbolic to data.

License GPL (>= 2)

URL http://www.r-project.org

Repository CRAN

Repository/R-Forge/Project rmetrics

Repository/R-Forge/Revision 4411

Date/Publication 2009-09-23 16:45:00

NeedsCompilation no

R topics documented:

`Bessel K Ratio` *Ratio of Bessel K Functions*

Description

Calculates the ratio of Bessel K functions of different orders

Usage

```
besselRatio(x, nu, orderDiff, useExpScaled = 700)
```

Arguments

`x`	Numeric, ≥ 0. Value at which the numerator and denominator Bessel functions are evaluated.
`nu`	Numeric. The order of the Bessel function in the denominator.
`orderDiff`	Numeric. The order of the numerator Bessel function minus the order of the denominator Bessel function.
`useExpScaled`	Numeric, ≥ 0. The smallest value of x for which the ratio is calculated using the exponentially-scaled Bessel function values.

Details

Uses the function `besselK` to calculate the ratio of two modified Bessel function of the third kind whose orders are different. The calculation of Bessel functions will underflow if the value of x is greater than around 740. To avoid underflow the exponentially-scaled Bessel functions can be returned by `besselK`. The ratio is actually unaffected by exponential scaling since the scaling cancels across numerator and denominator.

The Bessel function ratio is useful in calculating moments of the Generalized Inverse Gaussian distribution, and hence also for the moments of the hyperbolic and generalized hyperbolic distributions.

Value

The ratio

$$\frac{K_{\nu+k}(x)}{K_{\nu}(x)}$$

of two modified Bessel functions of the third kind whose orders differ by k.

Author(s)

David Scott <`d.scott@auckland.ac.nz`>

See Also

`besselK`, `gigMom`

Examples

```
nus <- c(0:5, 10, 20)
x <- seq(1, 4, length.out = 11)
k <- 3

raw <- matrix(nrow = length(nus), ncol = length(x))
scaled <- matrix(nrow = length(nus), ncol = length(x))
compare <- matrix(nrow = length(nus), ncol = length(x))

for (i in 1:length(nus)){
    for (j in 1:length(x)) {
        raw[i,j] <- besselRatio(x[j], nus[i],
                                orderDiff = k)
```

```
        scaled[i,j] <- besselRatio(x[j], nus[i],
                                   orderDiff = k, useExpScaled = 1)
        compare[i,j] <- raw[i,j]/scaled[i,j]
    }
}
raw
scaled
compare
```

Functions for Moments *Functions for Calculating Moments*

Description

Functions used to calculate the mean, variance, skewness and kurtosis of a hyperbolic distribution. Not expected to be called directly by users.

Usage

```
RLambda(zeta, lambda = 1)
SLambda(zeta, lambda = 1)
MLambda(zeta, lambda = 1)
WLambda1(zeta, lambda = 1)
WLambda2(zeta, lambda = 1)
WLambda3(zeta, lambda = 1)
WLambda4(zeta, lambda = 1)
gammaLambda1(hyperbPi, zeta, lambda = 1)
gammaLambda1(hyperbPi, zeta, lambda = 1)
```

Arguments

hyperbPi	Value of the parameter π of the hyperbolic distribution.
zeta	Value of the parameter ζ of the hyperbolic distribution.
lambda	Parameter related to order of Bessel functions.

Value

The functions RLambda and SLambda are used in the calculation of the mean and variance. They are functions of the Bessel functions of the third kind, implemented in R as besselK. The other functions are used in calculation of higher moments. See Barndorff-Nielsen, O. and Blaesild, P (1981) for details of the calculations.

The parameterisation of the hyperbolic distribution used for this and other components of the HyperbolicDist package is the (π, ζ) one. See hyperbChangePars to transfer between parameterizations.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Richard Trendall, Thomas Tran

References

Barndorff-Nielsen, O. and Blæsild, P (1981). Hyperbolic distributions and ramifications: contributions to theory and application. In *Statistical Distributions in Scientific Work*, eds., Taillie, C., Patil, G. P., and Baldessari, B. A., Vol. 4, pp. 19–44. Dordrecht: Reidel.

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

See Also

`dhyperb`, `hyperbMean`,`hyperbChangePars`, `besselK`

`Generalized Inverse Gaussian` *Generalized Inverse Gaussian Distribution*

Description

Density function, cumulative distribution function, quantile function and random number generation for the generalized inverse Gaussian distribution with parameter vector `Theta`. Utility routines are included for the derivative of the density function and to find suitable break points for use in determining the distribution function.

Usage

```
dgig(x, Theta, KOmega = NULL)
pgig(q, Theta, small = 10^(-6), tiny = 10^(-10), deriv = 0.3,
     subdivisions = 100, accuracy = FALSE, ...)
qgig(p, Theta, small = 10^(-6), tiny = 10^(-10), deriv = 0.3,
     nInterpol = 100, subdivisions = 100, ...)
rgig(n, Theta)
rgig1(n, Theta)
ddgig(x, Theta, KOmega = NULL, ...)
gigBreaks (Theta, small = 10^(-6), tiny = 10^(-10), deriv = 0.3, ...)
```

Arguments

`x, q`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations to be generated.
`Theta`	Parameter vector taking the form `c(lambda,chi,psi)` for `rgig`, or `c(chi,psi)` for `rgig1`.

`KOmega`	Sets the value of the Bessel function in the density or derivative of the density. See **Details**.
`small`	Size of a small difference between the distribution function and zero or one. See **Details**.
`tiny`	Size of a tiny difference between the distribution function and zero or one. See **Details**.
`deriv`	Value between 0 and 1. Determines the point where the derivative becomes substantial, compared to its maximum value. See **Details**.
`accuracy`	Uses accuracy calculated by `integrate` to try and determine the accuracy of the distribution function calculation.
`subdivisions`	The maximum number of subdivisions used to integrate the density returning the distribution function.
`nInterpol`	The number of points used in qhyperb for cubic spline interpolation (see `splinefun`) of the distribution function.
`...`	Passes arguments to `uniroot`. See **Details**.

Details

The generalized inverse Gaussian distribution has density

$$f(x) = \frac{(\psi/\chi)^{\frac{\lambda}{2}}}{2K_\lambda(\sqrt{\psi\chi})} x^{\lambda-1} e^{-\frac{1}{2}\left(\chi x^{-1}+\psi x\right)}$$

for $x > 0$, where $K_\lambda()$ is the modified Bessel function of the third kind with order λ.

The generalized inverse Gaussian distribution is investigated in detail in Jörgensen (1982).

Use `gigChangePars` to convert from the (δ, γ), (α, β), or (ω, η) parameterisations to the (χ, ψ), parameterisation used above.

`pgig` breaks the real line into eight regions in order to determine the integral of `dgig`. The break points determining the regions are found by `gigBreaks`, based on the values of `small`, `tiny`, and `deriv`. In the extreme tails of the distribution where the probability is `tiny` according to `gigCalcRange`, the probability is taken to be zero. For the generalized inverse Gaussian distribution the leftmost breakpoint is not affected by the value of `tiny` but is always taken as 0. In the inner part of the distribution, the range is divided in 6 regions, 3 above the mode, and 3 below. On each side of the mode, there are two break points giving the required three regions. The outer break point is where the probability in the tail has the value given by the variable `small`. The inner break point is where the derivative of the density function is `deriv` times the maximum value of the derivative on that side of the mode. In each of the 6 inner regions the numerical integration routine `safeIntegrate` (which is a wrapper for `integrate`) is used to integrate the density `dgig`.

`qgig` use the breakup of the real line into the same 8 regions as `pgig`. For quantiles which fall in the 2 extreme regions, the quantile is returned as `-Inf` or `Inf` as appropriate. In the 6 inner regions `splinefun` is used to fit values of the distribution function generated by `pgig`. The quantiles are then found using the `uniroot` function.

`pgig` and `qgig` may generally be expected to be accurate to 5 decimal places. Unfortunately, when `lambda` is less than -0.5, the accuracy may be as little as 3 decimal places.

Generalized inverse Gaussian observations are obtained via the algorithm of Dagpunar (1989).

Value

dgig gives the density, pgig gives the distribution function, qgig gives the quantile function, and rgig generates random variates. rgig1 generates random variates in the special case where $\lambda = 1$. An estimate of the accuracy of the approximation to the distribution function may be found by setting accuracy = TRUE in the call to phyperb which then returns a list with components value and error.

ddgig gives the derivative of dgig.

gigBreaks returns a list with components:

xTiny	Takes value 0 always.
xSmall	Value such that probability to the left is less than small.
lowBreak	Point to the left of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode
highBreak	Point to the right of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode
xLarge	Value such that probability to the right is less than small.
xHuge	Value such that probability to the right is less than tiny.
modeDist	The mode of the given generalized inverse Gaussian distribution.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Richard Trendall, and Melanie Luen.

References

Dagpunar, J.S. (1989). An easily implemented generalised inverse Gaussian generator. *Commun. Statist. -Simula.*, **18**, 703–710.

Jörgensen, B. (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution*. Lecture Notes in Statistics, Vol. 9, Springer-Verlag, New York.

See Also

safeIntegrate, integrate for its shortfalls, splinefun, uniroot and gigChangePars for changing parameters to the (χ, ψ) parameterisation, dghyp for the generalized hyperbolic distribution.

Examples

```
Theta <- c(1,2,3)
gigRange <- gigCalcRange(Theta, tol = 10^(-3))
par(mfrow = c(1,2))
curve(dgig(x, Theta), from = gigRange[1], to = gigRange[2],
      n = 1000)
title("Density of the\n Generalized Inverse Gaussian")
curve(pgig(x, Theta), from = gigRange[1], to = gigRange[2],
      n = 1000)
title("Distribution Function of the\n Generalized Inverse Gaussian")
dataVector <- rgig(500, Theta)
```

```
curve(dgig(x, Theta), range(dataVector)[1], range(dataVector)[2],
      n = 500)
hist(dataVector, freq = FALSE, add =TRUE)
title("Density and Histogram\n of the Generalized Inverse Gaussian")
logHist(dataVector, main =
   "Log-Density and Log-Histogram\n of the Generalized Inverse Gaussian")
curve(log(dgig(x, Theta)), add = TRUE,
      range(dataVector)[1], range(dataVector)[2], n = 500)
par(mfrow = c(2,1))
curve(dgig(x, Theta), from = gigRange[1], to = gigRange[2],
      n = 1000)
title("Density of the\n Generalized Inverse Gaussian")
curve(ddgig(x, Theta), from = gigRange[1], to = gigRange[2],
      n = 1000)
title("Derivative of the Density\n of the Generalized Inverse Gaussian")
par(mfrow = c(1,1))
gigRange <- gigCalcRange(Theta, tol = 10^(-6))
curve(dgig(x, Theta), from = gigRange[1], to = gigRange[2],
      n = 1000)
bks <- gigBreaks(Theta)
abline(v = bks)
```

`GeneralizedHyperbolic` *Generalized Hyperbolic Distribution*

Description

Density function, distribution function, quantiles and random number generation for the generalized hyperbolic distribution with parameter vector `Theta`. Utility routines are included for the derivative of the density function and to find suitable break points for use in determining the distribution function.

Usage

```
dghyp(x, Theta)
pghyp(q, Theta, small = 10^(-6), tiny = 10^(-10),
      deriv = 0.3, subdivisions = 100, accuracy = FALSE, ...)
qghyp(p, Theta, small = 10^(-6), tiny = 10^(-10),
      deriv = 0.3, nInterpol = 100, subdivisions = 100, ...)
rghyp(n, Theta)
ddghyp(x, Theta)
ghypBreaks(Theta, small = 10^(-6), tiny = 10^(-10), deriv = 0.3, ...)
```

Arguments

`x,q`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations to be generated.

Theta	Parameter vector taking the form c(lambda,alpha,beta,delta,mu).
small	Size of a small difference between the distribution function and zero or one. See **Details**.
tiny	Size of a tiny difference between the distribution function and zero or one. See **Details**.
deriv	Value between 0 and 1. Determines the point where the derivative becomes substantial, compared to its maximum value. See **Details**.
accuracy	Uses accuracy calculated by~integrate to try and determine the accuracy of the distribution function calculation.
subdivisions	The maximum number of subdivisions used to integrate the density returning the distribution function.
nInterpol	The number of points used in qghyp for cubic spline interpolation (see splinefun) of the distribution function.
...	Passes arguments to uniroot. See **Details**.

Details

The generalized hyperbolic distribution has density

$$f(x) = c(\lambda, \alpha, \beta, \delta) \times \frac{K_{\lambda-1/2}(\alpha\sqrt{\delta^2 + (x-\mu)^2})}{(\sqrt{\delta^2 + (x-\mu)^2}/\alpha)^{1/2-\lambda}} e^{\beta(x-\mu)}$$

where $K_\nu()$ is the modified Bessel function of the third kind with order ν, and

$$c(\lambda, \alpha, \beta, \delta) = \frac{(\sqrt{\alpha^2 - \beta^2}/\delta)^\lambda}{\sqrt{2\pi} K_\lambda(\delta\sqrt{\alpha^2 - \beta^2})}$$

Use ghypChangePars to convert from the (ζ, ρ), (ξ, χ), or $(\bar{\alpha}, \bar{\beta})$ parameterisations to the (α, β) parameterisation used above.

pghyp breaks the real line into eight regions in order to determine the integral of dghyp. The break points determining the regions are found by ghypBreaks, based on the values of small, tiny, and deriv. In the extreme tails of the distribution where the probability is tiny according to ghypCalcRange, the probability is taken to be zero. In the inner part of the distribution, the range is divided in 6 regions, 3 above the mode, and 3 below. On each side of the mode, there are two break points giving the required three regions. The outer break point is where the probability in the tail has the value given by the variable small. The inner break point is where the derivative of the density function is deriv times the maximum value of the derivative on that side of the mode. In each of the 6 inner regions the numerical integration routine safeIntegrate (which is a wrapper for integrate) is used to integrate the density dghyp.

qghyp use the breakup of the real line into the same 8 regions as pghyp. For quantiles which fall in the 2 extreme regions, the quantile is returned as -Inf or Inf as appropriate. In the 6 inner regions splinefun is used to fit values of the distribution function generated by pghyp. The quantiles are then found using the uniroot function.

pghyp and qghyp may generally be expected to be accurate to 5 decimal places.

The generalized hyperbolic distribution is discussed in Bibby and Sörenson (2003). It can be represented as a particular mixture of the normal distribution where the mixing distribution is the generalized inverse Gaussian. rghyp uses this representation to generate observations from the generalized hyperbolic distribution. Generalized inverse Gaussian observations are obtained via the algorithm of Dagpunar (1989) which is implemented in rgig.

Value

dghyp gives the density, pghyp gives the distribution function, qghyp gives the quantile function and rghyp generates random variates. An estimate of the accuracy of the approximation to the distribution function may be found by setting accuracy=TRUE in the call to pghyp which then returns a list with components value and error.

ddghyp gives the derivative of dghyp.

ghypBreaks returns a list with components:

xTiny	Value such that probability to the left is less than tiny.
xSmall	Value such that probability to the left is less than small.
lowBreak	Point to the left of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode.
highBreak	Point to the right of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode.
xLarge	Value such that probability to the right is less than small.
xHuge	Value such that probability to the right is less than tiny.
modeDist	The mode of the given generalized hyperbolic distribution.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Richard Trendall

References

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

Bibby, B. M. and Sörenson, M. (2003). Hyperbolic processes in finance. In *Handbook of Heavy Tailed Distributions in Finance*, ed., Rachev, S. T. pp. 212–248. Elsevier Science B.~V.

Dagpunar, J.S. (1989). An easily implemented generalised inverse Gaussian generator *Commun. Statist. -Simula.*, **18**, 703–710.

Prause, K. (1999) *The generalized hyperbolic models: Estimation, financial derivatives and risk measurement.* PhD Thesis, Mathematics Faculty, University of Freiburg.

See Also

dhyperb for the hyperbolic distribution, dgig for the generalized inverse Gaussian distribution safeIntegrate, integrate for its shortfalls, splinefun, uniroot and ghypChangePars for changing parameters to the (α, β) parameterisation

Examples

```
Theta <- c(1/2,3,1,1,0)
ghypRange <- ghypCalcRange(Theta, tol = 10^(-3))
par(mfrow = c(1,2))
curve(dghyp(x, Theta), from = ghypRange[1], to = ghypRange[2],
      n = 1000)
title("Density of the\n Generalized Hyperbolic Distribution")
curve(pghyp(x, Theta), from = ghypRange[1], to = ghypRange[2],
      n = 1000)
title("Distribution Function of the\n Generalized Hyperbolic Distribution")
dataVector <- rghyp(500, Theta)
curve(dghyp(x, Theta), range(dataVector)[1], range(dataVector)[2],
      n = 500)
hist(dataVector, freq = FALSE, add =TRUE)
title("Density and Histogram of the\n Generalized Hyperbolic Distribution")
logHist(dataVector, main =
   "Log-Density and Log-Histogramof the\n Generalized Hyperbolic Distribution")
curve(log(dghyp(x, Theta)), add = TRUE,
      range(dataVector)[1], range(dataVector)[2], n = 500)
par(mfrow = c(2,1))
curve(dghyp(x, Theta), from = ghypRange[1], to = ghypRange[2],
      n = 1000)
title("Density of the\n Generalized Hyperbolic Distribution")
curve(ddghyp(x, Theta), from = ghypRange[1], to = ghypRange[2],
      n = 1000)
title("Derivative of the Density of the\n Generalized Hyperbolic Distribution")
par(mfrow = c(1,1))
ghypRange <- ghypCalcRange(Theta, tol = 10^(-6))
curve(dghyp(x, Theta), from = ghypRange[1], to = ghypRange[2],
      n = 1000)
bks <- ghypBreaks(Theta)
abline(v = bks)
```

GeneralizedHyperbolicPlots

Generalized Hyperbolic Quantile-Quantile and Percent-Percent Plots

---

Description

qqghyp produces a generalized hyperbolic Q-Q plot of the values in y.

ppghyp produces a generalized hyperbolic P-P (percent-percent) or probability plot of the values in y.

Graphical parameters may be given as arguments to qqghyp, and ppghyp.

Usage

```
qqghyp(y, Theta, main = "Generalized Hyperbolic Q-Q Plot",
  xlab = "Theoretical Quantiles",
```

```
  ylab = "Sample Quantiles",
  plot.it = TRUE, line = TRUE, ...)

ppghyp(y, Theta, main = "Generalized Hyperbolic P-P Plot",
  xlab = "Uniform Quantiles",
  ylab = "Probability-integral-transformed Data",
  plot.it = TRUE, line = TRUE, ...)
```

Arguments

`y`	The data sample.
`Theta`	Parameters of the generalized hyperbolic distribution.
`xlab, ylab, main`	Plot labels.
`plot.it`	Logical. Should the result be plotted?
`line`	Add line through origin with unit slope.
`...`	Further graphical parameters.

Value

For qqghyp and ppghyp, a list with components:

`x`	The x coordinates of the points that are to be plotted.
`y`	The y coordinates of the points that are to be plotted.

References

Wilk, M. B. and Gnanadesikan, R. (1968) Probability plotting methods for the analysis of data. *Biometrika*. **55**, 1–17.

See Also

`ppoints`, `dghyp`.

Examples

```
par(mfrow = c(1,2))
y <- rghyp(200, c(2,2,1,2,2))
qqghyp(y, c(2,2,1,2,2),line = FALSE)
abline(0, 1, col = 2)
ppghyp(y, c(2,2,1,2,2))
```

ghypCalcRange *Range of a Generalized Hyperbolic Distribution*

Description

Given the parameter vector Theta of a generalized hyperbolic distribution, this function determines the range outside of which the density function is negligible, to a specified tolerance. The parameterization used is the (α, β) one (see `dghyp`). To use another parameterization, use `ghypChangePars`.

Usage

```
ghypCalcRange(Theta, tol = 10^(-5), density = TRUE, ...)
```

Arguments

`Theta`	Value of parameter vector specifying the hyperbolic distribution.
`tol`	Tolerance.
`density`	Logical. If `TRUE`, the bounds are for the density function. If `FALSE`, they should be for the probability distribution, but this has not yet been implemented.
`...`	Extra arguments for calls to `uniroot`.

Details

The particular generalized hyperbolic distribution being considered is specified by the value of the parameter value `Theta`.

If `density = TRUE`, the function gives a range, outside of which the density is less than the given tolerance. Useful for plotting the density. Also used in determining break points for the separate sections over which numerical integration is used to determine the distribution function. The points are found by using `uniroot` on the density function.

If `density = FALSE`, the function returns the message: `"Distribution function bounds not yet implemented"`.

Value

A two-component vector giving the lower and upper ends of the range.

Author(s)

David Scott `<d.scott@auckland.ac.nz>`

References

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

See Also

dghyp, ghypChangePars

Examples

```
Theta <- c(1,5,3,1,0)
maxDens <- dghyp(ghypMode(Theta), Theta)
ghypRange <- ghypCalcRange(Theta, tol = 10^(-3)*maxDens)
ghypRange
curve(dghyp(x, Theta), ghypRange[1], ghypRange[2])
## Not run: ghypCalcRange(Theta, tol = 10^(-3), density = FALSE)
```

---

ghypChangePars *Change Parameterizations of the Generalized Hyperbolic Distribution*

Description

This function interchanges between the following 4 parameterizations of the generalized hyperbolic distribution:

1. $\lambda, \alpha, \beta, \delta, \mu$
2. $\lambda, \zeta, \rho, \delta, \mu$
3. $\lambda, \xi, \chi, \delta, \mu$
4. $\lambda, \bar{\alpha}, \bar{\beta}, \delta, \mu$

These are the parameterizations given in Prause (1999)

Usage

```
ghypChangePars(from, to, Theta, noNames = FALSE)
```

Arguments

`from`	The set of parameters to change from.
`to`	The set of parameters to change to.
`Theta`	"from" parameter vector consisting of 5 numerical elements.
`noNames`	Logical. When `TRUE`, suppresses the parameter `names` in the output.

Details

In the 4 parameterizations, the following must be positive:

1. α, δ
2. ζ, δ
3. ξ, δ
4. $\bar{\alpha}, \delta$

Furthermore, note that in the first parameterization α must be greater than the absolute value of β; in the third parameterization, ξ must be less than one, and the absolute value of χ must be less than ξ; and in the fourth parameterization, $\bar{\alpha}$ must be greater than the absolute value of $\bar{\beta}$.

Value

A numerical vector of length 5 representing `Theta` in the `to` parameterization.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Jennifer Tso, Richard Trendall

References

Barndorff-Nielsen, O. and Blæsild, P. (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

Prause, K. (1999) *The generalized hyperbolic models: Estimation, financial derivatives and risk measurement.* PhD Thesis, Mathematics Faculty, University of Freiburg.

See Also

`dghyp`

Examples

```
Theta1 <- c(2,2,1,3,0)                    # Parameterization 1
Theta2 <- ghypChangePars(1, 2, Theta1)    # Convert to parameterization 2
Theta2                                    # Parameterization 2
ghypChangePars(2, 1, as.numeric(Theta2)) # Convert back to parameterization 1
```

`ghypMom` *Calculate Moments of the Generalized Hyperbolic Distribution*

Description

Function to calculate raw moments, mu moments, central moments and moments about any other given location for the generalized hyperbolic distribution.

Usage

```
ghypMom(order, Theta, momType = "raw", about = 0)
```

Arguments

`order`	Numeric. The order of the moment to be calculated. Not permitted to be a vector. Must be a positive whole number except for moments about zero.
`Theta`	Numeric. The parameter vector specifying the GIG distribution. Of the form `c(lambda, alpha, beta, delta, mu)` (see `dghyp`).
`momType`	Common types of moments to be calculated, default is "raw". See **Details**.
`about`	Numeric. The point around which the moment is to be calculated.

Details

Checking whether order is a whole number is carried out using the function is.wholenumber.

momType can be either "raw" (moments about zero), "mu" (moments about mu), or "central" (moments about mean). If one of these moment types is specified, then there is no need to specify the about value. For moments about any other location, the about value must be specified. In the case that both momType and about are specified and contradicting, the function will always calculate the moments based on about rather than momType.

To calculate moments of the generalized hyperbolic distribution, the function firstly calculates mu moments by formula defined below and then transforms mu moments to central moments or raw moments or moments about any other locations as required by calling momChangeAbout.

The mu moments are obtained from the recursion formula given in Scott, WÃ¼rtz and Tran (2008).

Value

The moment specified.

Author(s)

David Scott <d.scott@auckland.ac.nz>

References

Scott, D. J., WÃ¼rtz, D. and Tran, T. T. (2008) Moments of the Generalized Hyperbolic Distribution. Preprint.

See Also

ghypChangePars, is.wholenumber, momChangeAbout, momIntegrated, ghypMean, ghypVar, ghypSkew, ghypKurt.

Examples

```
Theta <- c(2,2,1,2,1)
mu <- Theta[5]
### mu moments
(m1 <- ghypMean(Theta))
m1 - mu
ghypMom(1, Theta, momType = "mu")
momIntegrated("ghyp", order = 1, param = Theta, about = mu)
ghypMom(2, Theta, momType = "mu")
momIntegrated("ghyp", order = 2, param = Theta, about = mu)
ghypMom(10, Theta, momType = "mu")
momIntegrated("ghyp", order = 10, param = Theta, about = mu)

### raw moments
ghypMean(Theta)
ghypMom(1, Theta, momType = "raw")
momIntegrated("ghyp", order = 1, param = Theta, about = 0)
ghypMom(2, Theta, momType = "raw")
momIntegrated("ghyp", order = 2, param = Theta, about = 0)
```

```
ghypMom(10, Theta, momType = "raw")
momIntegrated("ghyp", order = 10, param = Theta, about = 0)

### central moments
ghypMom(1, Theta, momType = "central")
momIntegrated("ghyp", order = 1, param = Theta, about = m1)
ghypVar(Theta)
ghypMom(2, Theta, momType = "central")
momIntegrated("ghyp", order = 2, param = Theta, about = m1)
ghypMom(10, Theta, momType = "central")
momIntegrated("ghyp", order = 10, param = Theta, about = m1)
```

---

gigCalcRange *Range of a Generalized Inverse Gaussian Distribution*

Description

Given the parameter vector Theta of a generalized inverse Gaussian distribution, this function determines the range outside of which the density function is negligible, to a specified tolerance. The parameterization used is the (χ, ψ) one (see dgig). To use another parameterization, use gigChangePars.

Usage

```
gigCalcRange(Theta, tol = 10^(-5), density = TRUE, ...)
```

Arguments

Theta	Value of parameter vector specifying the generalized inverse Gaussian distribution.
tol	Tolerance.
density	Logical. If TRUE, the bounds are for the density function. If FALSE, they should be for the probability distribution, but this has not yet been implemented.
...	Extra arguments for calls to uniroot.

Details

The particular generalized inverse Gaussian distribution being considered is specified by the value of the parameter value Theta.

If density = TRUE, the function gives a range, outside of which the density is less than the given tolerance. Useful for plotting the density. Also used in determining break points for the separate sections over which numerical integration is used to determine the distribution function. The points are found by using uniroot on the density function.

If density = FALSE, the function returns the message: "Distribution function bounds not yet implemented".

Value

A two-component vector giving the lower and upper ends of the range.

Author(s)

David Scott <d.scott@auckland.ac.nz>

References

Jörgensen, B. (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution*. Lecture Notes in Statistics, Vol. 9, Springer-Verlag, New York.

See Also

dgig, gigChangePars

Examples

```
Theta <- c(-0.5,5,2.5)
maxDens <- dgig(gigMode(Theta), Theta)
gigRange <- gigCalcRange(Theta, tol = 10^(-3)*maxDens)
gigRange
curve(dgig(x, Theta), gigRange[1], gigRange[2])
## Not run: gigCalcRange(Theta, tol = 10^(-3), density = FALSE)
```

---

gigChangePars — *Change Parameterizations of the Generalized Inverse Gaussian Distribution*

Description

This function interchanges between the following 4 parameterizations of the generalized inverse Gaussian distribution:

1. (λ, χ, ψ)
2. $(\lambda, \delta, \gamma)$
3. (λ, α, β)
4. (λ, ω, η)

See Jörgensen (1982) and Dagpunar (1989)

Usage

```
gigChangePars(from, to, Theta, noNames = FALSE)
```

Arguments

`from`	The set of parameters to change from.
`to`	The set of parameters to change to.
`Theta`	"`from`" parameter vector consisting of 3 numerical elements.
`noNames`	Logical. When `TRUE`, suppresses the parameter `names` in the output.

Details

The range of λ is the whole real line. In each parameterization, the other two parameters must take positive values.

Value

A numerical vector of length 3 representing `Theta` in the "`to`" parameterization.

Author(s)

David Scott <`d.scott@auckland.ac.nz`>

References

Jörgensen, B. (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution*. Lecture Notes in Statistics, Vol. 9, Springer-Verlag, New York.

Dagpunar, J. S. (1989). An easily implemented generalised inverse Gaussian generator, *Commun. Statist.—Simula.*, **18**, 703–710.

See Also

`dgig`

Examples

```
Theta1 <- c(-0.5,5,2.5)                # Parameterisation 1
Theta2 <- gigChangePars(1, 2, Theta1)  # Convert to parameterization 2
Theta2                                 # Parameterization 2
gigChangePars(2, 1, as.numeric(Theta2)) # Convert back to parameterization 1
```

`gigCheckPars` *Check Parameters of the Generalized Inverse Gaussian Distribution*

Description

Given a putative set of parameters for the generalized inverse Gaussian distribution, the functions checks if they are in the correct range, and if they correspond to the boundary cases.

Usage

```
gigCheckPars(Theta, ...)
```

Arguments

`Theta`	Numeric. Putative parameter values for a generalized inverse Gaussian distribution.
`...`	Further arguments for calls to `all.equal`.

Details

The vector `Theta` takes the form `c(lambda,chi,psi)`.

If either `chi` or `psi` is negative, an error is returned.

If `chi` is 0 (to within tolerance allowed by `all.equal`) then `psi` and `lambda` must be positive or an error is returned. If these conditions are satisfied, the distribution is identified as a gamma distribution.

If `psi` is 0 (to within tolerance allowed by `all.equal`) then `chi` must be positive and `lambda` must be negative or an error is returned. If these conditions are satisfied, the distribution is identified as an inverse gamma distribution.

If both `chi` and `psi` are positive, then the distribution is identified as a normal generalized inverse Gaussian distribution.

Value

A list with components:

`case`	Whichever of `'error'`, `'gamma'`, `invgamma`, or `'normal'` is identified by the function.
`errMessage`	An appropriate error message if an error was found, the empty string `""` otherwise.

Author(s)

David Scott <`d.scott@auckland.ac.nz`>

References

Paolella, Marc S. (2007) Intermediate Probability: A Computational Approach, Chichester: Wiley

See Also

`dgig`

Examples

```
gigCheckPars(c(-0.5,5,2.5))     # normal
gigCheckPars(c(0.5,-5,2.5))     # error
gigCheckPars(c(0.5,5,-2.5))     # error
gigCheckPars(c(0.5,-5,-2.5))    # error
gigCheckPars(c(0.5,0,2.5))      # gamma
gigCheckPars(c(-0.5,0,2.5))     # error
gigCheckPars(c(0.5,0,0))        # error
gigCheckPars(c(-0.5,0,0))       # error
gigCheckPars(c(0.5,5,0))        # error
gigCheckPars(c(-0.5,5,0))       # invgamma
```

---

gigMom — *Calculate Moments of the Generalized Inverse Gaussian Distribution*

Description

Functions to calculate raw moments and moments about a given location for the generalized inverse Gaussian (GIG) distribution, including the gamma and inverse gamma distributions as special cases.

Usage

```
gigRawMom(order, Theta)
gigMom(order, Theta, about = 0)
gammaRawMom(order, shape = 1, rate = 1, scale = 1/rate)
```

Arguments

`order`	Numeric. The order of the moment to be calculated. Not permitted to be a vector. Must be a positive whole number except for moments about zero.
`Theta`	Numeric. The parameter vector specifying the GIG distribution. Of the form `c(lambda, chi, psi)` (see `dgig`).
`about`	Numeric. The point around which the moment is to be calculated.
`shape`	Numeric. The shape parameter, must be non-negative, not permitted to be a vector.
`scale`	Numeric. The scale parameter, must be positive, not permitted to be a vector.
`rate`	Numeric. The rate parameter, an alternative way to specify the scale.

Details

The vector `Theta` of parameters is examined using `gigCheckPars` to see if the parameters are valid for the GIG distribution and if they correspond to the special cases which are the gamma and inverse gamma distributions. Checking of special cases and valid parameter vector values is carried out using the function `gigCheckPars`. Checking whether `order` is a whole number is carried out using the function `is.wholenumber`.

Raw moments (moments about zero) are calculated using the functions `gigRawMom` or `gammaRawMom`. For moments not about zero, the function `momChangeAbout` is used to derive moments about another point from raw moments. Note that raw moments of the inverse gamma distribution can be obtained from the raw moments of the gamma distribution because of the relationship between the two distributions. An alternative implementation of raw moments of the gamma and inverse gamma distributions may be found in the package **actuar** and these may be faster since they are written in C.

To calculate the raw moments of the GIG distribution it is convenient to use the alternative parameterization of the GIG in terms of ω and η, given as parameterization 3 in `gigChangePars`. Then the raw moment of the GIG distribution of order k is given by

$$\eta^k K_{\lambda+k}(\omega)/K_\lambda(\omega)$$

where $K_\lambda()$ is the modified Bessel function of the third kind of order λ.

The raw moment of the gamma distribution of order k with shape parameter α and rate parameter β is given by

$$\beta^{-k}\Gamma(\alpha+k)/\Gamma(\alpha)$$

The raw moment of order k of the inverse gamma distribution with shape parameter α and rate parameter β is the raw moment of order $-k$ of the gamma distribution with shape parameter α and rate parameter $1/\beta$.

Value

The moment specified. In the case of raw moments, `Inf` is returned if the moment is infinite.

Author(s)

David Scott <d.scott@auckland.ac.nz>

References

Paolella, Marc S. (2007) Intermediate Probability: A Computational Approach, Chichester: Wiley

See Also

`gigCheckPars`, `gigChangePars`, `is.wholenumber`, `momChangeAbout`, `momIntegrated`, `gigMean`, `gigVar`, `gigSkew`, `gigKurt`.

Examples

```
### Raw moments of the generalized inverse Gaussian distribution
Theta <- c(-0.5,5,2.5)
gigRawMom(1, Theta)
momIntegrated("gig", order = 1, param = Theta, about = 0)
gigRawMom(2, Theta)
momIntegrated("gig", order = 2, param = Theta, about = 0)
gigRawMom(10, Theta)
momIntegrated("gig", order = 10, param = Theta, about = 0)
gigRawMom(2.5, Theta)
```

```
### Moments of the generalized inverse Gaussian distribution
Theta <- c(-0.5,5,2.5)
(m1 <- gigRawMom(1, Theta))
gigMom(1, Theta)
gigMom(2, Theta, m1)
(m2 <- momIntegrated("gig", order = 2, param = Theta, about = m1))
gigMom(1, Theta, m1)
gigMom(3, Theta, m1)
momIntegrated("gig", order = 3, param = Theta, about = m1)

### Raw moments of the gamma distribution
shape <- 2
rate <- 3
Theta <- c(shape, rate)
gammaRawMom(1, shape, rate)
momIntegrated("gamma", order = 1, param = Theta, about = 0)
gammaRawMom(2, shape, rate)
momIntegrated("gamma", order = 2, param = Theta, about = 0)
gammaRawMom(10, shape, rate)
momIntegrated("gamma", order = 10, param = Theta, about = 0)

### Moments of the inverse gamma distribution
Theta <- c(-0.5,5,0)
gigRawMom(2, Theta)              # Inf
gigRawMom(-2, Theta)
momIntegrated("invgamma", order = -2,
              param = c(-Theta[1],Theta[2]/2), about = 0)

### An example where the moment is infinite: inverse gamma
Theta <- c(-0.5,5,0)
gigMom(1, Theta)
gigMom(2, Theta)
```

GIGPlots *Generalized Inverse Gaussian Quantile-Quantile and Percent-Percent Plots*

Description

qqgig produces a generalized inverse Gaussian QQ plot of the values in y.

ppgig produces a generalized inverse Gaussian PP (percent-percent) or probability plot of the values in y.

If line = TRUE, a line with zero intercept and unit slope is added to the plot.

Graphical parameters may be given as arguments to qqgig, and ppgig.

Usage

```
qqgig(y, Theta, main = "GIG Q-Q Plot",
  xlab = "Theoretical Quantiles",
  ylab = "Sample Quantiles",
  plot.it = TRUE, line = TRUE, ...)

ppgig(y, Theta, main = "GIG P-P Plot",
  xlab = "Uniform Quantiles",
  ylab = "Probability-integral-transformed Data",
  plot.it = TRUE, line = TRUE, ...)
```

Arguments

`y`	The data sample.
`Theta`	Parameters of the generalized inverse Gaussian distribution.
`xlab, ylab, main`	Plot labels.
`plot.it`	Logical. TRUE denotes the results should be plotted.
`line`	Logical. If TRUE, a line with zero intercept and unit slope is added to the plot.
`...`	Further graphical parameters.

Value

For `qqgig` and `ppgig`, a list with components:

`x`	The x coordinates of the points that are be plotted.
`y`	The y coordinates of the points that are be plotted.

References

Wilk, M. B. and Gnanadesikan, R. (1968) Probability plotting methods for the analysis of data. *Biometrika*. **55**, 1–17.

See Also

`ppoints`, `dgig`.

Examples

```
par(mfrow=c(1,2))
y <- rgig(1000,c(1,2,3))
qqgig(y,c(1,2,3),line=FALSE)
abline(0,1,col=2)
ppgig(y,c(1,2,3))
```

hyperbCalcRange *Range of a Hyperbolic Distribution*

Description

Given the parameter vector Theta of a hyperbolic distribution, this function calculates the range outside of which the distribution has negligible probability, or the density function is negligible, to a specified tolerance. The parameterization used is the (π, ζ) one (see dhyperb). To use another parameterization, use hyperbChangePars.

Usage

```
hyperbCalcRange(Theta, tol = 10^(-5), density = FALSE)
```

Arguments

Theta	Value of parameter vector specifying the hyperbolic distribution.
tol	Tolerance.
density	Logical. If FALSE, the bounds are for the probability distribution. If TRUE, they are for the density function.

Details

The particular hyperbolic distribution being considered is specified by the value of the parameter value Theta.

If density = FALSE, the function calculates the effective range of the distribution, which is used in calculating the distribution function and quantiles, and may be used in determining the range when plotting the distribution. By effective range is meant that the probability of an observation being greater than the upper end is less than the specified tolerance tol. Likewise for being smaller than the lower end of the range.

If density = TRUE, the function gives a range, outside of which the density is less than the given tolerance. Useful for plotting the density.

Value

A two-component vector giving the lower and upper ends of the range.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Jennifer Tso, Richard Trendall

References

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

See Also

dhyperb, hyperbChangePars

Examples

```
par(mfrow = c(1,2))
Theta <- c(3,5,1,0)
hyperbRange <- hyperbCalcRange(Theta, tol = 10^(-3))
hyperbRange
curve(phyperb(x, Theta), hyperbRange[1], hyperbRange[2])
maxDens <- dhyperb(hyperbMode(Theta), Theta)
hyperbRange <- hyperbCalcRange(Theta, tol = 10^(-3)*maxDens, density = TRUE)
hyperbRange
curve(dhyperb(x, Theta), hyperbRange[1], hyperbRange[2])
```

---

hyperbChangePars *Change Parameterizations of the Hyperbolic Distribution*

Description

This function interchanges between the following 4 parameterizations of the hyperbolic distribution:

1. π, ζ, δ, μ
2. $\alpha, \beta, \delta, \mu$
3. $\phi, \gamma, \delta, \mu$
4. ξ, χ, δ, μ

The first three are given in Barndorff-Nielsen and Blæsild (1983), and the fourth in Prause (1999)

Usage

```
hyperbChangePars(from, to, Theta, noNames = FALSE)
```

Arguments

`from`	The set of parameters to change from.
`to`	The set of parameters to change to.
`Theta`	"from" parameter vector consisting of 4 numerical elements.
`noNames`	Logical. When `TRUE`, suppresses the parameter `names` in the output.

Details

In the 4 parameterizations, the following must be positive:

1. ζ, δ
2. α, δ
3. ϕ, γ, δ
4. ξ, δ

Furthermore, note that in the second parameterization α must be greater than the absolute value of β, while in the fourth parameterization, ξ must be less than one, and the absolute value of χ must be less than ξ.

Value

A numerical vector of length 4 representing `Theta` in the `to` parameterization.

Author(s)

David Scott <`d.scott@auckland.ac.nz`>, Jennifer Tso, Richard Trendall

References

Barndorff-Nielsen, O. and Blæsild, P. (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

Prause, K. (1999) *The generalized hyperbolic models: Estimation, financial derivatives and risk measurement.* PhD Thesis, Mathematics Faculty, University of Freiburg.

See Also

`dhyperb`

Examples

```
Theta1 <- c(-2,1,3,0)                      # Parameterization 1
Theta2 <- hyperbChangePars(1, 2, Theta1)   # Convert to parameterization 2
Theta2                                     # Parameterization 2
hyperbChangePars(2, 1, as.numeric(Theta2)) # Convert back to parameterization 1
```

`hyperbCvMTest` *Cramer-von~Mises Test of a Hyperbolic Distribution*

Description

Carry out a Crämer-von~Mises test of a hyperbolic distribution where the parameters of the distribution are estimated, or calculate the p-value for such a test.

Usage

```
hyperbCvMTest(x, Theta, conf.level = 0.95, ...)
hyperbCvMTestPValue(xi, chi, Wsq, digits = 3)
## S3 method for class 'hyperbCvMTest'
print(x, prefix = "\t", ...)
```

Arguments

x	A numeric vector of data values for hyperbCvMTest, or object of class "hyperbCvMTest" for print.hyperbCvMTest.
Theta	Parameters of the hyperbolic distribution taking the form c(pi,zeta,delta,mu).
conf.level	Confidence level of the the confidence interval.
...	Further arguments to be passed to or from methods.
xi	Value of ξ in the (ξ, χ) parameterization of the hyperbolic distribution.
chi	Value of χ in the (ξ, χ) parameterisation of the hyperbolic distribution.
Wsq	Value of the test statistic in the Crämer-von~Mises test of the hyperbolic distribution.
digits	Number of decimal places for p-value.
prefix	Character(s) to be printed before the description of the test.

Details

hyperbCvMTest carries out a Crämer-von~Mises goodness-of-fit test of the hyperbolic distribution. The parameter Theta must be given in the (π, ζ) parameterisation.

hyperbCvMTestPValue calculates the p-value of the test, and is not expected to be called by the user. The method used is interpolation in Table 5 given in Puig & Stephens (2001), which assumes all the parameters of the distribution are unknown. Since the table used is limited, large p-values are simply given as ">~0.25" and very small ones as "<~0.01". The table is created as the matrix wsqTable when the package HyperbolicDist is invoked.

print.hyperbCvMTest prints the output from the Crämer-von~Mises goodness-of-fit test for the hyperbolic distribution in very similar format to that provided by print.htest. The only reason for having a special print method is that p-values can be given as less than some value or greater than some value, such as "<\ ~0.01", or ">\ ~0.25".

Value

hyperbCvMTest returns a list with class hyperbCvMTest containing the following components:

statistic	The value of the test statistic.
method	A character string with the value "Crämer-von~Mises test of hyperbolic distribution".
data.name	A character string giving the name(s) of the data.
parameter	The value of the parameter Theta.
p.value	The p-value of the test.

`warn` A warning if the parameter values are outside the limits of the table given in Puig & Stephens (2001).

`hyperbCvMTestPValue` returns a list with the elements `p.value` and `warn` only.

Author(s)

David Scott, Thomas Tran

References

Puig, Pedro and Stephens, Michael A. (2001), Goodness-of-fit tests for the hyperbolic distribution. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, **29**, 309–320.

Examples

```
Theta <- c(2,2,2,2)
dataVector <- rhyperb(500, Theta)
fittedTheta <- hyperbFit(dataVector)$Theta
hyperbCvMTest(dataVector, fittedTheta)
dataVector <- rnorm(1000)
fittedTheta <- hyperbFit(dataVector, startValues = "FN")$Theta
hyperbCvMTest(dataVector, fittedTheta)
```

`hyperbFit` *Fit the Hyperbolic Distribution to Data*

Description

Fits a hyperbolic distribution to data. Displays the histogram, log-histogram (both with fitted densities), Q-Q plot and P-P plot for the fit which has the maximum likelihood.

Usage

```
  hyperbFit(x, freq = NULL, breaks = NULL, ThetaStart = NULL,
            startMethod = "Nelder-Mead", startValues = "BN",
            method = "Nelder-Mead", hessian = FALSE,
            plots = FALSE, printOut = FALSE,
            controlBFGS = list(maxit=200),
            controlNM = list(maxit=1000), maxitNLM = 1500, ...)

  ## S3 method for class 'hyperbFit'
print(x,
        digits = max(3, getOption("digits") - 3), ...)

  ## S3 method for class 'hyperbFit'
plot(x, which = 1:4,
       plotTitles = paste(c("Histogram of ","Log-Histogram of ",
```

```
                            "Q-Q Plot of ","P-P Plot of "), x$obsName,
                          sep = ""),
      ask = prod(par("mfcol")) < length(which) && dev.interactive(), ...)
```

Arguments

`x`	Data vector for `hyperbFit`. Object of class `"hyperbFit"` for `print.hyperbFit` and `plot.hyperbFit`.
`freq`	A vector of weights with length equal to `length(x)`.
`breaks`	Breaks for histogram, defaults to those generated by `hist(x, right = FALSE, plot = FALSE)`.
`ThetaStart`	A user specified starting parameter vector Theta taking the form `c(pi,zeta,delta,mu)`.
`startMethod`	Method used by `hyperbFitStart` in calls to `optim`.
`startValues`	Code giving the method of determining starting values for finding the maximum likelihood estimate of Theta.
`method`	Different optimisation methods to consider. See **Details**.
`hessian`	Logical. If `TRUE` the value of the hessian is returned.
`plots`	Logical. If `FALSE` suppresses printing of the histogram, log-histogram, Q-Q plot and P-P plot.
`printOut`	Logical. If `FALSE` suppresses printing of results of fitting.
`controlBFGS`	A list of control parameters for `optim` when using the `"BFGS"` optimisation.
`controlNM`	A list of control parameters for `optim` when using the `"Nelder-Mead"` optimisation.
`maxitNLM`	A positive integer specifying the maximum number of iterations when using the `"nlm"` optimisation.
`digits`	Desired number of digits when the object is printed.
`which`	If a subset of the plots is required, specify a subset of the numbers `1:4`.
`plotTitles`	Titles to appear above the plots.
`ask`	Logical. If `TRUE`, the user is *ask*ed before each plot, see `par(ask = .)`.
`...`	Passes arguments to `par`, `hist`, `logHist`, `qqhyperb` and `pphyperb`.

Details

`startMethod` can be either `"BFGS"` or `"Nelder-Mead"`.

`startValues` can be one of the following:

- `"US"`User-supplied.
- `"BN"`Based on Barndorff-Nielsen (1977).
- `"FN"`A fitted normal distribution.
- `"SL"`Based on a fitted skew-Laplace distribution.
- `"MoM"`Method of moments.

For the details concerning the use of `ThetaStart`, `startMethod`, and `startValues`, see `hyperbFitStart`.

The three optimisation methods currently available are:

- "BFGS"Uses the quasi-Newton method "BFGS" as documented in optim.
- "Nelder-Mead"Uses an implementation of the Nelder and Mead method as documented in optim.
- "nlm"Uses the nlm function in R.

For details of how to pass control information for optimisation using optim and nlm, see optim and nlm.

When method = "nlm" is used, warnings may be produced. These do not appear to be a problem.

Value

A list with components:

Theta	A vector giving the maximum likelihood estimate of Theta, as (pi,zeta,delta,mu).
maxLik	The value of the maximised log-likelihood.
hessian	If hessian was set to TRUE, the value of the hessian. Not present otherwise.
method	Optimisation method used.
conv	Convergence code. See the relevant documentation (either optim or nlm) for details on convergence.
iter	Number of iterations of optimisation routine.
x	The data used to fit the hyperbolic distribution.
xName	A character string with the actual x argument name.
ThetaStart	Starting value of Theta returned by call to hyperbFitStart.
svName	Descriptive name for the method finding start values.
startValues	Acronym for the method of finding start values.
KNu	Value of the Bessel function in the fitted density.
breaks	The cell boundaries found by a call to hist.
midpoints	The cell midpoints found by a call to hist.
empDens	The estimated density found by a call to hist.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Ai-Wei Lee, Jennifer Tso, Richard Trendall, Thomas Tran

References

Barndorff-Nielsen, O. (1977) Exponentially decreasing distributions for the logarithm of particle size, *Proc. Roy. Soc. Lond.*, A**353**, 401–419.

Fieller, N. J., Flenley, E. C. and Olbricht, W. (1992) Statistics of particle size data. *Appl. Statist.*, **41**, 127–146.

See Also

optim, nlm, par, hist, logHist, qqhyperb, pphyperb, dskewlap and hyperbFitStart.

Examples

```
Theta <- c(2,2,2,2)
dataVector <- rhyperb(500, Theta)
## See how well hyperbFit works
hyperbFit(dataVector)
hyperbFit(dataVector, plots = TRUE)
fit <- hyperbFit(dataVector)
par(mfrow = c(1,2))
plot(fit, which = c(1,3))

## Use nlm instead of default
hyperbFit(dataVector, method = "nlm")
```

---

hyperbFitStart *Find Starting Values for Fitting a Hyperbolic Distribution*

Description

Finds starting values for input to a maximum likelihood routine for fitting hyperbolic distribution to data.

Usage

```
hyperbFitStart(x, breaks = NULL, startValues = "BN",
               ThetaStart = NULL, startMethodSL = "Nelder-Mead",
               startMethodMoM = "Nelder-Mead", ...)
hyperbFitStartMoM(x, startMethodMoM = "Nelder-Mead", ...)
```

Arguments

`x`	Data vector.
`breaks`	Breaks for histogram. If missing, defaults to those generated by `hist(x, right = FALSE, plot = FALSE`
`startValues`	Vector of the different starting values to consider. See **Details**.
`ThetaStart`	Starting values for Theta if `startValues = "US"`.
`startMethodSL`	Method used by call to `optim` in finding skew Laplace estimates.
`startMethodMoM`	Method used by call to `optim` in finding method of moments estimates.
`...`	Passes arguments to `optim`.

Details

Possible values of the argument `startValues` are the following:

- `"US"`User-supplied.
- `"BN"`Based on Barndorff-Nielsen (1977).
- `"FN"`A fitted normal distribution.

- "SL"Based on a fitted skew-Laplace distribution.
- "MoM"Method of moments.

If startValues = "US" then a value must be supplied for ThetaStart.

If startValues = "MoM", hyperbFitStartMoM is called. These starting values are based on Barndorff-Nielsen *et al* (1985).

If startValues = "SL", or startValues = "MoM" an initial optimisation is needed to find the starting values. These optimisations call optim.

Value

hyperbFitStart returns a list with components:

ThetaStart	A vector with elements pi, lZeta (log of zeta), lDelta (log of delta), and mu giving the starting value of Theta.
xName	A character string with the actual x argument name.
breaks	The cell boundaries found by a call to hist.
midpoints	The cell midpoints found by a call to hist.
empDens	The estimated density found by a call to hist.

hyperbFitStartMoM returns only the method of moments estimates as a vector with elements pi, lZeta (log of zeta), lDelta (log of delta), and mu.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Ai-Wei Lee, Jennifer Tso, Richard Trendall, Thomas Tran

References

Barndorff-Nielsen, O. (1977) Exponentially decreasing distributions for the logarithm of particle size, *Proc. Roy. Soc. Lond.*, A**353**, 401–419.

Barndorff-Nielsen, O., Blæsild, P., Jensen, J., and Sörenson, M. (1985). The fascination of sand. In *A celebration of statistics, The ISI Centenary Volume*, eds., Atkinson, A. C. and Fienberg, S. E., pp. 57–87. New York: Springer-Verlag.

Fieller, N. J., Flenley, E. C. and Olbricht, W. (1992) Statistics of particle size data. *Appl. Statist.*, **41**, 127–146.

See Also

HyperbolicDistribution, dskewlap, hyperbFit, hist, and optim.

Examples

```
Theta <- c(2,2,2,2)
dataVector <- rhyperb(500,Theta)
hyperbFitStart(dataVector,startValues="FN")
hyperbFitStartMoM(dataVector)
hyperbFitStart(dataVector,startValues="MoM")
```

Hyperbolic *Hyperbolic Distribution*

Description

Density function, distribution function, quantiles and random number generation for the hyperbolic distribution with parameter vector Theta. Utility routines are included for the derivative of the density function and to find suitable break points for use in determining the distribution function.

Usage

```
dhyperb(x, Theta, KNu = NULL, logPars = FALSE)
phyperb(q, Theta, small = 10^(-6), tiny = 10^(-10),
        deriv = 0.3, subdivisions = 100, accuracy = FALSE, ...)
qhyperb(p, Theta, small = 10^(-6), tiny = 10^(-10),
        deriv = 0.3, nInterpol = 100, subdivisions = 100, ...)
rhyperb(n, Theta)
ddhyperb(x, Theta, KNu = NULL, ...)
hyperbBreaks(Theta, small = 10^(-6), tiny = 10^(-10), deriv = 0.3, ...)
```

Arguments

x,q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations to be generated.
Theta	Parameter vector taking the form c(pi,zeta,delta,mu).
KNu	Sets the value of the Bessel function in the density or derivative of the density. See **Details**.
logPars	Logical; if TRUE the second and third components of Theta are taken to be log(zeta) and log(delta) respectively.
small	Size of a small difference between the distribution function and zero or one. See **Details**.
tiny	Size of a tiny difference between the distribution function and zero or one. See **Details**.
deriv	Value between 0 and 1. Determines the point where the derivative becomes substantial, compared to its maximum value. See **Details**.
accuracy	Uses accuracy calculated by integrate to try and determine the accuracy of the distribution function calculation.
subdivisions	The maximum number of subdivisions used to integrate the density returning the distribution function.
nInterpol	The number of points used in qhyperb for cubic spline interpolation (see splinefun) of the distribution function.
...	Passes arguments to uniroot. See **Details**.

Details

The hyperbolic distribution has density

$$f(x) = \frac{1}{2\sqrt{1+\pi^2}K_1(\zeta)} e^{-\zeta[\sqrt{1+\pi^2}\sqrt{1+(\frac{x-\mu}{\delta})^2}-\pi\frac{x-\mu}{\delta}]}$$

where $K_1()$ is the modified Bessel function of the third kind with order 1.

A succinct description of the hyperbolic distribution is given in Barndorff-Nielsen and Blæsild (1983). Three different possibleparameterisations are described in that paper. A fourth parameterization is given in Prause (1999). All use location and scale parameters μ and δ. There are two other parameters in each case.

Use `hyperbChangePars` to convert from the (α, β) (ϕ, γ) or (ξ, χ) parameterisations to the (π, ζ) parameterisation used above.

`phyperb` breaks the real line into eight regions in order to determine the integral of `dhyperb`. The break points determining the regions are found by `hyperbBreaks`, based on the values of `small`, `tiny`, and `deriv`. In the extreme tails of the distribution where the probability is `tiny` according to `hyperbCalcRange`, the probability is taken to be zero. In the range between where the probability is `tiny` and `small` according to `hyperbCalcRange`, an exponential approximation to the hyperbolic distribution is used. In the inner part of the distribution, the range is divided in 4 regions, 2 above the mode, and 2 below. On each side of the mode, the break point which forms the 2 regions is where the derivative of the density function is `deriv` times the maximum value of the derivative on that side of the mode. In each of the 4 inner regions the numerical integration routine `safeIntegrate` (which is a wrapper for `integrate`) is used to integrate the density `dhyperb`.

`qhyperb` uses the breakup of the real line into the same 8 regions as `phyperb`. For quantiles which fall in the 2 extreme regions, the quantile is returned as `-Inf` or `Inf` as appropriate. In the range between where the probability is `tiny` and `small` according to `hyperbCalcRange`, an exponential approximation to the hyperbolic distribution is used from which the quantile may be found in closed form. In the 4 inner regions `splinefun` is used to fit values of the distribution function generated by `phyperb`. The quantiles are then found using the `uniroot` function.

`phyperb` and `qhyperb` may generally be expected to be accurate to 5 decimal places.

The hyperbolic distribution is a special case of the generalized hyperbolic distribution (Barndorff-Nielsen and Blæsild (1983)). The generalized hyperbolic distribution can be represented as a particular mixture of the normal distribution where the mixing distribution is the generalized inverse Gaussian. `rhyperb` uses this representation to generate observations from the hyperbolic distribution. Generalized inverse Gaussian observations are obtained via the algorithm of Dagpunar (1989).

Value

`dhyperb` gives the density, `phyperb` gives the distribution function, `qhyperb` gives the quantile function and `rhyperb` generates random variates. An estimate of the accuracy of the approximation to the distribution function may be found by setting `accuracy = TRUE` in the call to `phyperb` which then returns a list with components `value` and `error`.

`ddhyperb` gives the derivative of `dhyperb`.

`hyperbBreaks` returns a list with components:

`xTiny`	Value such that probability to the left is less than `tiny`.

xSmall	Value such that probability to the left is less than small.
lowBreak	Point to the left of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode
highBreak	Point to the right of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode
xLarge	Value such that probability to the right is less than small.
xHuge	Value such that probability to the right is less than tiny.
modeDist	The mode of the given hyperbolic distribution.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Ai-Wei Lee, Jennifer Tso, Richard Trendall

References

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

Dagpunar, J.S. (1989). An easily implemented generalized inverse Gaussian generator *Commun. Statist. -Simula.*, **18**, 703–710.

Prause, K. (1999) *The generalized hyperbolic models: Estimation, financial derivatives and risk measurement.* PhD Thesis, Mathematics Faculty, University of Freiburg.

See Also

safeIntegrate, integrate for its shortfalls, splinefun, uniroot and hyperbChangePars for changing parameters to the (π, ζ) parameterisation, dghyp for the generalized hyperbolic distribution.

Examples

```
Theta <- c(2,1,1,0)
hyperbRange <- hyperbCalcRange(Theta, tol = 10^(-3))
par(mfrow = c(1,2))
curve(dhyperb(x, Theta), from = hyperbRange[1], to = hyperbRange[2],
      n = 1000)
title("Density of the\n Hyperbolic Distribution")
curve(phyperb(x, Theta), from = hyperbRange[1], to = hyperbRange[2],
      n = 1000)
title("Distribution Function of the\n Hyperbolic Distribution")
dataVector <- rhyperb(500, Theta)
curve(dhyperb(x, Theta), range(dataVector)[1], range(dataVector)[2],
      n = 500)
hist(dataVector, freq = FALSE, add =TRUE)
title("Density and Histogram\n of the Hyperbolic Distribution")
logHist(dataVector, main =
        "Log-Density and Log-Histogram\n of the Hyperbolic Distribution")
curve(log(dhyperb(x, Theta)), add = TRUE,
      range(dataVector)[1], range(dataVector)[2], n = 500)
```

```
par(mfrow = c(2,1))
curve(dhyperb(x, Theta), from = hyperbRange[1], to = hyperbRange[2],
      n = 1000)
title("Density of the\n Hyperbolic Distribution")
curve(ddhyperb(x, Theta), from = hyperbRange[1], to = hyperbRange[2],
      n = 1000)
title("Derivative of the Density\n of the Hyperbolic Distribution")
par(mfrow = c(1,1))
hyperbRange <- hyperbCalcRange(Theta, tol = 10^(-6))
curve(dhyperb(x, Theta), from = hyperbRange[1], to = hyperbRange[2],
      n = 1000)
bks <- hyperbBreaks(Theta)
abline(v = bks)
```

HyperbolicDistribution

The Package 'HyperbolicDist': Summary Information

Description

This package provides a collection of functions for working with the hyperbolic and related distributions.

For the hyperbolic distribution functions are provided for the density function, distribution function, quantiles, random number generation and fitting the hyperbolic distribution to data (`hyperbFit`). The function `hyperbChangePars` will interchange parameter values between different parameterisations. The mean, variance, skewness, kurtosis and mode of a given hyperbolic distribution are given by `hyperbMean`, `hyperbVar`, `hyperbSkew`, `hyperbKurt`, and `hyperbMode` respectively. For assessing the fit of the hyperbolic distribution to a set of data, the log-histogram is useful. See `logHist`. Q-Q and P-P plots are also provided for assessing the fit of a hyperbolic distribution. A Crämer-von~Mises test of the goodness of fit of data to a hyperbolic distribution is given by `hyperbCvMTest`. S3 `print`, `plot` and `summary` methods are provided for the output of `hyperbFit`.

For the generalized hyperbolic distribution functions are provided for the density function, distribution function, quantiles, and for random number generation. The function `ghypChangePars` will interchange parameter values between different parameterisations. The mean, variance, and mode of a given generalized hyperbolic distribution are given by `ghypMean`, `ghypVar`, `ghypSkew`, `ghypKurt`, and `ghypMode` respectively. Q-Q and P-P plots are also provided for assessing the fit of a generalized hyperbolic distribution.

For the generalized inverse Gaussian distribution functions are provided for the density function, distribution function, quantiles, and for random number generation. The function `gigChangePars` will interchange parameter values between different parameterisations. The mean, variance, skewness, kurtosis and mode of a given generalized inverse Gaussian distribution are given by `gigMean`, `gigVar`, `gigSkew`, `gigKurt`, and `gigMode` respectively. Q-Q and P-P plots are also provided for assessing the fit of a generalized inverse Gaussian distribution.

For the skew-Laplace distribution functions are provided for the density function, distribution function, quantiles, and for random number generation. Q-Q and P-P plots are also provided for assessing the fit of a skew-Laplace distribution.

Acknowledgements

A number of students have worked on the package: Ai-Wei Lee, Jennifer Tso, Richard Trendall, and Thomas Tran.

Thanks to Ross Ihaka and Paul Murrell for their willingness to answer my questions, and to all the core group for the development of R.

LICENCE

Author(s)

David Scott <d.scott@auckland.ac.nz>

References

Barndorff-Nielsen, O. (1977) Exponentially decreasing distributions for the logarithm of particle size, *Proc. Roy. Soc. Lond.*, A**353**, 401–419.

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

Fieller, N. J., Flenley, E. C. and Olbricht, W. (1992) Statistics of particle size data. *Appl. Statist.*, **41**, 127–146.

Jörgensen, B. (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution*. Lecture Notes in Statistics, Vol. 9, Springer-Verlag, New York.

Prause, K. (1999) *The generalized hyperbolic models: Estimation, financial derivatives and risk measurement.* PhD Thesis, Mathematics Faculty, University of Freiburg.

HyperbPlots — *Hyperbolic Quantile-Quantile and Percent-Percent Plots*

Description

`qqhyperb` produces a hyperbolic Q-Q plot of the values in `y`.

`pphyperb` produces a hyperbolic P-P (percent-percent) or probability plot of the values in `y`.

Graphical parameters may be given as arguments to `qqhyperb`, and `pphyperb`.

Usage

```
qqhyperb(y, Theta, main = "Hyperbolic Q-Q Plot",
  xlab = "Theoretical Quantiles",
  ylab = "Sample Quantiles",
  plot.it = TRUE, line = TRUE, ...)

pphyperb(y, Theta, main = "Hyperbolic P-P Plot",
  xlab = "Uniform Quantiles",
  ylab = "Probability-integral-transformed Data",
  plot.it = TRUE, line = TRUE, ...)
```

Arguments

`y`	The data sample.
`Theta`	Parameters of the hyperbolic distribution.
`xlab, ylab, main`	Plot labels.
`plot.it`	Logical. Should the result be plotted?
`line`	Add line through origin with unit slope.
`...`	Further graphical parameters.

Value

For `qqhyperb` and `pphyperb`, a list with components:

`x`	The x coordinates of the points that are to be plotted.
`y`	The y coordinates of the points that are to be plotted.

References

Wilk, M. B. and Gnanadesikan, R. (1968) Probability plotting methods for the analysis of data. *Biometrika*. **55**, 1–17.

See Also

`ppoints`, `dhyperb`, `hyperbFit`

Examples

```
par(mfrow = c(1,2))
y <- rhyperb(200, c(2,2,2,2))
qqhyperb(y, c(2,2,2,2),line = FALSE)
abline(0, 1, col = 2)
pphyperb(y, c(2,2,2,2))
```

`hyperbWSqTable` *Percentage Points for the Cramer-von Mises Test of the Hyperbolic Distribution*

Description

This gives Table 5 of Puig & Stephens (2001) which is used for testing the goodness-of-fit of the hyperbolic distribution using the Crämer-von~Mises test. It is for internal use by `hyperbCvMTest` and `hyperbCvMTestPValue` only and is not intended to be accessed by the user. It is loaded automatically when the package `HyperbolicDist` is invoked.

Usage

```
data(hyperbWSqTable)
```

Format

The `hyperbWSqTable` matrix has 55 rows and 5 columns, giving percentage points of W^2 for different values of ξ and α (the rows), and of χ (the columns).

Source

Puig, Pedro and Stephens, Michael A. (2001), Goodness-of-fit tests for the hyperbolic distribution. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, **29**, 309–320.

`is.wholenumber` *Is Object Numeric and Whole Numbers*

Description

Checks whether an object is numeric and if so, are all the elements whole numbers, to a given tolerance.

Usage

```
is.wholenumber(x, tolerance = .Machine$double.eps^0.5)
```

Arguments

`x`	The object to be tested.
`tolerance`	Numeric ≥ 0. Absolute differences greater than `tolerance` are treated as real differences.

Details

The object x is first tested to see if it is numeric. If not the function returns 'FALSE'. Then if all the elements of x are whole numbers to within the tolerance given by tolerance the function returns 'TRUE'. If not it returns 'FALSE'.

Value

Either 'TRUE' or 'FALSE' depending on the result of the test.

Author(s)

David Scott <d.scott@auckland.ac.nz>.

References

Based on a post by Tony Plate <tplate@acm.org> on R-help.

Examples

```
is.wholenumber(-3:5)                           # TRUE
is.wholenumber(c(0,0.1,1.3,5))                 # FALSE
is.wholenumber(-3:5 + .Machine$double.eps)     # TRUE
is.wholenumber(-3:5 + .Machine$double.eps^0.5) # FALSE
is.wholenumber(c(2L,3L))                       # TRUE
is.wholenumber(c("2L","3L"))                   # FALSE
is.wholenumber(0i ^ (-3:3))                    # FALSE
is.wholenumber(matrix(1:6, nrow = 3))          # TRUE
is.wholenumber(list(-1:3,2:6))                 # FALSE
is.numeric(list(-1:3,2:6))                     # FALSE
is.wholenumber(unlist(list(-1:3,2:6)))         # TRUE
```

---

logHist *Plot Log-Histogram*

Description

Plots a log-histogram, as in for example Feiller, Flenley and Olbricht (1992).

The intended use of the log-histogram is to examine the fit of a particular density to a set of data, as an alternative to a histogram with a density curve. For this reason, only the log-density histogram is implemented, and it is not possible to obtain a log-frequency histogram.

The log-histogram can be plotted with histogram-like dashed vertical bars, or as points marking the tops of the log-histogram bars, or with both bars and points.

Usage

```
logHist(x, breaks = "Sturges",
        include.lowest = TRUE, right = TRUE,
        main = paste("Log-Histogram of", xName),
        xlim = range(breaks), ylim = NULL, xlab = xName,
        ylab = "Log-density", nclass = NULL, htype = "b", ...)
```

Arguments

`x` A vector of values for which the log-histogram is desired.

`breaks` One of:

- a vector giving the breakpoints between log-histogram cells;
- a single number giving the number of cells for the log-histogram;
- a character string naming an algorithm to compute the number of cells (see **Details**);
- a function to compute the number of cells.

In the last three cases the number is a suggestion only.

`include.lowest` Logical. If `TRUE`, an 'x[i]' equal to the 'breaks' value will be included in the first (or last, for `right = FALSE`) bar.

`right` Logical. If `TRUE`, the log-histograms cells are right-closed (left open) intervals.

`main, xlab, ylab`
These arguments to `title` have useful defaults here.

`xlim` Sensible default for the range of x values.

`ylim` Calculated by `logHist`, see **Details**.

`nclass` Numeric (integer). For compatibility with `hist` only, `nclass` is equivalent to `breaks` for a scalar or character argument.

`htype` Type of histogram. Possible types are:

- '"h"' for a *h*istogram only;
- '"p"' for *p*oints marking the top of the histogram bars only;
- '"b"' for *b*oth.

`...` Further graphical parameters for calls to `plot` and `points`.

Details

Uses `hist.default` to determine the cells or classes and calculate counts.

To calculate `ylim` the following procedure is used. The upper end of the range is given by the maximum value of the log-density, plus 25% of the absolute value of the maximum. The lower end of the range is given by the smallest (finite) value of the log-density, less 25% of the difference between the largest and smallest (finite) values of the log-density.

A log-histogram in the form used by Feiller, Flenley and Olbricht (1992) is plotted. See also Barndorff-Nielsen (1977) for use of log-histograms.

Value

Returns a list with components:

- `breaks` The $n + 1$ cell boundaries (= `breaks` if that was a vector).
- `counts` n integers; for each cell, the number of `x[]` inside.
- `logDensity` Log of $\hat{f}(x_i)$, which are estimated density values.
  If `all(diff(breaks) == 1)`, estimated density values are the relative frequencies `counts/n` and in general satisfy $\sum_i \hat{f}(x_i)(b_{i+1} - b_i) = 1$, where b_i = `breaks[i]`.
- `mids` The n cell midpoints.
- `xName` A character string with the actual `x` argument name.
- `heights` The location of the tops of the vertical segments used in drawing the log-histogram.
- `ylim` The value of `ylim` calculated by `logHist`.

Author(s)

David Scott <`d.scott@auckland.ac.nz`>, Richard Trendall, Thomas Tran

References

Barndorff-Nielsen, O. (1977) Exponentially decreasing distributions for the logarithm of particle size, *Proc. Roy. Soc. Lond.*, A**353**, 401–419.

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

Fieller, N. J., Flenley, E. C. and Olbricht, W. (1992) Statistics of particle size data. *Appl. Statist.*, **41**, 127–146.

See Also

`hist`

Examples

```
data(SandP500)
### Consider proportional changes in the index
change <- SandP500[-length(SandP500)]/SandP500[-1]
hist(change)
logHist(change)
### Show points only
logHist(change, htype = "p", pch = 20, cex = 0.5)
### Fit the hyperbolic distribution to the changes
hyperbFit(change)
```

mamquam *Size of Gravels from Mamquam River*

Description

Size of gravels collected from a sandbar in the Mamquam River, British Columbia, Canada. Summary data, giving the frequency of observations in 16 different size classes.

Usage

```
data(mamquam)
```

Format

The mamquam data frame has 16 rows and 2 columns.

[, 1]	midpoints	midpoints of intervals (psi units)
[, 2]	counts	number of observations in interval

Details

Gravel sizes are determined by passing clasts through templates of particular sizes. This gives a range in which the size of each clast lies. Sizes (in mm) are then converted into psi units by taking the base 2 logarithm of the size. The midpoints specified are the midpoints of the psi unit ranges, and counts gives the number of observations in each size range. The classes are of length 0.5 psi units. There are 3574 observations.

Source

Rice, Stephen and Church, Michael (1996) Sampling surficial gravels: the precision of size distribution percentile estimates. *J. of Sedimentary Research*, **66**, 654–665.

Examples

```
data(mamquam)
str(mamquam)
attach(mamquam)
### Construct data from frequency summary, taking all observations
### at midpoints of intervals
psi <- rep(midpoints, counts)
barplot(table(psi))
### Fit the hyperbolic distribution
hyperbFit(psi)

### Actually hyperbFit can deal with frequency data
hyperbFit(midpoints, freq=counts)
```

`momChangeAbout` *Obtain Moments About a New Location*

Description

Using the moments up to a given order about one location, this function either returns the moments up to that given order about a new location as a vector or it returns a moment of a specific order defined by users (order <= maximum order of the given moments) about a new location as a single number. A generalization of using raw moments to obtain a central moment or using central moments to obtain a raw moment.

Usage

```
momChangeAbout(order = "all", oldMom, oldAbout, newAbout)
```

Arguments

`order`	One of: • the character string "all", the default; • a positive integer less than the maximum order of `oldMom`.
`oldMom`	Numeric. Moments of orders 1, 2, ..., about the point `oldAbout`.
`oldAbout`	Numeric. The point about which the moments `oldMom` have been calculated.
`newAbout`	Numeric. The point about which the desired moment or moments are to be obtained.

Details

Suppose m_k denotes the k-th moment of a random variable X about a point a, and m_k^* denotes the k-th moment about b. Then m_k^* may be determined from the moments $m_1, m_2, \ldots, m_k$ according to the formula

$$m_k^* = \sum_{i=0}^{k} (a-b)^i m^{k-i}$$

This is the formula implemented by the function `momChangeAbout`. It is a generalization of the well-known formulae used to change raw moments to central moments or to change central moments to raw moments. See for example Kendall and Stuart (1989), Chapter 3.

Value

The moment of order `order` about the location `newAbout` when `order` is specified. The vector of moments about the location `newAbout` from first order up to the maximum order of the `oldMom` when `order` takes the value `"all"` or is not specified.

Author(s)

David Scott <`d.scott@auckland.ac.nz`>, Christine Yang Dong <`c.dong@auckland.ac.nz`>

References

Kendall, M. G. and Stuart, A. (1969). The Advanced Theory of Statistics, Volume 1, 3rd Edition. London: Charles Griffin & Company.

Examples

```
### Gamma distribution
k <- 4
shape <- 2
old <- 0
new <- 1
sampSize <- 1000000

### Calculate 1st to 4th raw moments
m <- numeric(k)
for (i in 1:k){
   m[i] <- gamma(shape + i)/gamma(shape)
}
m

### Calculate 4th moment about new
momChangeAbout(k, m, old, new)
### Calculate 3rd about new
momChangeAbout(3, m, old, new)

### Calculate 1st to 4th moments about new
momChangeAbout(oldMom = m, oldAbout = old, newAbout = new)
momChangeAbout(order = "all", m, old, new)

### Approximate kth moment about new using sampling
x <- rgamma(sampSize, shape)
mean((x - new)^k)
```

---

momIntegrated *Moments Using Integration*

Description

Calculates moments and absolute moments about a given location for the generalized hyperbolic and related distributions.

Usage

```
momIntegrated(densFn, order, param = NULL, about = 0, absolute = FALSE)
```

Arguments

densFn	Character. The name of the density function whose moments are to be calculated. See **Details**.
order	Numeric. The order of the moment or absolute moment to be calculated.
param	Numeric. A vector giving the parameter values for the distribution specified by densFn. If no param values are specified, then the default parameter values of each distribution are used instead.
about	Numeric. The point about which the moment is to be calculated.
absolute	Logical. Whether absolute moments or ordinary moments are to be calculated. Default is FALSE.

Details

Denote the density function by f. Then if order$= k$ and about$= a$, momIntegrated calculates

$$\int_{-\infty}^{\infty} (x-a)^k f(x)dx$$

when absolute = FALSE and

$$\int_{-\infty}^{\infty} |x-a|^k f(x)dx$$

when absolute = TRUE.

Only certain density functions are permitted.

When densFn="ghyp" or "generalized hyperbolic" the density used is dghyp. The default value for param is c(1,1,0,1,0).

When densFn="hyperb" or "hyperbolic" the density used is dhyperb. The default value for param is c(0,1,1,0).

When densFn="gig" or "generalized inverse gaussian" the density used is dgig. The default value for param is c(1,1,1).

When densFn="gamma" the density used is dgamma. The default value for param is c(1,1).

When densFn="invgamma" or "inverse gamma" the density used is the density of the inverse gamma distribution given by

$$f(x) = \frac{u^\alpha e^{-u}}{x\Gamma(\alpha)}, \quad u = \theta/x$$

for $x > 0$, $\alpha > 0$ and $\theta > 0$. The parameter vector param = c(shape,rate) where shape $= \alpha$ and rate$= 1/\theta$. The default value for param is c(-1,1).

When densFn="vg" or "variance gamma" the density used is dvg from the package **VarianceGamma**. In this case, the package **VarianceGamma** must be loaded or an error will result. The default value for param is c(0,1,0,1).

Value

The value of the integral as specified in **Details**.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

See Also

dghyp, dhyperb, dgamma, dgig, VarianceGamma

Examples

```
### Calculate the mean of a generalized hyperbolic distribution
### Compare the use of integration and the formula for the mean
m1 <- momIntegrated("ghyp", param = c(1/2,3,1,1,0), order = 1, about = 0)
m1
ghypMean(c(1/2,3,1,1,0))
### The first moment about the mean should be zero
momIntegrated("ghyp", order = 1, param = c(1/2,3,1,1,0), about = m1)
### The variance can be calculated from the raw moments
m2 <- momIntegrated("ghyp", order = 2, param = c(1/2,3,1,1,0), about = 0)
m2
m2 - m1^2
### Compare with direct calculation using integration
momIntegrated("ghyp", order = 2, param = c(1/2,3,1,1,0), about = m1)
momIntegrated("generalized hyperbolic", param = c(1/2,3,1,1,0), order = 2,
              about = m1)
### Compare with use of the formula for the variance
ghypVar(c(1/2,3,1,1,0))
```

---

momRecursion *Computes the moment coefficients recursively for generalized hyperbolic and related distributions*

Description

This function computes all of the moments coefficients by recursion based on Scott, WÃ¼rtz and Tran (2008). See **Details** for the formula.

Usage

```
momRecursion(order = 12, printMatrix = FALSE)
```

Arguments

order	Numeric. The order of the moment coefficients to be calculated. Not permitted to be a vector. Must be a positive whole number except for moments about zero.
printMatrix	Logical. Should the coefficients matrix be printed?

Details

The moment coefficients recursively as $a_{1,1} = 1$ and

$$a_{k,\ell} = a_{k-1,\ell-1} + (2\ell - k + 1)a_{k-1,\ell}$$

with $a_{k,\ell} = 0$ for $\ell < \lfloor (k+1)/2 \rfloor$ or $\ell > k$ where k = `order`, ℓ is equal to the integers from $(k+1)/2$ to k.

This formula is given in Scott, WÃ¼rtz and Tran (2008, working paper).

The function also calculates M which is equal to $2\ell - k$. It is a common term which will appear in the formulae for calculating moments of generalized hyperbolic and related distributions.

Value

`a`	The non-zero moment coefficients for the specified order.
`l`	Integers from (`order`+1)/2 to `order`. It is used when computing the moment coefficients and the mu moments.
`M`	The common term used when computing mu moments for generalized hyperbolic and related distributions, M = $2\ell - k$, k=`order`
`lmin`	The minimum of ℓ, which is equal to (`order`+1)/2.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References

Scott, D. J., WÃ¼rtz, D. and Tran, T. T. (2008) Moments of the Generalized Hyperbolic Distribution. Preprint.

Examples

```
momRecursion(order = 12)

#print out the matrix
momRecursion(order = 12, "true")
```

`resistors` *Resistance of One-half-ohm Resistors*

Description

This data set gives the resistance in ohms of 500 nominally one-half-ohm resistors, presented in Hahn and Shapiro (1967). Summary data giving the frequency of observations in 28 intervals.

Usage

```
data(resistors)
```

Format

The `resistors` data frame has 28 rows and 2 columns.

[, 1]	midpoints	midpoints of intervals (ohm)
[, 2]	counts	number of observations in interval

Source

Hahn, Gerald J. and Shapiro, Samuel S. (1967) *Statistical Models in Engineering*. New York: Wiley, page 207.

References

Chen, Hanfeng, and Kamburowska, Grazyna (2001) Fitting data to the Johnson system. *J. Statist. Comput. Simul.*, **70**, 21–32.

Examples

```
data(resistors)
str(resistors)
attach(resistors)
### Construct data from frequency summary, taking all observations
### at midpoints of intervals
resistances <- rep(midpoints,counts)
hist(resistances)
logHist(resistances)
## Fit the hyperbolic distribution
hyperbFit(resistances)

## Actually fit.hyperb can deal with frequency data
hyperbFit(midpoints, freq=counts)
```

`safeIntegrate` *Safe Integration of One-Dimensional Functions*

Description

Adaptive quadrature of functions of one variable over a finite or infinite interval.

Usage

```
safeIntegrate(f, lower, upper, subdivisions=100,
              rel.tol = .Machine$double.eps^0.25, abs.tol = rel.tol,
              stop.on.error = TRUE, keep.xy = FALSE, aux = NULL, ...)
```

Arguments

f	An R function taking a numeric first argument and returning a numeric vector of the same length. Returning a non-finite element will generate an error.
lower, upper	The limits of integration. Can be infinite.
subdivisions	The maximum number of subintervals.
rel.tol	Relative accuracy requested.
abs.tol	Absolute accuracy requested.
stop.on.error	Logical. If true (the default) an error stops the function. If false some errors will give a result with a warning in the `message` component.
keep.xy	Unused. For compatibility with S.
aux	Unused. For compatibility with S.
...	Additional arguments to be passed to `f`. Remember to use argument names *not* matching those of `safeIntegrate(.)`!

Details

This function is just a wrapper around `integrate` to check for equality of `upper` and `lower`. A check is made using `all.equal`. When numerical equality is detected, if `lower` (and hence `upper`) is infinite, the value of the integral and the absolute error are both set to 0. When `lower` is finite, the value of the integral is set to 0, and the absolute error to the average of the function values at `upper` and `lower` times the difference between `upper` and `lower`.

When `upper` and `lower` are determined to be different, the result is exactly as given by `integrate`.

Value

A list of class `"integrate"` with components:

value	The final estimate of the integral.
abs.error	Estimate of the modulus of the absolute error.
subdivisions	The number of subintervals produced in the subdivision process.
message	`"OK"` or a character string giving the error message.
call	The matched call.

See Also

The function `integrate` and `all.equal`.

Examples

```
integrate(dnorm, -1.96, 1.96)
safeIntegrate(dnorm, -1.96, 1.96)  # Same as for integrate()
integrate(dnorm, -Inf, Inf)
safeIntegrate(dnorm, -Inf, Inf)    # Same as for integrate()
integrate(dnorm, 1.96, 1.96)       # OK here but can give an error
safeIntegrate(dnorm, 1.96, 1.96)
integrate(dnorm, -Inf, -Inf)
```

```
safeIntegrate(dnorm, -Inf, -Inf)   # Avoids nonsense answer
integrate(dnorm, Inf, Inf)
safeIntegrate(dnorm, Inf, Inf)     # Avoids nonsense answer
```

`Sample Moments` *Sample Skewness and Kurtosis*

Description

Computes the sample skewness and sample kurtosis.

Usage

```
skewness(x, na.rm = FALSE)
kurtosis(x, na.rm = FALSE)
```

Arguments

`x`	A numeric vector containing the values whose skewness or kurtosis is to be computed.
`na.rm`	A logical value indicating whether `NA` values should be stripped before the computation proceeds.

Details

If $N = \mathrm{length}(x)$, then the skewness of x is defined as

$$N^{-1}\mathrm{sd}(x)^{-3}\sum_i (x_i - \mathrm{mean}(x))^3.$$

If $N = \mathrm{length}(x)$, then the kurtosis of x is defined as

$$N^{-1}\mathrm{sd}(x)^{-4}\sum_i (x_i - \mathrm{mean}(x))^4 - 3.$$

Value

The skewness or kurtosis of `x`.

Note

These functions and the description of them are taken from the package `e1071`. They are included to avoid having to require an additional package.

Author(s)

Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer, and Andreas Weingessel

Examples

```
x <- rnorm(100)
skewness(x)
kurtosis(x)
```

SandP500 *S\&P 500*

Description

This data set gives the value of Standard and Poor's most notable stock market price index (the S\&P 500) at year end, from 1800 to 2001.

Usage

```
data(SandP500)
```

Format

A vector of 202 observations.

Source

http://www.globalfindata.com

References

Brown, Barry W., Spears, Floyd M. and Levy, Lawrence B. (2002) The log F: a distribution for all seasons. *Computational Statistics*, **17**, 47–58.

Examples

```
data(SandP500)
### Consider proportional changes in the index
change<-SandP500[-length(SandP500)]/SandP500[-1]
hist(change)
### Fit hyperbolic distribution to changes
hyperbFit(change)
```

SkewLaplace *Skew-Laplace Distribution*

Description

Density function, distribution function, quantiles and random number generation for the skew-Laplace distribution.

Usage

```
dskewlap(x, Theta, logPars = FALSE)
pskewlap(q, Theta)
qskewlap(p, Theta)
rskewlap(n, Theta)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations to be generated.
Theta	Vector of parameters of the skew-Laplace distribution: α, β and μ.
logPars	Logical. If TRUE the first and second components of Theta are taken to be log(alpha) and log(beta) respectively.

Details

The central skew-Laplace has mode zero, and is a mixture of a (negative) exponential distribution with mean β, and the negative of an exponential distribution with mean α. The weights of the positive and negative components are proportional to their means.

The general skew-Laplace distribution is a shifted central skew-Laplace distribution, where the mode is given by μ.

The density is given by:

$$f(x) = \frac{1}{\alpha + \beta} e^{(x-\mu)/\alpha}$$

for $x \leq \mu$, and

$$f(x) = \frac{1}{\alpha + \beta} e^{-(x-\mu)/\beta}$$

for $x \geq \mu$

Value

`dskewlap` gives the density, `pskewlap` gives the distribution function, `qskewlap` gives the quantile function and `rskewlap` generates random variates. The distribution function is obtained by elementary integration of the density function. Random variates are generated from exponential observations using the characterization of the skew-Laplace as a mixture of exponential observations.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Ai-Wei Lee, Richard Trendall

References

Fieller, N. J., Flenley, E. C. and Olbricht, W. (1992) Statistics of particle size data. *Appl. Statist.*, **41**, 127–146.

See Also

hyperbFitStart

Examples

```
Theta <- c(1,2,1)
par(mfrow = c(1,2))
curve(dskewlap(x, Theta), from = -5, to = 8, n = 1000)
title("Density of the\n Skew-Laplace Distribution")
curve(pskewlap(x, Theta), from = -5, to = 8, n = 1000)
title("Distribution Function of the\n Skew-Laplace Distribution")
dataVector <- rskewlap(500, Theta)
curve(dskewlap(x, Theta), range(dataVector)[1], range(dataVector)[2],
      n = 500)
hist(dataVector, freq = FALSE, add =TRUE)
title("Density and Histogram\n of the Skew-Laplace Distribution")
logHist(dataVector, main =
        "Log-Density and Log-Histogram\n of the Skew-Laplace Distribution")
curve(log(dskewlap(x, Theta)), add = TRUE,
      range(dataVector)[1], range(dataVector)[2], n = 500)
```

SkewLaplacePlots — *Skew-Laplace Quantile-Quantile and Percent-Percent Plots*

Description

qqskewlap produces a skew-Laplace QQ plot of the values in y.

ppskewlap produces a skew-Laplace PP (percent-percent) or probability plot of the values in y.

If line = TRUE, a line with zero intercept and unit slope is added to the plot.

Graphical parameters may be given as arguments to qqskewlap, and ppskewlap.

Usage

```
qqskewlap(y, Theta, main = "Skew-Laplace Q-Q Plot",
  xlab = "Theoretical Quantiles",
  ylab = "Sample Quantiles",
  plot.it = TRUE, line = TRUE, ...)
```

```
ppskewlap(y, Theta, main = "Skew-Laplace P-P Plot",
  xlab = "Uniform Quantiles",
  ylab = "Probability-integral-transformed Data",
  plot.it = TRUE, line = TRUE, ...)
```

Arguments

`y` The data sample.

`Theta` Parameters of the skew-Laplace distribution.

`xlab, ylab, main` Plot labels.

`plot.it` Logical. TRUE denotes the results should be plotted.

`line` Logical. If TRUE, a line with zero intercept and unit slope is added to the plot.

`...` Further graphical parameters.

Value

For `qqskewlap` and `ppskewlap`, a list with components:

`x` The x coordinates of the points that are be plotted.

`y` The y coordinates of the points that are be plotted.

References

Wilk, M. B. and Gnanadesikan, R. (1968) Probability plotting methods for the analysis of data. *Biometrika*. **55**, 1–17.

See Also

`ppoints`, `dskewlap`.

Examples

```
par(mfrow=c(1,2))
y <- rskewlap(1000,c(0.5,1,2))
qqskewlap(y,c(0.5,1,2),line=FALSE)
abline(0,1,col=2)
ppskewlap(y,c(0.5,1,2))
```

Specific Generalized Hyperbolic Moments and Mode
Moments and Mode of the Generalized Hyperbolic Distribution

Description

Functions to calculate the mean, variance, skewness, kurtosis and mode of a specific generalized hyperbolic distribution.

Usage

```
ghypMean(Theta)
ghypVar(Theta)
ghypSkew(Theta)
ghypKurt(Theta)
ghypMode(Theta)
```

Arguments

Theta	Parameter vector of the generalized hyperbolic distribution.

Value

ghypMean gives the mean of the generalized hyperbolic distribution, ghypVar the variance, ghypSkew the skewness, ghypKurt the kurtosis, and ghypMode the mode. The formulae used for the mean is given in Prause (1999). The variance, skewness and kurtosis are obtained using the recursive formula implemented in ghypMom which can calculate moments of all orders about any point.

The mode is found by a numerical optimisation using optim. For the special case of the hyperbolic distribution a formula for the mode is available, see hyperbMode.

The parameterization of the generalized hyperbolic distribution used for these functions is the (α, β) one. See ghypChangePars to transfer between parameterizations.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Thomas Tran

References

Prause, K. (1999) *The generalized hyperbolic models: Estimation, financial derivatives and risk measurement.* PhD Thesis, Mathematics Faculty, University of Freiburg.

See Also

dghyp, ghypChangePars, besselK, RLambda.

Examples

```
Theta <- c(2,2,1,2,2)
ghypMean(Theta)
ghypVar(Theta)
ghypSkew(Theta)
ghypKurt(Theta)
ghypMode(Theta)
maxDens <- dghyp(ghypMode(Theta), Theta)
ghypRange <- ghypCalcRange(Theta, tol = 10^(-3)*maxDens)
curve(dghyp(x, Theta), ghypRange[1], ghypRange[2])
abline(v = ghypMode(Theta), col = "blue")
abline(v = ghypMean(Theta), col = "red")
```

Specific Generalized Inverse Gaussian Moments and Mode

Moments and Mode of the Generalized Inverse Gaussian Distribution

---

Description

Functions to calculate the mean, variance, skewness, kurtosis and mode of a specific generalized inverse Gaussian distribution.

Usage

```
gigMean(Theta)
gigVar(Theta)
gigSkew(Theta)
gigKurt(Theta)
gigMode(Theta)
```

Arguments

`Theta`	Parameter vector of the generalized inverse Gaussian distribution.

Value

`gigMean` gives the mean of the generalized inverse Gaussian distribution, `gigVar` the variance, `gigSkew` the skewness, `gigKurt` the kurtosis, and `gigMode` the mode. The formulae used are as given in Jorgensen (1982), pp. 13–17. Note that the kurtosis is the standardised fourth cumulant or what is sometimes called the kurtosis excess. (See `http://mathworld.wolfram.com/Kurtosis.html` for a discussion.)

The parameterization used for the generalized inverse Gaussian distribution is the (χ, ψ) one (see `dgig`). To use another parameterization, use `gigChangePars`.

Author(s)

David Scott <d.scott@auckland.ac.nz>

References

Jorgensen, B. (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution*. Lecture Notes in Statistics, Vol. 9, Springer-Verlag, New York.

See Also

`dgig`, `gigChangePars`, `besselK`

Examples

```
Theta <- c(-0.5,5,2.5)
gigMean(Theta)
gigVar(Theta)
gigSkew(Theta)
gigKurt(Theta)
gigMode(Theta)
```

`Specific Hyperbolic Distribution Moments and Mode`
Moments and Mode of the Hyperbolic Distribution

Description

Functions to calculate the mean, variance, skewness, kurtosis and mode of a specific hyperbolic distribution.

Usage

```
hyperbMean(Theta)
hyperbVar(Theta)
hyperbSkew(Theta)
hyperbKurt(Theta)
hyperbMode(Theta)
```

Arguments

`Theta`	Parameter vector of the hyperbolic distribution.

Details

The formulae used for the mean, variance and mode are as given in Barndorff-Nielsen and Blæsild (1983), p. 702. The formulae used for the skewness and kurtosis are those of Barndorff-Nielsen and Blæsild (1981), Appendix 2.

Note that the variance, skewness and kurtosis can be obtained from the functions for the generalized hyperbolic distribution as special cases. Likewise other moments can be obtained from the function `ghypMom` which implements a recursive method to moments of any desired order. Note that functions for the generalized hyperbolic distribution use a different parameterization, so care is required.

Value

hyperbMean gives the mean of the hyperbolic distribution, hyperbVar the variance, hyperbSkew the skewness, hyperbKurt the kurtosis and hyperbMode the mode.

Note that the kurtosis is the standardised fourth cumulant or what is sometimes called the kurtosis excess. (See http://mathworld.wolfram.com/Kurtosis.html for a discussion.)

The parameterization of the hyperbolic distribution used for this and other components of the HyperbolicDist package is the (π, ζ) one. See hyperbChangePars to transfer between parameterizations.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Richard Trendall, Thomas Tran

References

Barndorff-Nielsen, O. and Blæsild, P (1981). Hyperbolic distributions and ramifications: contributions to theory and application. In *Statistical Distributions in Scientific Work*, eds., Taillie, C., Patil, G. P., and Baldessari, B. A., Vol. 4, pp. 19–44. Dordrecht: Reidel.

Barndorff-Nielsen, O. and Blæsild, P (1983). Hyperbolic distributions. In *Encyclopedia of Statistical Sciences*, eds., Johnson, N. L., Kotz, S. and Read, C. B., Vol. 3, pp. 700–707. New York: Wiley.

See Also

dhyperb, hyperbChangePars, besselK, ghypMom, ghypMean, ghypVar, ghypSkew, ghypKurt

Examples

```
Theta <- c(2,2,2,2)
hyperbMean(Theta)
hyperbVar(Theta)
hyperbSkew(Theta)
hyperbKurt(Theta)
hyperbMode(Theta)
```

---

summary.hyperbFit *Summarizing Hyperbolic Distribution Fit*

Description

summary Method for class "hyperbFit".

Usage

```
## S3 method for class 'hyperbFit'
summary(object, ...)

## S3 method for class 'summary.hyperbFit'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	An object of class `"hyperbFit"`, resulting from a call to `hyperbFit`.
`x`	An object of class `"summary.hyperbFit"`, resulting from a call to `summary.hyperbFit`.
`digits`	The number of significant digits to use when printing.
`...`	Further arguments passed to or from other methods.

Details

`summary.hyperbFit` calculates standard errors for the estimates of π, ζ, δ, and μ of the hyperbolic distribution parameter vector Theta if the Hessian from the call to `optim` or `nlm` is available. Because the parameters in the call to the optimiser are π, $\log(\zeta)$, $\log(\delta)$, and μ, the delta method is used to obtain the standard errors for ζ and δ.

Value

If the Hessian is available, `summary.hyperbFit` computes standard errors for the estimates of π, ζ, δ, and μ, and adds them to `object` as `object$sds`. Otherwise, no calculations are performed and the composition of `object` is unaltered.

`summary.hyperbFit` invisibly returns `x` with class changed to `summary.hyperbFit`.

See `hyperbFit` for the composition of an object of class `hyperbFit`.

`print.summary.hyperbFit` prints a summary in the same format as `print.hyperbFit` when the Hessian is not available from the fit. When the Hessian is available, the standard errors for the parameter estimates are printed in parentheses beneath the parameter estimates, in the manner of `fitdistr` in the package `MASS`.

See Also

`hyperbFit`, `summary`.

Examples

```
### Continuing the  hyperbFit(.) example:
Theta <- c(2,2,2,2)
dataVector <- rhyperb(500, Theta)
fit <- hyperbFit(dataVector, method = "BFGS", hessian = TRUE)
print(fit)
summary(fit)
```

Index