Package 'GEVStableGarch'

July 26, 2015

Type Package

Title ARMA-GARCH/APARCH models with GEV and stable distributions

Version 1.1

Date 2015-07-19

Author Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes

Maintainer Thiago do Rego Sousa <thiagoestatistico@gmail.com>

Description Package for ARMA-GARCH or ARMA-
APARCH modelling with GEV and stable conditional distributions.

License GPL (>=2)

Depends R(>= 2.15.0),fGarch,fExtremes,stabledist,skewt,Rsolnp

Additional_repositories http://www.robustanalysis.com

R topics documented:

GEVStableGarch-package *ARMA-GARCH/APARCH modelling with GEV and stable distributions*

Description

This package is design to perform maximum likelihood estimation of ARMA-GARCH/APARCH models with Generalized Extreme Distribution (GEV) or stable conditional distributions.

Details

Package:	GEVStableGarch
Type:	Package
Version:	1.1
Date:	2015-07-19
License:	GPL(>=2)
Depends:	R(>= 2.15.0), fGarch, fExtremes, stabledist, skewt, Rsolnp

This package contains functions for estimating and simulating combined ARMA-GARCH or ARMA-APARCH models with error distributions following GEV or stable densities. The current version has a new algorithm that allows the user to enforce stationarity during estimation. The package also allows the user to estimate several models and choose the best one according to a goodness-of-fit Criteria.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes

Maintainer: Thiago do Rego Sousa <thiagoestatistico@gmail.com>

References

Brockwell, P. J, e Davis, R. A. Introduction to Time Series and Forecasting. Springer, New York, 1996.

Mittnik S. , Paolella M.S. , Rachev, S.T. Stationarity of stable power- GARCH processes. Journal of Econometrics 106, 97-107. 2002.

Nolan, J. P. Numerical calculations of stable densities and distribution functions. Communications in Statistics - Stochastic Models 13: 759- 774, 1997.

Nolan, J. P. Maximum likelihood estimation and diagnostics for stable distribution. In O. E.Barndorff-Nielsen, T. Mikosch, e S. I. Resnick (Eds.), Levy Processes: Theory and Applications: 379-400. Boston, Birkhauser, 2000.

Wurtz, D., Y. Chalabi, e L. Luksan. Parameter estimation of ARMA models with GARCH/APARCH errors: An R and SPlus software implementation. Journal of Statistical Software, 2006.

Zhao X. ,Scarrott C. J. , Oxley L. , Reale M. GARCH dependence in extreme value models with Bayesian inference. Mathematics an Computers in Simulation, Vol. 81, Issue 7, 1430-1440. 2011.

fGEVSTABLEGARCH-class *Class* "fGEVSTABLEGARCH"

Description

The class fGEVSTABLEGARCH represents an ARMA-GARCH/APARCH model estimated wih function `gsFit`

Objects from the Class

Objects can be created by calls of the form `gsFit`. This object contain the estimated parameters of a time series process.

Slots

`call`: Object of class `"call"`: the call of the `gsFit` function.

`formula`: Object of class `"formula"`: a formula object specifying mean and variance equation.

`method`: Object of class `"character"`: a string denoting the optimization method.

`convergence`: Object of class `"numeric"`: an integer code. 0 indicates successful convergence of the of the estimation routine. This variable is the same returned by the one of the optimization methods, which could be one of `solnp` or nlminb routines. Notice that sometimes the optimization algorithm will return a "true" convergence, even when the optimized negative log-likelihood equals to 1e99. In this case, the variable convergence will be equal to 1 to indicate that convergence was not achieved.

`messages`: Object of class `"list"`: a character string giving any additional information collected during the estimation of the model.

`data`: Object of class `"numeric"`: a numeric vector containing the data of the time series to be estimated.

`fit`: Object of class `"list"`: a list with the results from the parameter estimation.

`residuals`: Object of class `"numeric"`: a numeric vector with the residual values.

`h.t`: Object of class `"numeric"`: a numeric vector with the conditional variances.

`sigma.t`: Object of class `"numeric"`: a numeric vector with the conditional standard deviations.

`title`: Object of class `"character"`: a string with the title.

`description`: Object of class `"character"`: a string with a description.

Methods

show `signature(object = "fGEVSTABLEGARCH")`: prints an object of class 'fGEVSTABLEGARCH'.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes.

See Also

`gsFit`

fGEVSTABLEGARCHSPEC-class

Class "fGEVSTABLEGARCHSPEC"

Description

This class represents an ARMA-GARCH/APARCH model with innovations following an i.i.d stable, GEV distributions. Other distributions are also allowed.

Objects from the Class

Objects can be created by calls of the function `gsSpec`. This object specifies an univariate ARMA-GARCH/APARCH model with GEV or stable conditional distribution. Note that we only allow the user to create an object that do not violate the model specifications. For more details about the definition of those models see: see Mittnik et al. (2002) (stable innovations), Zhao et al. (2011) (GEV innovations) and Wuertz et al. (2008) (finite variance innovations).

Slots

`call`**:** Object of class `"call"`: the call of the `gsSpec` function.

`formula`**:** Object of class `"formula"`: a list with two formula entries for the mean and variance equation of the combined ARMA-GARCH/APARCH model.

`model`**:** Object of class `"list"`: a list with the model parameters specifying the Autorregressive, the GARCH/APARCH and the innovations parameters.

`presample`**:** Object of class `"matrix"`: a numeric matrix with presample values.

`distribution`**:** Object of class `"character"`: a character string with the name of the conditional distribution.

`rseed`**:** Object of class `"numeric"`: an integer with the random number generator seed.

Methods

show `signature(object = "fGEVSTABLEGARCHSPEC")`: prints an object of class 'fGEVSTABLE-GARCH'

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes.

References

Mittnik S, Paolella M, Rachev S (2002). "Stationarity of Stable Power-GARCH Processes." Journal of Econometrics, 106, 97-107.

Wuertz D, Chalabi Y, Luksan L (2009). "Parameter Estimation of ARMA Models with GARCH/APARCH Errors: An R and SPlus Software Implementation." Journal of Statistical Software, forthcoming. URL http://www-stat.wharton.upenn.edu/~steele/Courses/956/RResources/GarchAndR/WurtzEtAlGarch.pdf.

Zhao X, Scarrott C, Oxley L, Reale M (2011). "GARCH Dependence in Extreme Value Models with Bayesian Inference." Mathematics and Computers in Simulation, 81(7), 1430-1440.

GAt *Generalized Asymmetric t Distribution*

Description

Functions to compute density, distribution function, quantile function and to generate random variates for the generalized asymmetric t distribution (GAt). Notice that this is the same t3-distribution mentioned in the literature defined by Paolella (1997). The GAt distribution includes the Student's t, Laplace, Cauchy and the normal distribution when the shape parameter (nu) goes to infinity (see see Mittnik and Paolella (2000)).

Usage

```
dGAt(x, mean = 0, sd = 1, nu = 2, d = 3, xi = 1, log = FALSE)
pGAt(q, mean = 0, sd = 1, nu = 2, d = 3, xi = 1)
qGAt(p, mean = 0, sd = 1, nu = 2, d = 3, xi = 1)
rGAt(n, mean = 0, sd = 1, nu = 2, d = 3, xi = 1)
```

Arguments

`mean, sd, d, nu, xi`
: location parameter `mean`, scale parameter `sd`, shape 1 parameter `nu`, shape 2 parameter `d`, asymmetry parameter `xi`.

`n`
: the number of observations.

`p`
: a numeric vector of probabilities.

`x, q`
: a numeric vector of quantiles.

`log`
: a logical; if TRUE, densities are given as log densities.

Value

`d*` returns the density, `p*` returns the distribution function, `q*` returns the quantile function, and `r*` generates random deviates,
all values are numeric vectors.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes.

References

Mittnik S, Paolella M, Rachev S (2002). "Stationarity of Stable Power-GARCH Processes" Journal of Econometrics, 106, 97-107.

Paolella M (1997). Tail Estimation and Conditional Modeling of Heteroskedstic Time-Series, PhD Thesis Institute of Statistics and Econometrics. Ph.D. thesis, Christian Albrechts University of Kiel.

Examples

```
# Simulate Random Values and compare with
# the empirical density and probability functions

# Configure plot and generate random values
par(mfrow = c(2, 2))
set.seed(1000)
r = rGAt(n = 1000)
plot(r, type = "l", main = "GAt Random Values", col = "steelblue")

# Plot empirical density and compare with true density:
hist(r, n = 25, probability = TRUE, border = "white", col = "steelblue")
box()
x = seq(min(r), max(r), length = 201)
lines(x, dGAt(x), lwd = 2)

# Plot density function and compare with true df:
plot(sort(r), (1:1000/1000), main = "Probability", col = "steelblue",
     ylab = "Probability")
lines(x, pGAt(x), lwd = 2)

# Compute quantiles:
# Here we compute the quantiles corresponding to the probability points from
# -10 to 10 and expect to obtain the same input sequence
round(qGAt(pGAt(q = seq(-10, 10, by = 0.5))), digits = 6)
```

---

gsFit *Estimation of ARMA-GARCH/APARCH models*

Description

This function uses Maximum Likelihood technique to estimate the parameters of ARMA-GARCH or ARMA-APARCH model with several conditional distributions. The user can also enforce stationarity during estimation

Usage

```
gsFit(formula = ~garch(1, 1), data, cond.dist = c("stableS0", "stableS1",
"stableS2", "gev", "GAt", "norm", "std", "sstd", "skstd", "ged"),
include.mean = TRUE, algorithm = c("sqp", "sqp.restriction",
"nlminb", "nlminb+nm"), control = NULL,
tolerance = NULL, title = NULL, description = NULL)
```

Arguments

`formula`	a formula object specifying mean and variance equation.
`data`	a numeric vector containing the data of the time series to be estimated.

`cond.dist` a character string naming conditional distribution of innovations. The package was created to accept the following distributions: `"stableS0"`, `"stableS1"`, `"stableS2"`, `"gev"` and `"GAt"`. Other common distributions are also possible such as the `"norm"`, `"std"`, `"sstd"`, `"skstd"`, `"ged"`, but they are not the main feature of this work.

`include.mean` This is a boolean variable. It intercept is TRUE than we estimate the model with intercept, otherwise we will not use the intercept variable.

`algorithm` The algorithm to be used to optimize the likelihood of the process. This parameter should be one of the following strings: "nlminb" (nlminb R internal Routine for optimization) or "SQP" (Sequential quadratic programming routine implemented in package RSolnp).

`control` Control parameters used either in "SQP" routine (see the RSolnp documentation) or in "nlminb" (see the nlminb R documentation).

`tolerance` Tolerance for parameter estimation.

`title` a string with the title.

`description` a string with a description.

Details

The starting values are key in getting any possible convergence value and they were chosen to reflect this. For example, the GEV shape starting value was chosen to be 0.01 since in many cases we can not obtain good results. There is also the possibility to fit models with conditional "Normal", "t-student" and "skew t-student" distributions but the main purpose of this routine is to deal with "GEV" and "stable". The parameters will be interpreted according to the following equations (see Wurtz et al. ,2006)

$$X_t = \mu + \sum_{i=1}^{m} a_i X_{t-i} + \sum_{j=1}^{n} b_j \varepsilon_{t-j} + \varepsilon_t$$

$$\varepsilon_t = \sigma_t z_t, \quad z_t \overset{iid}{\sim} D(0,1) \ ,$$

$$\sigma_t^\delta = \omega + \sum_{i=1}^{p} \alpha_i (\varepsilon_{t-i} - \gamma_i |\varepsilon_{t-i}|)^\delta + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta$$

where

$$X_t$$

is the ARMA process,

$$\sigma_t$$

is the APARCH process and

$$D(0,1)$$

is the standard density distribution to be used in the model.

Value

`gsFit`

returns a S4 object of class `"fGEVSTABLEGARCH"` with the following slots:

`@call` the call of the `gsFit` function.

`@formula` a list with two formula entries, one for the mean and the other one for the variance equation. Note that only ARMA, GARCH or APARCH models.

`@method`	a string denoting the optimization method.
`@data`	a numeric vector containing the data of the time series to be estimated.
`@fit`	a list with the results from the parameter estimation.
`@residuals`	a numeric vector with the residual values.
`@h.t`	a numeric vector with the residual values ($h_t = \sigma_t^\delta$).
`@sigma.t`	a numeric vector with the conditional standard deviation.
`@title`	a string with the title.
`@description`	a string with a description.

The entries of the @fit slot show the results from the optimization.

Enhances

The estimation of ARMA-GARCH/APARCH models with conditional stable distribution is mainly dependent on the time taken during the calculation of density points. Our routine uses the standard R stable density implemented in package **stabledist**. There is also another numerical routine to calculate stable densities which is part of an R package called **stable**. This package implements a faster computation of stable densities that are accurately enough to perform numerical optimization. When **GEVStableGarch** loads it verifies if package **stable** is available. If it is found the function GSGarch.Fit will perform optimization using **stable**'s routine. Otherwise it will use stable density from package **stabledist**. Package **stable** is available at `http://www.robustanalysis.com`.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes

References

Mittnik S. , Paolella M.S. , Rachev, S.T. Stationarity of stable power- GARCH processes. Journal of Econometrics 106, 97-107. 2002.

Wurtz, D., Y. Chalabi, e L. Luksan. Parameter estimation of ARMA models with GARCH/APARCH errors: An R and SPlus software implementation. Journal of Statistical Software, 2006.

Zhao X. ,Scarrott C. J. , Oxley L. , Reale M. GARCH dependence in extreme value models with Bayesian inference. Mathematics an Computers in Simulation, Vol. 81, Issue 7, 1430-1440. 2011.

Yinyu Ye. Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming. Ph.D. Thesis, Department of EES, Stanford University, 1987.

Examples

```
# This examples uses the dataset of the package fGarch to estimate
# an ARMA(1,1)-GARCH(1,1) with GEV conditional distribution.
library(fGarch)
data(dem2gbp)
x = dem2gbp[, 1]
gev.model = gsFit(data = x , formula = ~garch(1,1), cond.dist = "norm")
```

gsMomentAparch *Computation of moments for several conditional distribution*

Description

Computation of the moments expression E(|Z| - gamma * Z) ^ delta

Usage

```
gsMomentAparch(cond.dist = c("stableS1", "gev", "GAt", "norm",
"std", "sstd", "skstd", "ged"), shape = 1.5, skew = 0, delta = 1, gm = 0)
```

Arguments

`cond.dist`	a character string naming conditional distribution of innovations. The package was created to accept the following distributions: `"stableS0"`, `"stableS1"`, `"stableS2"`, `"gev"` and `"GAt"`. Other common distributions are also possible such as the `"norm"`, `"std"`, `"sstd"`, `"skstd"`, `"ged"`, but they are not the main feature of this work.
`shape, skew`	The shape and skew parameter for the conditional distribution.
`delta, gm`	The delta and gm parameteres of the ARMA-GARCH/APARCH model

Details

EXPLICAR COMO EH FEITO ESSES CALCULOS. COLOCAR EXPRESSOES DO ARTIGO.

Value

Returns the following expression for several conditional distributions E(|Z| - gamma * Z) ^ delta

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes

References

NOSSO ARTIGO JSS

Diongue AK (2008). "An investigation of Stable-Paretian Asymmetric Power GARCH Model." Journal des Sciences, 8(4), 15-26.

Mittnik S, Paolella MS (2000). "Conditional Density and Value-At-Risk Prediction of Asian Currency Exchange Rates." Journal of Forecasting, 19(4), 313-333.

Mittnik S, Paolella M, Rachev S (2002). "Stationarity of Stable Power-GARCH Processes." Journal of Econometrics, 106, 97-107.

Examples

```
# EXEMPLOS GSMOMENTAPARCH
```

`gsSelect` *Selects the best model according to goodness-of-fit*

Description

This function estimates ARMA-GARCH/APARCH models with varying order and return the one with minimum AIC, BIC or AICc

Usage

```
gsSelect(data, order.max = c(1, 1, 1, 1),
selection.criteria = c("AIC", "AICc", "BIC"), is.aparch = FALSE,
cond.dist = c("stableS0", "stableS1", "stableS2", "gev",
"GAt", "norm", "std", "sstd", "skstd", "ged"),
include.mean = TRUE,
algorithm = c("sqp", "sqp.restriction", "nlminb", "nlminb+nm"), ...)
```

Arguments

`data`	Data with the time series to be estimated. This object can not contain NULL elements.
`order.max`	Maximum order of models to search
`selection.criteria`	The goodness-of-fit measure to use be minimized.
`is.aparch`	Boolean variable indicating whether to search for ARMA-GARCH or ARMA-APARCH models
`cond.dist`	a character string naming conditional distribution of innovations. The package was created to accept the following distributions: `"stableS0"`, `"stableS1"`, `"stableS2"`, `"gev"` and `"GAt"`. Other common distributions are also possible such as the `"norm"`, `"std"`, `"sstd"`, `"skstd"`, `"ged"`, but they are not the main feature of this work.
`include.mean`	This is a boolean variable. It intercept is TRUE than we estimate the model with intercept, otherwise we will not use the intercept variable.
`algorithm`	The algorithm to be used to optimize the likelihood of the process. This parameter should be one of the following strings: "nlminb" (nlminb R internal Routine for optimization) or "SQP" (Sequential quadratic programming routine implemented in package RSolnp).
`...`	Additional parameters to be passed to function GSfit during estimation.

Value

The model with minimum AIC found within the specified range in parameters mMAX, nMAX, pMAX, qMAX.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes

References

Brockwell, P. J, e Davis, R. A. Introduction to Time Series and Forecasting. Springer, New York, 1996.

Examples

```
# AIC fit using models from ARMA(0,0)-GARCH(1,0) to ARMA(1,1)-GARCH(1,1)
library(fGarch)
data(dem2gbp)
x = dem2gbp[,1]

model = gsSelect (data = x, order.max = c(0,0,1,1), is.aparch = FALSE,
          algorithm = "sqp", cond.dist = "norm", selection.criteria = "BIC")
```

`gsSim` *Simulation of ARMA-GARCH/APARCH process*

Description

This functions simulate time series following ARMA-GARCH/APARCH models with several conditional distributions, including GEV and stable distributions.

Usage

```
gsSim(spec = garchSpec(), n = 100, n.start = 100)
```

Arguments

`spec`	A model specified with function gsSpec
`n`	The size of simulated time serie
`n.start`	

Details

The initial values of the time series are fixed and the recursion formulas of the model are used to simulate the dynamics of the process. We do not verify the stationarity conditions of the model because the simulation of non-stationary process could also be of interest.

Value

The function returns an object containing the following items:

`model`	A string describing the estimated model.
`cond.dist`	The conditional distribution used to fit the model.
`series`	An array of two columns. The first column is the simulated process $$X_t$$ and the second one is the $$\sigma_t$$ simulated process.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes.

References

Brockwell, P. J, e Davis, R. A. Introduction to Time Series and Forecasting. Springer, New York, 1996.

Nolan, J. P. Numerical calculations of stable densities and distribution functions. Communications in Statistics - Stochastic Models 13: 759- 774, 1997.

Examples

```
# Simulation of a ARMA-APARCH process with stable conditional distribution
#x <- GSgarch.Sim(N = 2500, mu = 0.1,a = c(0.2,0.3),b = c(0.2,0.5),
#omega = 0.1, alpha = c(0.1,0.2),beta = c(0.1,0.1),gm=c(0.3,-0.3),
#delta = 1,skew = 0.3,shape = 1.9, cond.dis = "stable")
```

---

gsSpec *Specification of ARMA-GARCH/APARCH models with GEV or stable distributions*

Description

Specifies an ARMA-GARCH or ARMA-APARCH model with innovations following GEV, stable or other common conditional distributions.

Usage

```
gsSpec(model = list(), presample = NULL,
    cond.dist = c("stableS0", "stableS1", "stableS2", "gev",
    "GAt", "norm", "std", "sstd", "skstd", "ged"),
    rseed = NULL)
```

Arguments

`model` a list of ARMA-GARCH/APARCH model parameters:
`omega` - the coefficient of the variance equation, by default 1e-6;
`alpha` - the value or vector of autoregressive coefficients;
`beta` - the value or vector of variance coefficients;
The values for the ARMA part are:
`mu` - the mean value;
`ar` - the autoregressive coefficients;
`ma` - the moving average coefficients.
The parameters for the conditional distributions are:
`skew` - the skewness parameter;
`shape` - the shape parameter.

`presample` presample - a numeric "matrix" with 3 columns and at least max(m,n,p,q) rows. The first culumn are the innovations, the second the conditional variances, and the last the time series. When presample is missing, we construct our presample matrix as [z,h,y] where z = rnorm(0,1), h = "uev" recursion initialization described in Wuertz et al. (2006) and y = mu. Note that the conditional variance column can contain only strictly positive numbers and the function `gsSpec` check for invalid values inside the presample matrix.

`cond.dist` a character string naming conditional distribution of innovations. The package was created to accept the following distributions: `"stableS0"`, `"stableS1"`, `"stableS2"`, `"gev"` and `"GAt"`. Other common distributions are also possible such as the `"norm"`, `"std"`, `"sstd"`, `"skstd"`, `"ged"`, but they are not the main feature of this work.

`rseed` the seed for the intitialization of the random number generator for the innovations.

Details

This functions uses the interface of the `garchSpec` routine from package **fGarch** to simulate random values of the ARMA-GARCH/APARCH model with conditional GEV or stable distribution.

Value

The returned value is an object of class `"fGEVSTABLEGARCHSPEC"`.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes.

References

Mittnik S, Paolella M, Rachev S (2002). "Stationarity of Stable Power-GARCH Processes." Journal of Econometrics, 106, 97-107.

PACOTE FGARCH

Examples

```
# stable-GARCH from Curto et al. (2009) for the DJIA dataset
spec.stable = gsSpec(model = list(mu = 0.0596, omega = 0.0061, alpha = 0.0497, beta = 0.9325, skew = -0.9516, 
sim.stable = gsSim(spec = spec.stable, n = 1000)

# GEV-GARCH model from Zhao et al. (2011)
spec.gev = gsSpec(model = list(mu = 0.21, a = 0.32, omega = 0.01,
alpha = 0.45, beta = 0.08, shape = 0.08), cond.dist = "gev")
sim.gev = gsSim(spec = spec.gev, n = 1000)
```

`show-methods` *GEVSTABLEGARCH Package Show Methods*

Description

Methods to organize the output to the user when printing objects of class GEVSTABLEGARCH

Methods

`signature(object = "fGEVSTABLEGARCH")` Print function for objects of class `"fGEVSTABLEGARCH"`

`signature(object = "fGEVSTABLEGARCHSPEC")` Print function for objects of class `"fGEVSTABLEGARCHSPEC"`

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes.

`skstd` *Skew Student's t Distribtuion from Fernandez and Steel (1997)*

Description

Functions to compute density, distribution function, quantile function and to generate random variates for the Skew Student's t Distribtuion from Fernandez and Steel (1997). Notice that this function is different from the skew Student's t `sstd` from package fGarch. Although the two distributions use the same approach from Fernandez and Steel (1997), the second one was reparameterized in such a way that when the parameters mean = 0 and sd = 1 the distribution will have a true mean equal zero and the true variance equals to one, no matter the value of the skew parameter. The distributions `skstd` and `sstd` are the same when the asymmetry parameter `xi` equals to 1, in which case they are symmetric.

Usage

```
dskstd(x, mean = 0, sd = 1, nu = 3, xi = 1, log = FALSE)
pskstd(q, mean = 0, sd = 1, nu = 3, xi = 1)
qskstd(p, mean = 0, sd = 1, nu = 3, xi = 1)
rskstd(n, mean = 0, sd = 1, nu = 3, xi = 1)
```

Arguments

`mean, sd, nu, xi`
location parameter `mean`, scale parameter `sd`, shape parameter `nu`. asymmetry parameter `xi`.

`n`	the number of observations.
`p`	a numeric vector of probabilities.
`x, q`	a numeric vector of quantiles.
`log`	a logical; if TRUE, densities are given as log densities.

Value

`d*` returns the density, `p*` returns the distribution function, `q*` returns the quantile function, and `r*` generates random deviates,
all values are numeric vectors.

Author(s)

Thiago do Rego Sousa, Cira Etheowalda Guevara Otiniano and Silvia Regina Costa Lopes.

References

Fernandez Steel. "On Bayesian Modeling of Fat Tails and Skewness." Journal of the American Statistical Association, 93(441), 359-371

Examples

```
# Simulate Random Values and compare with
# the empirical density and probability functions

# Configure plot and generate random values
par(mfrow = c(2, 2))
set.seed(1000)
r = rskstd(n = 1000)
plot(r, type = "l", main = "Skew Student's t Random Values", col = "steelblue")

# Plot empirical density and compare with true density:
hist(r, n = 25, probability = TRUE, border = "white", col = "steelblue")
box()
x = seq(min(r), max(r), length = 201)
lines(x, dskstd(x), lwd = 2)

# Plot density function and compare with true df:
plot(sort(r), (1:1000/1000), main = "Probability", col = "steelblue",
     ylab = "Probability")
lines(x, pskstd(x), lwd = 2)

# Compute quantiles:
# Here we compute the quantiles corresponding to the probability points from
# -10 to 10 and expect to obtain the same input sequence
round(qskstd(pskstd(q = seq(-10, 10, by = 0.5))), digits = 6)
```

Index