Package ‘SkewHyperbolic’

November 26, 2023

Type Package

Version 0.4-2

Date 2023-11-26

Title The Skew Hyperbolic Student t-Distribution

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

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

Depends R (>= 3.0.1)

Imports grDevices, graphics, stats, DistributionUtils, GeneralizedHyperbolic

Suggests RUnit

Encoding UTF-8

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 and to calculate moments of any order about any centre. To assess goodness of fit, there are functions to generate a Q-Q plot, a P-P plot and a tail plot.

License GPL (>= 2)

URL https://r-forge.r-project.org/projects/rmetrics/

NeedsCompilation no

Repository CRAN

Date/Publication 2023-11-26 00:50:02 UTC

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 *t*-distribution.

Functions are provided for the density function (`dskewhyp`), distribution function (`pskewhyp`), quantiles (`qskewhyp`) and random number generation (`rskewhyp`). There are functions that fit the distribution to data (`skewhypFit`). The mean, variance, skewness, kurtosis and mode can be found using the functions `skewhypMean`, `skewhypVar`, `skewhypSkew`, `skewhypKurt` and `skewhypMode` respectively, and there is also a function to calculate moments of any order `skewhypMom`. To assess goodness of fit, there are 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`.

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.

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

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

See Also

`dskewhyp`, `skewhypMean`,`skewhypMom`, `skewhypFit`, `skewhypFitStart`, `qqskewhyp`, `GeneralizedHyperbolicDistributi` `ais` *Australian Institute of Sport data*

Description

Data on 102 male and 100 female athletes collected at the Australian Institute of Sport, courtesy of Richard Telford and Ross Cunningham.

Usage

```
data(ais)
```

Format

A data frame with 202 observations on 13 variables.

[, 1]	sex	sex
[, 2]	sport	sport
[, 3]	rcc	red cell count
[, 4]	wcc	white cell count
[, 5]	Hc	Hematocrit
[, 6]	Hg	Hemoglobin
[, 7]	Fe	plasma ferritin concentration
[, 8]	bmi	body mass index, weight/(height)^2
[, 9]	ssf	sum of skin folds
[,10]	Bfat	body fat percentage
[,11]	lbm	lean body mass
[,12]	Ht	height (cm)
[,13]	Wt	weight (Kg)

Source

Cook and Weisberg (1994) via the package **sn**. This help file is a modification of the help file from the package **sn**.

References

Cook and Weisberg (1994), *An Introduction to Regression Graphics*. John Wiley & Sons, New York.

Examples

```
data(ais)
Fe <- ais$Fe
### Not enough data to find starting values
### Use default parameter values as starting values
FeFit <- skewhypFit(Fe, startValues = "US", paramStart = c(0,1,1,1))
```

```
### Ferritin must always be non-negative
### Probability of negative values is small for fitted distribution
pskewhyp(0, param = FeFit$param)
```

---

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 student 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)
```

skewhypCalcRange *Range of a Skew Hyperbolic Student t-Distribution*

Description

Given the parameter vector param, or parameter values of a skew hyperbolic Student t-distribution, this function determines the range outside of which the density function or distribution function are negligible, to a specified tolerance.

Usage

```
skewhypCalcRange(mu = 0, delta = 1, beta = 1, nu = 1,
                 param = c(mu,delta,beta,nu), density = TRUE,
                 tol= 10^(-5), ...)
skewhypStepSize(dist, delta, beta, nu, side = c("right","left"))
```

Arguments

mu	Location parameter μ, default is 0.
delta	Scale 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).
density	Logical. If TRUE bounds refer to the density function, otherwise bounds refer to the distribution function.
tol	Density function value at the endpoints of the range returned by the function.
dist	Numeric. Current distance value.
side	Character. "right" for a step to the right, "left" for a step to the right.
...	Passes additional arguments to uniroot.

Details

The particular skew hyperbolic distribution being considered is specified by either the individual parameter values, or the parameter vector param. If both are specified, the values in param will overwriete the other ones. In addition the parameter values are examined by calling the function skewhypCheckPars to see if they are valid.

The function skewhypCalcRange returns the range outside of which the density function or distribution function are less than the given tolerance. The points are found by using uniroot on the density or distribution function.

The function skewhypStepSize is used for stepping to the right or the left to obtain an enclosing interval so uniroot can be used to search. When the tail is declining exponentially the step is just a linear function of the current distance from the mode. If the tail is declining only as a power of x, an exponential step is used.

skewhypStepSize is for internal use and is not expected to be called by users. It is documented here for completeness.

Value

The function skewhypCalcRange returns a two component vector giving the lower and upper limits of the range.

skewhypStepSize returns the size of the step.

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, skewhypCheckPars

Examples

```
param <- c(0,1,10,10)
range <- skewhypCalcRange(param = param, tol = 10^(-2))
range
curve(dskewhyp(x, param = c(0,1,5,10), range[1], range[2]))

param <- c(0,1,20,1)
(range <- skewhypCalcRange(param = param))
round(integrate(dskewhyp, -Inf, range[1], param = param)$value,7)
round(integrate(dskewhyp, range[2], Inf, param = param)$value,7)
```

---

skewhypCheckPars *Check Parameters of the Skew Hyperbolic Student t-distribution*

Description

Given a set of parameters for the skew hyperbolic Student *t*-distribution, the function checks that the parameters are in the correct range, and that the set has the correct length of 4.

Usage

```
skewhypCheckPars(param)
```

Arguments

param	A numeric vector of proposed parameters for the skew hyperbolic *t*-distribution.

Details

The vector `param` should be of the form `c(mu,delta,beta,nu)`. If either `delta` or `nu` is not greater than zero an error message is returned. If the vector `param` does not have a length of 4 then an error message is returned.

Value

A list with components:

`case`	Either `'error'` or `'normal'`, as identified by the function.
`errMessage`	An appropriate error message if an error was found, otherwise an empty string.

Author(s)

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

See Also

`dskewhyp`

Examples

```
skewhypCheckPars(c(0,1,1,1))     #normal
skewhypCheckPars(c(0,0,1,1))     #error
skewhypCheckPars(c(0,1,1,-1))    #error
skewhypCheckPars(c(0,1,1))       #error
```

`SkewHyperbolicDistribution`
Skewed Hyperbolic Student t-Distribution

Description

Density function, distribution function, quantiles and random number generation for the skew hyperbolic Student *t*-distribution, with parameters β (skewness), δ (scale), μ (location) and ν (shape). Also a function for the derivative of the density function.

Usage

```
dskewhyp(x, mu = 0, delta = 1, beta = 1, nu = 1,
         param = c(mu,delta,beta,nu), log = FALSE,
         tolerance = .Machine$double.eps^0.5)
pskewhyp(q, mu = 0, delta = 1, beta = 1, nu = 1,
         param = c(mu, delta, beta, nu), log.p = FALSE,
         lower.tail = TRUE, subdivisions = 100,
         intTol = .Machine$double.eps^0.25, valueOnly = TRUE, ...)
qskewhyp(p, mu = 0, delta = 1, beta = 1, nu = 1,
         param = c(mu,delta, beta, nu),
```

```
          lower.tail = TRUE, log.p = FALSE,
          method = c("spline","integrate"),
          nInterpol = 501, uniTol = .Machine$double.eps^0.25,
          subdivisions = 100, intTol = uniTol, ...)
rskewhyp(n, mu = 0, delta = 1, beta = 1, nu = 1,
          param = c(mu,delta,beta,nu), log = FALSE)
ddskewhyp(x, mu = 0, delta = 1, beta = 1, nu = 1,
           param = c(mu,delta,beta,nu),log = FALSE,
           tolerance = .Machine$double.eps^0.5)
```

Arguments

`x,q`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of random variates to be generated.
`mu`	Location parameter μ, default is 0.
`delta`	Scale 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,log.p`	Logical; if `log = TRUE`, probabilities are given as log(p).
`method`	Character. If `"spline"` quantiles are found from a spline approximation to the distribution function. If `"integrate"`, the distribution function used is always obtained by integration.
`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.
`subdivisions`	The maximum number of subdivisions used to integrate the density and determine the accuracy of the distribution function calculation.
`intTol`	Value of `rel.tol` and hence `abs.tol` in calls to `integrate`. See `integrate`.
`valueOnly`	Logical. If `valueOnly = TRUE` calls to `pskewhyp` only return the value obtained for the integral. If `valueOnly = FALSE` an estimate of the accuracy of the numerical integration is also returned.
`nInterpol`	Number of points used in `qskewhyp` for cubic spline interpolation of the distribution function.
`uniTol`	Value of `tol` in calls to `uniroot`. See `uniroot`.
`...`	Passes additional arguments to `integrate` in `pskewhyp` and `qskewhyp`, and to `uniroot` in `qskewhyp`.

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. In addition the parameter values are examined by calling the function `skewhypCheckPars` to see if they are valid.

The density function is

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

when $\beta \neq 0$, and

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

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

`pskewhyp` uses the function `integrate` to numerically integrate the density function. The integration is from `-Inf` to `x` if `x` is to the left of the mode, and from `x` to `Inf` if `x` is to the right of the mode. The probability calculated this way is subtracted from 1 if required. Integration in this manner appears to make calculation of the quantile function more stable in extreme cases.

Calculation of quantiles using `qhyperb` permits the use of two different methods. Both methods use `uniroot` to find the value of x for which a given q is equal $F(x)$ where F denotes the cumulative distribution function. The difference is in how the numerical approximation to F is obtained. The obvious and more accurate method is to calculate the value of $F(x)$ whenever it is required using a call to `phyperb`. This is what is done if the method is specified as `"integrate"`. It is clear that the time required for this approach is roughly linear in the number of quantiles being calculated. A Q-Q plot of a large data set will clearly take some time. The alternative (and default) method is that for the major part of the distribution a spline approximation to $F(x)$ is calculated and quantiles found using `uniroot` with this approximation. For extreme values (for which the tail probability is less than 10^{-7}), the integration method is still used even when the method specifed is `"spline"`.

If accurate probabilities or quantiles are required, tolerances (`intTol` and `uniTol`) should be set to small values, say 10^{-10} or 10^{-12} with `method = "integrate"`. Generally then accuracy might be expected to be at least 10^{-9}. If the default values of the functions are used, accuracy can only be expected to be around 10^{-4}. Note that on 32-bit systems `.Machine$double.eps^0.25 = 0.0001220703` is a typical value.

Note that when small values of ν are used, and the density is skewed, there are often some extreme values generated by `rskewhyp`. These look like outliers, but are caused by the heaviness of the skewed tail, see **Examples**.

The extreme skewness of the distribution when β is large in absolute value and ν is small make this distribution very challenging numerically.

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 `valueOnly = FALSE` in the call to `pskewyhp` which returns a list with components `value` and `error`.

`ddskewhyp` gives the derivative of `dskewhyp`.

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, `skewhypCheckPars`, `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(x, param = param),
      range[1], range[2], n = 500)
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")
DistributionUtils::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)

##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")

##  example of density and random numbers for beta large and nu small
```

```
par(mfrow = c(1,2))
param1 <- c(0,1,10,1)
data1 <- rskewhyp(1000, param = param1)
curve(dskewhyp(x, param = param1), range(data1)[1], range(data1)[2],
      n = 1000, col = 2, main = "Density and Histogram -- when nu is small")
hist(data1, freq = FALSE, add = TRUE)

DistributionUtils::logHist(data1, main =
      "Log-Density and Log-Histogram -- when nu is small")
curve(dskewhyp(x, param = param1, log = TRUE),
      from = min(data1), to = max(data1), n = 500, add = TRUE, col = 2)
```

SkewHyperbolicPlots *Skew Hyperbolic Student 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-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-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	Scale 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(0,1,0,10)
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")
```

`skewhypFit` *Fit the Skew Hyperbolic Student 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 = FALSE, 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),...)
```

Arguments

`x`	Data vector for `skewhypFit`. Object of class `"skewhypFit"` for `print.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.
`...`	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 string 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, density, 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 Fitting a Skew Hyperbolic Student 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)
skewhypFitStartMM(x, nuTol = 1e-6, nuStart = 5, ...)
```

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)`.
`nuTol`	Tolerance for numerical determination of `nu` in `skewhypFitStartMM`
`nuStart`	Initial value for numerical determinationof `nu` in `skewhypFitStartMM`
`...`	Passes additional arguments to `skewhypFitStartMM`

Details

startValues can be either "US"(User-supplied), "LA"(Linear approximation), or "MM"(Method of moments).

If startValues = "US" then a value for paramStart must be supplied. The parameters are checked for validity by the function skewhypCheckPars.

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.

If startValues = "MM", start values are found using the method of moments as outlined in Aas and Haff (2006).

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 startValues = "MM", or to 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, skewhypCheckPars

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 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`	Scale parameter σ, 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)`.
`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.In addition the parameter values are examined by calling the function `skewhypCheckPars` to see if they are valid.

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, skewhypCheckPars, skewhypMom

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")
```

---

skewhypMom — *Calculate Moments of the Skew Hyperbolic Student t-Distribution.*

Description

This function can be used to calculate the raw moments, mu moments, central moments, and moments about any other given location for the skew hyperbolic *t*-distribution.

Usage

```
skewhypMom(order, mu = 0, delta = 1, beta = 1, nu = 1,
           param = c(mu,delta,beta,nu), 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 integer, except for moments about 0.
mu	Location parameter μ, default is 0.
delta	Scale 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).

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, default is zero. See **Details**.

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. In addition the parameter values are examined by calling the function skewhypCheckPars to see if they are valid.

order is also checked by calling the function is.wholenumber in the DistributionUtils package to see whether a whole number is given.

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

To calculate the moments of the skew hyperbolic *t*-distribution, the function first calculates the mu moments by the formula defined below, and then transforms them to any of the other types of moment by calling momChangeAbout in the DistributionUtils package.

The mu moments of the skew hyperbolic *t*-distribution are given by:

$$\bar{M}_k = \sum_{\ell=\lfloor (k+1)/2 \rfloor}^{k} a_{k,\ell} \beta^{2\ell-k} \left[\frac{\delta^{2\ell}\Gamma(\nu/2-\ell)}{\Gamma(\nu/2)2^{\ell}} \right]$$

where $k =$ order and $k > 0$ and $a_{k,\ell}$ is the recursive coefficient (see momRecursion for details).

This formula is given in Scott, Würtz and Tran (2008). Note that the [.] part of this formula is actually equivalent to the formula for the raw moments of the inverse gamma distribution, so the function calls gammaRawMom in the GeneralizedHyperbolic package when implementing the computations.

Value

The function returns 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>, Fiona Grimson

References

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

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

See Also

skewhypCheckPars, skewhypMean, is.wholenumber, momRecursion, momChangeAbout and gigMom.

Examples

```
param = c(1,2,3,10)
##Raw moments of the skew hyperbolic t distribution
skewhypMom(3, param = param, momType = "raw")
##Mu moments
skewhypMom(3, param = param, momType = "mu")
##Central moments
skewhypMom(3, param = param, momType = "central")
##Moments about any location
skewhypMom(3, param = param, about = 5)
```

---

skewhypParam *Parameter Sets for the Skew Hyperbolic t-Distribution*

Description

These objects store different parameter sets of the skew hyperbolic t distribution for testing or demonstrating purpose as matrices.

Specifically, the parameter sets skewhypSmallShape and skewhypLargeShape have constant location parameter of $\mu = 0$ and scale parameter of $\delta = 1$.

The skewness parameter β takes values from {0,2} in skewhypSmallShape and skewhypSmallParam, and from {-5,0,1,2,5} in skewhypLargeShape and skewhypLargeParam.

The shape parameter ν takes values from {1,5} in skewhypSmallShape and skewhypSmallParam, and from {1,2,5,10,20} in skewhypLargeShape and skewhypLargeParam.

Usage

```
data(skewhypParam)
```

Format

skewhypSmallShape: a 4 by 4 matrix;

skewhypLargeShape: a 25 by 4 matrix;

skewhypSmallParam: a 16 by 4 matrix;

skewhypLargeParam: a 400 by 4 matrix.

Author(s)

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

Examples

```
data(skewhypParam)
### Testing the accuracy of skewhypMean
for (i in 1:nrow(skewhypSmallParam)) {
  param <- skewhypSmallParam[i, ]
  x <- rskewhyp(1000, param = param)
  sampleMean <- mean(x)
  distMean <- skewhypMean(param = param)
  difference <- abs(sampleMean - distMean)
  print(difference)
}
```

---

skewhypTailPlotLine *Tail Plot Line*

Description

Adds skew hyperbolic *t*-distribution line to a tail plot

Usage

```
skewhypTailPlotLine(x, mu = 0, delta = 1, beta = 1, nu = 1,
                      param = c(mu,delta,beta,nu),
                      side = c("right", "left"), ...)
```

Arguments

x	A vector of values for which the tail plot has been drawn.
side	Character. "right" (the default) if the tail plot is of the right-hand tail, "left" if the tail plot is of the left-hand tail.
mu	Location parameter μ, default is 0.
delta	Scale 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).
...	Other graphical parameters (see par.

Details

The function tailPlot from **DistributionUtils** can be used to draw either a left-hand or right-hand tail plot of the data x. See for example Resnick (2007), p.105. The left-hand tail plot plots the empirical distribution of the data against the order statistics, for order statistic values below the median. The right-hand tail plot plots one minus the empirical distribution of the data against the order statistics, for order statistic values above the median. The default is for the y-axis to be plotted on a log scale.

skewhypTailPlotLine adds the line derived from the given skew hyperbolic t-distribution to an already drawn tail plot.

Value

Returns NULL invisibly.

Author(s)

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

References

Aas, Kjersti and Hobæk Haff, Ingrid (2006) The generalised hyperbolic skew Student's t-distribution. *Journal of Financial Econometrics*, **4**, 275–309.

Resnick, S. (2007) *Heavy-Tail Phenomena*, New York: Springer.

See Also

tailPlot and skewhypFit.

Examples

```
### Draw tail plot of some data
tailPlot <- DistributionUtils :: tailPlot  ## for convenience below
param <- c(0,1,1,10)
x <- rskewhyp(200, param = param)
tailPlot(x)
### Add skew hyperbolic t-distribution line
skewhypTailPlotLine(x, param = param)
### Parameters from fit may look better
paramFit <- skewhypFit(x, plots = FALSE)$param
tailPlot(x)
skewhypTailPlotLine(x, param = param)
skewhypTailPlotLine(x, param = paramFit, col = "steelblue")

### Left tail example
tailPlot(x, side = "l")
### Add skew hyperbolic t-distribution line
skewhypTailPlotLine(x, param = paramFit, side = "l")
### Log scale on both axes
tailPlot(x, side = "r", log = "xy")
### Add skew hyperbolic t-distribution line
skewhypTailPlotLine(x, param = paramFit, side = "r")
```

`summary.skewhypFit` *Summarising the Skew Hyperbolic Student t-Distribution Fit*

Description

`summary` Method for class `"skewhypFit"`.

Usage

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

Arguments

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

Details

`summary.skewhypFit` calculates standard errors for errors for the estimates of μ, δ, β and ν of the skew hyperbolic Student *t*-distribution parameter vector `param`, if the Hessian from the call to `optim` or `nlm` is available. Because the parameters in the call to the optimiser are μ, $\log(\delta)$, β and $\log(\nu)$ the delta method is used to obtain standard errors for δ and ν

Value

If the Hessian is available `summary.skewhyhpFit` computes standard errors of μ, δ, β and ν, and adds them to `object` as `object$sds`. Otherwise, no calculations are performed and the composition `object` is unaltered.

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

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

`print.summary.skewhypFit` prints a summary in the same format as `print.skewhypFit` 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`.

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

skewhypFit, dskewhyp, summary

Examples

```
## Continuing the skewhypFit(.) example:
data(lrdji)
djfit <- skewhypFit(lrdji, print = FALSE, plot = FALSE, hessian = TRUE)
print(djfit)
summary(djfit)
```

Index