Package 'SkewHyperbolic'

September 30, 2009

Type Package

Version 0.1-1

Date 2009-09-03

Title Functions for working with the skewed hyperbolic Student's t-distribution

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

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

Depends R (>= 2.3.0), HyperbolicDist

Description Functions are provided for the density function, distribution function, quantiles and random number generation for the skew hyperbolic t-distribution. There are also functions that fit the distribution to data. There are functions for the mean, variance, skewness, kurtosis and mode of a given distribution. To assess goodness of fit, there are also functions to generate a Q-Q plot and a P-P plot.

License GPL (>= 2)

LazyLoad yes

R topics documented:

SkewHyperbolic-package *The Package 'SkewHyperbolic': Summary Information*

Description

This package provides a collection of functions for working with the skew hyperbolic Student's t-distribution.

Functions are provided for the density function (`dskewhyp`), distribution function (`pskewhyp`), quantiles (`qskewhyp`) and random number generation (`rskewhyp`). There are also functions that fit the distribution to data (`skewhypFit`). The mean, variance, skewness, kurtosis and mode can be found usig the functions `skewhypMean`, `skewhypVar`, `skewhypSkew`, `skewhypKurt` and `skewhypMode` respectively. To assess goodness of fit, there are also functions to generate a Q-Q plot (`qqskewhyp`) and a P-P plot (`ppskewhyp`). S3 methods `print`, `plot` and `summary` are provided for the output of `skwewhypFit`.

Package:	SkewHyperbolic
Type:	Package
Version:	0.1-1
Date:	2009-08-28
License:	GPL(>=2)
LazyLoad:	yes

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

See Also

`dskewhyp`, `skewhypMean`, `skewhypFit`, `skewhypFitStart`, `skewhypBreaks`, `qqskewhyp`, `HyperbolicDist`.

`lrdji` *Dow Jones Log Return Data*

Description

Log returns of daily closing value data from the dow jones index, from 04/JAN/1999 to 08/JUL/2003. The original data used to calculate these was the dji data set available in the QRMlib package.

Usage

```
data(lrdji)
```

Format

A vector of 1132 observations.

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

Source

library(QRMlib) data(dji)

References

McNeil, A. & Ulman, S. (2008). QRMlib *http://cran.r-project.org/web/packages/QRMlib/index.html*

Examples

```
data(lrdji)
##fit a skew hyperbolic students t-distribution to the data
fit<-skewhypFit(lrdji, plot=TRUE, print=TRUE)
```

---

lrnokeur *Log Returns of the NOK/EUR Exchange Rate*

Description

Log returns of daily closing value data of the NOK/EUR (Norwegian Kroner/Euro) exchange rate, from 04/JAN/1999 to 08/JUL/2003. The original data was downloaded from the oanda website. The data was selected to be as similar as possible to the data used in the Aas & Haff article (see References).

Usage

```
data(lrnokeur)
```

Format

A vector of 1647 observations.

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

Source

http://www.oanda.com

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

Examples

```
##Fit the skew hyperbolic students-t distribution to the data
data(lrnokeur)
fit <- skewhypFit(lrnokeur, method="nlm", plot=TRUE, print=TRUE)
```

`qqskewhyp` *Skew Hyperbolic Student's t-Distribution Quantile-Quantile and Percent-Percent Plots*

Description

`qqskewhyp` produces a skew hyperbolic t-distribution Q-Q plot of the values in `y`, `ppskewhyp` produces a skew hyperbolic t-distribution P-P (percent-percent) plot or probability plot of the values in `y`. Graphical parameters may be given as arguments to `qqskewhyp` and `ppskewhyp`.

Usage

```
qqskewhyp(y, mu = 0, delta = 1, beta = 1, nu = 1, param = c(mu, delta,
beta, nu), main = "Skew Hyperbolic Student's-t QQ Plot", xlab =
"Theoretical Quantiles", ylab = "Sample Quantiles", plot.it = TRUE, line
= TRUE, ...)
ppskewhyp(y, beta = NULL, delta = NULL, mu = NULL, nu = NULL, param =
c(mu, delta, beta, nu), main = "Skew Hyperbolic Student's-t P-P Plot",
xlab = "Uniform Quantiles", ylab =
"Probability-integral-transformed Data", plot.it = TRUE, line = TRUE, ...)
```

Arguments

`y`	The sample data.
`mu`	Location parameter μ, default is 0.
`delta`	Peakedness parameter δ, default is 1.
`beta`	Skewness parameter β, default is 1.
`nu`	Shape parameter ν, default is 1.
`param`	Specifying the parameters as a vector of the form `c(mu, delta, beta, nu)`.
`main,xlab,ylab`	Plot labels.
`plot.it`	Logical; if `plot.it = TRUE` the results will be plotted.
`line`	Logical; if `line = TRUE` a line is added through the origin with unit slope.
`...`	Further graphical parameters.

Details

Users may either specify the values of the parameters individually or as a vector. If both forms are specified, then the values specified by the vector `param` will overwrite the other ones.

Value

For `qqskewhyp` and `ppskewhyp`, a list with components:

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

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

See Also

`ppoints`, `qqplot`, `dskewhyp`

Examples

```
par(mfrow = c(1,2))
param <- c(0,1,20,10)
y <- rskewhyp(500, param=param)
qqskewhyp(y, param=param, main="Skew Hyperbolic\n Q-Q Plot")
ppskewhyp(y, param=param, main="Skew Hyperbolic\n P-P Plot")
```

`skewhypBreaks` *Break points for the Skew Hyperbolic Student's t-Distribuiton*

Description

Utility routines that calculate suitable breakpoints for use in determining the distribution function, and the derivative of the density function.

Usage

```
skewhypBreaks(mu = 0, delta = 1, beta = 1, nu = 1, param =
  c(mu, delta, beta, nu), small = 10^(-6), tiny = 10^(-10),
  deriv = 0.3, ...)
skewhypCalcRange(mu = 0, delta = 1, beta = 1, nu = 1,
  param = c(mu,delta,beta,nu), tol= 10^(-5), ...)
ddskewhyp(x, mu = 0, delta = 1, beta = 1, nu = 1, param =
  c(mu,delta,beta,nu),log = FALSE, tolerance =
  .Machine$double.eps ^ 0.5)
```

Arguments

x	Vector of quantiles
mu	Location parameter μ, default is 0.
delta	Peakedness parameter δ, default is 1.
beta	Skewness parameter β, default is 1.
nu	Shape parameter ν, default is 1.
param	Specifying the parameters as a vector of the form `c(mu, delta, beta, nu)`.
small	Size of a small difference between the distribution function and 0 or 1.
tiny	Size of a tiny difference between the distribution function and 0 or 1.
deriv	Value between 0 and 1 which determines the point at which the value of the derivative becomes substantial compared to its maximal value, see **Details**.
log	Logical; if `log = TRUE`, probabilities are given as log(p).
tolerance	Specified level of tolerance when checking if parameter beta is equal to 0 in `ddskewhyp`.
tol	Tolerance used in `skewhypCalcRange`, see **Details**.
...	Passes additional arguments to `uniroot`.

Value

`ddskewhyp` gives the derivative of `dskewhyp`.

`skewhypBreaks` returns a list with components:

xTiny	Value such that the probabilities to the left are less than `tiny`.
xSmall	Value such that probabilities to the left are less than `small`.
lowBreak	Point to the left of the mode such that the derivative of the density at that point 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 at that point is `deriv` times the value of the maximum value on that side of the mode.
xLarge	Value such that the probabities to the right are less than `xSmall`.
xHuge	Value such that probability to the right is less than `tiny`.
modeDist	The mode of the given skewed hyperbolic distribution, calculated by `skewhypMode`.

`skewhypCalcRange` returns the quantile values at the endpoints of the region of the density function where the probablity is greater than `tol`.

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

See Also

`uniroot`, `dskewhyp`, `skewhypMode`

Examples

```
param <- c(0,1,10,10)
range <- skewhypCalcRange(param=param, tol=10^(-3))

#plots of density and derivative
par(mfrow=c(2,1))
curve(dskewhyp(x, param=param), range[1], range[2], n=1000)
title("Density of the Skew\n Hyperbolic Distribution")
curve(ddskewhyp(x, param=param), range[1], range[2], n=1000)
title("Derivative of the Density\n of the Skew Hyperbolic Distribution")

#plot of the density marking the break points
par(mfrow=c(1,1))
range <- skewhypCalcRange(param=param, tol=10^(-6))
curve(dskewhyp(x, param=param), range[1], range[2], n=1000)
title("Density of the Skew Hyperbolic Distribution\n with Breakpoints")
breaks <- skewhypBreaks(param=param)
abline(v=breaks)
```

SkewHyperbolicDistribution

Skewed Hyperbolic Student's t-Distribution

Description

Density function, distribution function, quantiles and random number generation for the Generalised Hyperbolic Skew Student's t-Distribution, with parameters β (skewness), δ (peakedness), μ (location) and ν (shape).

Usage

```
dskewhyp(x, mu = 0, delta = 1, beta = 1, nu = 1, param = c(mu, delta,
  beta, nu), log = FALSE, tolerance = .Machine$double.eps^0.5)
pskewhyp(q, mu = 0, delta = 1, beta = 1, nu = 1,param =
  c(mu,delta,beta,nu), log = FALSE, lower.tail = TRUE, small = 10^(-6),
  tiny = 10^(-10), subdivisions = 100, accuracy = FALSE, ...)
qskewhyp(p, mu = 0, delta = 1, beta = 1, nu = 1, param =
  c(mu,delta,beta,nu), small = 10^(-6), tiny = 10^(-10), deriv = 0.3,
  nInterpol = 100, subdivisions = 100, ...)
rskewhyp(n, mu = 0, delta = 1, beta = 1, nu = 1, param =
  c(mu,delta,beta,nu), log = FALSE)
```

Arguments

x,q	Vector of quantiles.
p	Vector of probabilities.
n	Number of random variates to be generated.
mu	Location parameter μ, default is 0.
delta	Peakedness parameter δ, default is 1.
beta	Skewness parameter β, default is 1.

nu	Shape parameter ν, default is 1.
param	Specifying the parameters as a vector of the form c(mu, delta, beta, nu).
log	Logical; if log = TRUE, probabilities are given as log(p).
lower.tail	Logical; if lower.tail = TRUE, the cumulative density is taken from the lower tail.
tolerance	Specified level of tolerance when checking if parameter beta is equal to 0.
small	Size of a small difference between the distribution function and 0 or 1.
tiny	Size of a tiny difference between the distribution function and 0 or 1.
subdivisions	The maximum number of subdivisions used to integrate the density and determine the accuracy of the distribution function calculation.
accuracy	Logical; if accuracy = TRUE, accuracy calculated by integrate to try and determine the accuracy of the distribution function calculation.
deriv	Value between 0 and 1 which determines the point at which the value of the derivative becomes substantial compared to its maximal value, see **Details**.
nInterpol	Number of points used in qskewhyp for cubic spline interpolation of the distribution function.
...	Passes additional arguments to integrate.

Details

Users may either specify the values of the parameters individually or as a vector. If both forms are specified, then the values specified by the vector param will overwrite the other ones.

The density function is

$$f(x) = \frac{2^{(\frac{1-\nu}{2})}\delta^{\nu}|\beta|^{(\frac{\nu+1}{2})}K_{(\frac{\nu+1}{2})}\sqrt{(\beta^2(\delta^2+(x-\mu)^2))}exp(\beta(x-\mu))}{\Gamma(\frac{\nu}{2})\sqrt{(\pi)}\sqrt{(\delta^2+(x-\mu)^2)^{(\frac{\nu+1}{2})}}}$$

when $\beta \neq 0$, and

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{(\pi)}\delta\Gamma(\frac{\nu}{2})}\left(1+\frac{(x-\mu)^2}{\delta^2}\right)^{\frac{-(\nu+1)}{2}}$$

when $\beta = 0$, where $K_{nu}(.)$ is the modified Bessel function of the third kind with order nu, and Γ is the gamma function.

pskewhyp breaks the real line in to 8 regions in order to determine the integral of dhyperb. The breakpoints determining the regions are calculated by skewhypBreaks, based on the values of small, tiny, and deriv.

The inner area is divided into two regions above and two below the mode. The breakpoint that divides these is calculated to be where the derivative of the density function is deriv times the value of the maximum derivative on that side of the mode. In the extreme tails of the distribution where the probability is tiny the probability is taken to be zero. In the remaining regions the integral of the density is calculated using the numerical routine safeIntegrate (a wrapper for integrate).

qhyperb Used the same breakup of the real line as pskewhyp. For quantiles that fall in the two extreme regions the quantile is returned as Inf or -Inf as appropriate. In the remaining

regions `splinefun` is used to fit values of the distribution function calulated by `pskewhyp`. The quantiles are then found by the `uniroot` function.

Note that when small values of ν are used, for example less than ten, and the density is skewed, there are often quite extreme values generated by `rskewhyp`. These look like outliers, but are caused by the heaviness of the skewed tail.

Value

`dskewhyp` gives the density function, `pskewhyp` gives the distribution function, `qskewhyp` gives the quantile function and `rskewhyp` generates random variates.

An estimate of the accuracy of the approximation to the distribution function can be found by setting `accuracy=TRUE` in the call to `pskewyhp` which returns a list with components `value` and `error`.

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

See Also

`safeIntegrate`, `integrate` for its shortfalls, `skewhypBreaks`, `logHist`. Also `skewhypMean` for information on moments and mode, and `skewhypFit` for fitting to data.

Examples

```
param <- c(0,1,40,10)
par(mfrow=c(1,2))
range <- skewhypCalcRange(param=param, tol=10^(-2))

#curves of density and distribution
curve(dskewhyp(x, param=param), range[1], range[2], n=1000)
title("Density of the \n Skew Hyperbolic Distribution")
curve(pskewhyp, range[1], range[2], n=500, param=param)
title("Distribution Function of the \n Skew Hyperbolic Distribution")

#curves of density and log density
par(mfrow=c(1,2))
data <- rskewhyp(1000, param=param)
curve(dskewhyp(x, param=param), range(data)[1], range(data)[2],
      n=1000, col=2)
hist(data, freq=FALSE, add=TRUE)
title("Density and Histogram of the\n Skew Hyperbolic Distribution")
logHist(data, main="Log-Density and Log-Histogram of\n the Skew
      Hyperbolic Distribution")
curve(dskewhyp(x, param=param, log=TRUE), range(data)[1], range(data)[2],
      n=500, add=TRUE, col=2)
```

skewhypFit *Fit the Skew Hyperbolic Student's t-Distribution to Data*

Description

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

Usage

```
skewhypFit(x, freq = NULL, breaks = NULL, startValues = "LA",
  paramStart = NULL, method = "Nelder-Mead", hessian = TRUE,
  plots = TRUE, printOut = TRUE, controlBFGS = list(maxit = 200),
  controlNM = list(maxit = 1000), maxitNLM = 1500, ...)
## S3 method for class 'skewhypFit':
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(), ...)
## S3 method for class 'skewhypFit':
print(x,digits = max(3, getOption("digits") - 3),...)
## S3 method for class 'skewhypFit':
summary(object,...)
```

Arguments

x	Data Vector for `skewhypFit`. Object of class `"skewhypFit"` for `plot.skewhypFit`, `print.skewhypFit` and `summary.skewhypFit`.
freq	Vector of weights with length equal to length of x.
breaks	Breaks for histogram, defaults to those generated by `hist(x, plot=FALSE, right=FALSE)`. If `startValues = "LA"` then 30 breaks are used by default.
startValues	Code giving the method of determining starting values for finding the maximum likelihood estimates of the parameters.
paramStart	If `startValues = "US"` the user must specify a vector of starting parameter values in the form `c(mu, delta, beta, nu)`.
method	Different optimisation methods to consider, see **Details**.
hessian	Logical; if `hessian = TRUE` the value of the hessian is returned.
plots	Logical; if `plots = TRUE` the histogram, log-histogram, Q-Q and P-P plots are printed.
printOut	Logical; if `printOut = TRUE` results of the fitting are printed.
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.
which	If a subset of 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 asked before plot change, see par(ask = .).
digits	Desired number of digits when the object is printed.
object	Object must be of class "skewhypFit"
...	Passes arguments to optim, nlm, hist,logHist, qqskewhyp and ppskewhyp.

Details

startValues can be either "US"(User-supplied) or "LA" (Linear approximation) If startValues = "US" then a value for paramStart must be supplied. For the details concerning the use of startValues and paramStart see skewhypFitStart.

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 the details of how to pass control information using optim and nlm, see optim and nlm.

Value

skewhypFit returns a list with components:

param	A vector giving the maximum likelihood estimates of the parameters in the form c(mu, delta, beta, nu).
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 optim or nlm for details.
iter	Number of iterations of optimisation routine.
x	The data used to fit the distribution.
xName	Character stirng with the actual x argument name.
paramStart	Starting values of the parameters returned by skewhypFitStart.
svName	Name of the method used to find starting values.
startValues	Acronym of method used to find starting values.
breaks	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 if startValues = "US" or density if startValues = "LA".

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

See Also

optim, nlm, par, hist, logHist, qqskewhyp, ppskewhyp, dskewhyp and skewhypFitStart.

Examples

```
## See how well skewhypFit works
param <- c(0,1,4,10)
data <- rskewhyp(500, param=param)
fit <- skewhypFit(data)
## Use data set NOK/EUR as per Aas&Haff
data(lrnokeur)
nkfit <- skewhypFit(lrnokeur, method = "nlm")
## Use data set DJI
data(lrdji)
djfit <- skewhypFit(lrdji)
```

---

skewhypFitStart *Find Starting Values for Fittting a Skew Hyperbolic Student's t-Distribution*

Description

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

Usage

```
skewhypFitStart(x, breaks = NULL, startValues = "LA", paramStart = NULL)
skewhypFitStartLA(x, breaks = NULL)
```

Arguments

x	Data vector.
breaks	Breaks for histogram. If missing defaults to those generated by hist(x, right = FALSE, plot =FALSE). If startValues = "LA" then 30 breaks are used by default.
startValues	Code giving the method of determining starting values for finding the maximum likelihood estimates of the parameters.
paramStart	If startValues = "US" the user must specify a vector of starting parameter values in the form c(mu, delta, beta, nu).

Details

startValues can be either "US"(User-supplied) or "LA" (Linear approximation).

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

If startValues = "LA" a linear approximation is made to the log-density in each of the tails, from which the estimates for ν and β are found. The remaining two parameters, δ and μ are found by solving the moment equations for mean and variance. Since the variance does not exist for values of $\nu \leq 4$, the estimate of ν will be at least 4.1. Note that if the distribution is too skewed, there are not enough points in the lighter tail to fit the required linear model, and the method will stop and return a warning. User supplied values will have to be used in this case.

Value

skewhypFitStart returns a list with components:

paramStart	A vector of the form c(mu, delta, beta, nu) giving the generated starting values of the parameters.
breaks	The cell boundaries found by a call to hist.
midpoints	The cell midpoints found by a call to hist.
empDens	The estimated density at the midpoints found by a call to hist if startValues = "US" or density if startValues = "LA".
svName	Name of the method used to find the starting values.

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

See Also

hist, density, dskewhyp, skewhypFit

Examples

```
#find starting values to feed to skewhypFit
data(lrnokeur)
skewhypFitStart(lrnokeur, startValues="LA")$paramStart
#user supplied values
skewhypFitStart(lrnokeur, startValues="US",
paramStart=c(0,0.01,0,5))$paramStart
```

---

skewhypMeanVarMode *Moments and Mode of the Skew Hyperbolic Student's t-Distribution.*

Description

Functions to calculate the mean, variance, skewness, kurtosis and mode of a specified skew hyperbolic t-distribution.

Usage

```
skewhypMean(mu = 0, delta = 1, beta = 1, nu = 1, param = c(mu, delta,
  beta, nu))
skewhypVar(mu = 0, delta = 1, beta = 1, nu = 1, param = c(mu, delta,
  beta, nu))
skewhypSkew(mu = 0, delta = 1, beta = 1, nu = 1, param = c(mu, delta,
  beta, nu))
skewhypKurt(mu = 0, delta = 1, beta = 1, nu = 1, param = c(mu, delta,
  beta, nu))
skewhypMode(mu = 0, delta = 1, beta = 1, nu = 1, param = c(mu, delta,
  beta, nu), tolerance = .Machine$double.eps ^ 0.5)
```

Arguments

mu	Location parameter μ, default is 0.
delta	Peakedness parameter σ, default is 1.
beta	Skewness parameter β, default is 1. Negative values give a negative skew, positive values give a positive skew.
nu	Shape parameter ν, default is 1.
param	Specifying the parameters as a vector of the form `c(mu, delta, beta, nu)`.
tolerance	A difference smaller than this value is taken to be zero.

Details

Users may either specify the values of the parameters individually or as a vector. If both forms are specified, then the values specified by the vector `param` will overwrite the other ones.

The moments are calculated as per formulae in Aas&Haff(2006) and the mode is calculated by numerical optimisation of the density function using `optim`.

Note that the mean does not exist when $\nu = 2$, the variance does not exist for $\nu \leq 4$, the skewness does not exist for $\nu \leq 6$, and the kurtosis does not exist for $\nu \leq 8$.

Value

`skewhypMean` gives the mean of the skew hyperbolic t-distribution, `skewhypVar` the variance, `skewhypSkew` the skewness, `skewhypKurt` the kurtosis and `skewhypMode` the mode.

Author(s)

David Scott ⟨d.scott@auckland.ac.nz⟩, Fiona Grimson

References

Aas, K. and Haff, I. H. (2006). The Generalised Hyperbolic Skew Student's *t*-distribution, *Journal of Financial Econometrics*, **4**, 275–309.

See Also

`dskewhyp`, `optim`

Examples

```
param <- c(10,1,5,9)
skewhypMean(param=param)
skewhypVar(param=param)
skewhypSkew(param=param)
skewhypKurt(param=param)
skewhypMode(param=param)
range <- skewhypCalcRange(param=param)
curve(dskewhyp(x, param=param), range[1], range[2])
abline(v=skewhypMode(param=param), col="red")
abline(v=skewhypMean(param=param), col="blue")
```

Index