

Package ‘MSGARCH’

July 27, 2016

Type Package

Title Markov-Switching GARCH models

Version 0.10

Date 2016-07-27

Author Keven Bluteau [aut, cre],
David Ardia [aut]
Kris Boudt [ctb]
Brian Peterson [ctb]

Maintainer Keven Bluteau <keven.bluteau@unine.ch>

Description The MSGARCH package offer functionalities to fit (by Maximum Likelihood or Bayesian), simulate, and forecast various Markov-Switching GARCH processes.

License GPL (>= 2)

Imports Rcpp,
adaptMCMC,
DEoptim,
nloptr,
methods,
stringr,
ggplot2,
reshape2

LinkingTo Rcpp, RcppArmadillo

Depends RcppArmadillo

Suggests knitr,
rmarkdown,
testthat

VignetteBuilder knitr

RoxygenNote 5.0.1

R topics documented:

MSGARCH-package	2
AIC	3
BIC	4
cdf	5

create.spec	6
DIC	8
fit.bayes	9
fit.mle	10
ht	12
kernel	13
pdf	14
pit	15
Plast	16
pred	16
Pstate	17
risk	18
rnd	19
sim	20
sp500ret	21
unc.vol	21
Index	23

MSGARCH-package

The R package MSGARCH

Description

The R package MSGARCH aims to provide a comprehensive set of functionalities for Markov-switching GARCH processes, including fitting, filtering, forecasting, and simulating. Other functions related to Value-at-Risk, Expected-Shortfall, and conditional distributions are also available. The main functions of the package are coded in C++ with Rcpp (Eddelbuettel and Francois, 2011) and RcppArmadillo (Eddelbuettel and Sanderson, 2014). In the R package MSGARCH there is no equation for the mean as in the R package rugarch (Ghalanos, 2015). This means that we assume that before modeling, the user has filter the mean from their time series.

We provide a variety of single-regime GARCH process and regime-switching process as well as many conditional distributions. This allows for a rich modeling environment for Markov-switching GARCH models. Each single-regime process in a one-lag process (e.g, GARCH(1,1)). Allowing for only one-lag is sufficient in Finance and reduce models complexity which can become a problem during the optimization process.

The authors acknowledge Google for financial support via the Google Summer of Code 2016 project "MSGARCH"; see <https://summerofcode.withgoogle.com/projects/#6497774455488512>, the International Institute of Forecasting and Industrielle-Alliance.

References

- Eddelbuettel, D. & Francois, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, 40, pp. 1-18, <http://www.jstatsoft.org/v40/i08/>.
- Eddelbuettel, D. & Sanderson, C. (2014). RcppArmadillo: Accelerating R with High-Performance C++ Linear Algebra. *Computational Statistics & Data Analysis*, 71, pp. 1054-1063, <http://dx.doi.org/10.1016/j.csda.2013.02.005>.
- Ghalanos, A. (2015). rugarch: Univariate GARCH Models. <https://cran.r-project.org/web/packages/rugarch/>.

AIC

*Compute Akaike information criterion (AIC).***Description**

Compute Akaike information criterion (AIC).

Usage

```
AIC(fit)
```

Arguments

`fit` Fit object of type MSGARCH_MLE_FIT created with `fit.mle` or MSGARCH_BAY_FIT created with `fit.bayes`.

Details

If a matrix of MCMC posterior draws estimates is given, the AIC on the posterior mean is calculated.

Value

AIC value.

References

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19, pp. 716-723.

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# fit the model by MLE
fit = MSGARCH::fit.mle(spec = spec, y = sp500ret)

# compute AIC
AIC = MSGARCH::AIC(fit = fit)
```

BIC

*Compute Bayesian information criterion (BIC).***Description**

Compute Bayesian information criterion (BIC).

Usage

```
BIC(fit)
```

Arguments

`fit` Fit object of type MSGARCH_MLE_FIT created with `fit.mle` or MSGARCH_BAY_FIT created with `fit.bayes`

Details

If a matrix of MCMC posterior draws estimates is given, the BIC on the posterior mean is calculated.

Value

BIC value.

References

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, pp. 461-464.

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# fit the model by MLE
fit = MSGARCH::fit.mle(spec = spec, y = sp500ret)

# compute BIC
BIC = MSGARCH::BIC(fit = fit)
```

cdf	<i>Cumulative density function at $T + 1$.</i>
-----	---

Description

Method returning the cumulative density at $T + 1$ of a vector of points.

Usage

```
cdf(spec, x, theta, y, log = TRUE)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
x	Vector (of size N) of point to be evaluated.
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.
log	Boolean indicating if the log cumulative is returned. (Default: log = TRUE)

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually. The [cdf](#) method uses the last variance estimate by filtering.

Value

(Log-)Cumulative density at $T + 1$ of the points x (vector of size N or matrix of size M x N).

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# generate random draws
x = rnorm(100)

# run cdf method on random draws
cdf = MSGARCH::cdf(spec = spec, x = x, theta = spec$theta0, y = sp500ret, log = FALSE)
```

create.spec	<i>Specification creation</i>
-------------	-------------------------------

Description

Function for creating a variance specification before fitting and using the R package MSGARCH functionalities.

Usage

```
create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
  do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
```

Arguments

model	Vector (of size K) containing the variance model specifications. Valid models are "sGARCH", "eGARCH", "gjrGARCH", "tGARCH", and "GAS". (Default: model = c("sGARCH", "sGARCH"))
distribution	Vector (of size K) of conditional densities. Valid distribution are "norm", "std", and "ged". The vector must be of the same length as the models vector. (Default: distribution = c("norm", "norm"))
do.skew	Vector (of size K) of boolean indicating if the conditional density is skewed. The vector must be of the same length as the distributions vector. (Default: do.skew = c(FALSE, FALSE))
do.mix	Boolean indicating if the specification is a mixture type. If the argument is TRUE, a Mixture of GARCH is created, while if the argument is FALSE, a Markov-Switching GARCH is created (see details). (Default: do.mix = FALSE)
do.shape.ind	Boolean indicating if the distribution are Regime-Independent. If the argument is TRUE, all distributions are the same and the distribution parameters does not dependent on the regime in which the distribution is attributed to. If the argument is TRUE, all distributions in the distribution argument and all skew argument must be the same. (Default: do.shape.ind = FALSE)

Details

The Markov-Switching specification created is based on the Haas et al. (2004a) MSGARCH specification. It is a MSGARCH model that is separated in K single-regimes specifications which are updated in parallel. Under this specification, the conditional variance is a function of the past data and the current state. The Mixture of GARCH option is based on the Haas et al. (2004b). A Mixture of GARCH is a mixture of distribution where the variance process of each distribution is a single-regime process.

Value

A list of class MSGARCH_SPEC containing variables related to the created specification.
The list contains:

- theta0 : Vector (of size d) of default parameters.
- is.mix : Boolean indicating if the specification is a mixture.

- `is.shape.ind` : Boolean indicating if the distribution parameters are regime-independent.
- `K` : Number of regimes.
- `sigma0` : Default variance-covariance matrix (of size $K \times K$) used for the Bayesian estimation.
- `lower` : Vector (of size d) of lower parameters bound.
- `upper` : Vector (of size d) of upper parameters bound.
- `ineqlb` : Vector (of size d) of lower inequality bound.
- `inequb` : Vector (of size d) of upper inequality bound.
- `n.params` : Vector (of size K) of the total number of parameters by regime including distribution parameters.
- `n.params.vol` : Vector (of size K) of the total number of parameters by regime excluding distribution parameters.
- `do.init` : Boolean indicating the default `do.init` argument.
- `label` : Vector (of size d) of parameters label.
- `name` : Vector (of size K) of model specification name.
- `func` : List of R functions internally used.
- `rcpp.func` : List of Rcpp functions internally used.

The MSGARCH_SPEC class possesses these methods:

- `sim` : Simulation method.
- `ht` : Conditional volatility in each regime.
- `kernel` : Kernel method.
- `unc.vol` : Unconditional volatility in each regime.
- `pred` : Predictive density method at $T + 1$.
- `pit` : Probability Integral Transform at $T + 1$.
- `risk` : Value-at-Risk And Expected-Shortfall methods at $T + 1$.
- `rnd` : Simulation method at $T + 1$.
- `pdf` : Probability density function at $T + 1$.
- `cdf` : Cumulative density function at $T + 1$.
- `Pstate` : State probabilities filtering method.
- `Plast` : State probabilities at $T + 1$.
- `fit.mle` : Maximum Likelihood estimation.
- `fit.bayes` : Bayesian estimation.
- `print` and `summary` : Summary of the created specification.

References

- Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31, pp. 307-327.
- Creal, D. Koopman, S. J. & Lucas, A. (2013). Generalized Autoregressive Score Models with Applications. *Journal of Applied Econometrics*, 28, pp. 777-795.
- Fernandez, C. & Steel, M. F. (1998). On Bayesian Modeling of Fat Tails and Skewness. *Journal of the American Statistical Association*, 93, pp. 359-371.

Glosten, L. R. Jagannathan, R. & Runkle, D. E. (1993). On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. *Journal of Finance*, 48, pp. 1779-1801.

Haas, M. Mittnik, S. & Paoletta, M. S. (2004a). A New Approach to Markov-Switching GARCH Models. *Journal of Financial Econometrics*, 2, pp. 493-530.

Haas, M. Mittnik, S. & Paoletta, M. S. (2004b). Mixed Normal Conditional Heteroskedasticity. *Journal of Financial Econometrics*, 2, pp. 211-250.

Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, 59, pp. 347-370.

Zakoian, J.-M. (1994). Threshold Heteroskedastic Models *Journal of Economic. Dynamics and Control*, 18, pp. 931-955.

Examples

```
# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "gjrGARCH"), distribution = c("norm", "std"),
                             do.skew = c(TRUE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
print(spec)
```

DIC

Compute Deviance Information Criterion (DIC).

Description

Compute Deviance Information Criterion (DIC).

Usage

`DIC(fit)`

Arguments

`fit` Fit object of type MSGARCH_BAY_FIT created with `fit.bayes`.

Value

A list containing six variables:

- DIC : Deviance Information Criterion.
- IC : Bayesian Predictive Information Criterion.
- pD : Effective number of parameters ($pD = \bar{D} - \hat{D}$)
- pV : Effective number of parameters ($pV = \text{var}(\hat{D})/2$)
- D.bar: Expected value of the deviance over the posterior
- D.hat: Deviance at the mean posterior estimate

References

Gelman, A. Carlin, J. B. Stern, H. S. & Rubin, D. B. (2003). Bayesian Data Analysis. *Chapman and Hall/CRC*

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# fit the model by Bayesian estimation
fit = MSGARCH::fit.bayes(spec = spec, y = sp500ret)

# compute DIC
DIC = MSGARCH::DIC(fit = fit)
```

fit.bayes	<i>Bayesian estimation.</i>
-----------	-----------------------------

Description

Method that performs Bayesian estimation of a MSGARCH_SPEC object on a set of observations.

Usage

```
fit.bayes(spec, y, ctr = list())
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
y	Vector (of size T) of observations.
ctr	A list of control parameters. The control parameters have three components: <ul style="list-style-type: none"> • N.burn (integer ≥ 0): Number of discarded draws. (default: N.burn = 1000) • N.mcmc (integer > 0): Number of draws. (default: N.mcmc = 5000) • N.thin (integer > 0): Thinning factor (every N.thin draws are kept). (default: N.thin = 10)

Details

The total number of draws is equal to N.mcmc / N.thin. The Bayesian estimation uses the R package adaptMCMC (Andreas, 2012) which implements the adaptive sampler of Vihola (2012).

Value

A list of class MSGARCH_BAY_FIT containing four components:

- theta : The MCMC chain (matrix from the R package coda (Plummer et al., 2006) of size N.mcmc / N.thin).
- accept : Acceptation rate of the sampler.
- y : Vector (of size T) of observations.

- spec : Specification.

The MSGARCH_BAY_FIT contains these methods:

- AIC : Compute Akaike information criterion (AIC).
- BIC : Compute Bayesian information criterion (BIC).
- DIC : Compute Deviance Information Criterion (DIC).

References

Andreas, S. (2012). adaptMCMC: Implementation of a Generic Adaptive Monte Carlo Markov Chain Sampler. <https://cran.r-project.org/web/packages/adaptMCMC/>.

Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H. & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21, pp. 1087-1092.

Plummer, M. Best, N. Cowles, K. & Vines, K. (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News*, 6, pp.7-11. <https://cran.r-project.org/web/packages/coda/>.

Vihola, M. (2012). Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate. *Statistics and Computing*, 22, pp. 997-1008.

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# fit the model on the data with Bayesian estimation
fit = MSGARCH::fit.bayes(spec = spec, y = sp500ret,
                          ctr = list(N.burn = 100, N.mcmc = 1000, N.thin = 1))
```

fit.mle

ML estimation.

Description

Method that performs Maximum Likelihood estimation of a MSGARCH_SPEC object on a set of observations.

Usage

```
fit.mle(spec, y, ctr = list())
```

Arguments

<code>spec</code>	Model specification created with create.spec .
<code>y</code>	Vector (of size T) of observations.
<code>ctr</code>	List of control parameters. The control parameters have two components to it: <ul style="list-style-type: none"> • <code>theta0</code> : Starting parameters (vector of size d). If no starting parameters is provided, the default starting parameters of the specification are used. • <code>do.init</code> : Boolean indicating if there is a pre-optimization with the R package DEoptim (Ardia et al., 2011). (Default: <code>do.init</code> = FALSE) • <code>NP</code> : Number of parameter vectors in the population in DEoptim optimization. (Default: <code>NP</code> = 500) • <code>itermax</code> : Maximum iteration (population generation) allowed in DEoptim optimization. (Default: <code>maxit</code> = 500)

Details

The Maximum likelihood estimation uses the R package `nloptr` (Johnson, 2014) for main optimizer while it uses the R package `DEoptim` when `do.init` = TRUE.

Value

A list of class `MSGARCH_MLE_FIT` containing five components:

- `theta` : Optimal parameters (vector of size d).
- `ll_likelihood` : log-likelihood of y given the optimal parameters.
- `spec` : Specification.
- `is.init` : Indicating if estimation was made with `do.init` option.
- `y` : Vector (of size T) of observations..

The `MSGARCH_MLE_FIT` contains these methods:

- [AIC](#) : Compute Akaike information criterion (AIC).
- [BIC](#) : Compute Bayesian information criterion (BIC).

References

Ardia, D.; Mullen, K. M.; Peterson, B. G. & Ulrich, J. (2015). DEoptim: Differential Evolution in R. <https://cran.r-project.org/web/packages/DEoptim/>.

Johnson, S. G. (2014). The NLOpt Nonlinear-Optimization. <https://cran.r-project.org/web/packages/NLOpt/>.

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)
```

```
# fit the model on the data with ML estimation
fit = MSGARCH::fit.mle(spec = spec, y = sp500ret,
                      ctr = list(do.init = TRUE, NP = 100, itermax = 100))
```

ht

Conditional volatility in each regime.

Description

Method returning the conditional volatility in each regime.

Usage

```
ht(spec, theta, y)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually.

Value

Conditional volatility time serie (array of size $(T + 1) \times M \times K$) in each regime.

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                           do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

# Compute the conditional volatility
ht = MSGARCH::ht(spec = spec, theta = spec$theta0, y = sp500ret)

plot(ht)
```

kernel	<i>Kernel function.</i>
--------	-------------------------

Description

Method returning the kernel value of a vector of observations given a model specification.

Usage

```
kernel(spec, theta, y, log = TRUE)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.
log	Boolean indicating if the log kernel is returned. (Default: log = TRUE)

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually. The kernel is a combination of the prior and the likelihood function. The kernel is equal to $\text{prior}(\theta) + L(y|\theta)$ where L is the likelihood of y given the parameter θ . When doing optimization, the goal is to minimize the negative log-kernel.

- Details on the prior
The prior is different for each specification. It ensures that the θ makes the conditional variance process stationary, positive, and that it respect that the sums of the probabilities in the case of a multiple-regime models are all equal to 1. If any of these three conditions is not respected the prior return $-1e10$, meaning that the optimizer or sampler will know that θ is not a good candidate.

Value

(Log-)Kernel value (scalar or vector of size M) of the vector of observations.

References

Hamilton, J. D. (1989) A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, 57, pp.357-38

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

# compute the kernel
kernel = MSGARCH::kernel(spec = spec, theta = spec$theta0, y = sp500ret, log = TRUE)
```

pdf	<i>Probability density function at $T + 1$.</i>
-----	--

Description

Method returning the probability density at $T + 1$ of a vector of points.

Usage

```
pdf(spec, x, theta, y, log = TRUE)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
x	Vector (of size N) of point to be evaluated
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.
log	Boolean indicating if the log-density is returned. (Default: log = TRUE)

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually. The [pdf](#) method uses the last variance estimate by filtering.

Value

(Log-)Probability density at $T + 1$ of the points x (vector of size N or matrix of size M x N).

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# generate random draws
x = rnorm(100)

# run pdf method on random draws
pdf = MSGARCH::pdf(spec = spec, x = x, theta = spec$theta, y = sp500ret, log = FALSE)
```

pit	<i>Probability Integral Transform at $T + 1$.</i>
-----	--

Description

Method returning the predictive Probability integral transform (PIT) at $T + 1$.

Usage

```
pit(spec, x, theta, y, do.norm = FALSE)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
x	Vector (of size N) of point to be evaluated
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.
do.norm	Boolean indicating if the PIT value are transforms into standard Normal variate. (Default: do.norm = FALSE)

Details

If a matrix of MCMC posterior draws estimates is given, the Bayesian Probability integral transform is calculated. The do.norm argument transforms the PIT value into Normal variate so that normality test can be done.

Value

Probability integral transform at $T + 1$ of the points x or Normal variate derived from the Probability integral transform of x (vector of size N).

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# generate random draws
x = rnorm(100)

# run pit method on random draws
pit = MSGARCH::pit(spec = spec, x = x, theta = spec$theta0, y = sp500ret, do.norm = FALSE)
```

Plast	<i>State probabilities at $T + 1$.</i>
-------	---

Description

Method returning the state probabilities at $T + 1$.

Usage

```
Plast(spec, theta, y)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually.

Value

State probabilities at $T + 1$ (matrix of size M x K).

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH","sGARCH"), distribution = c("norm","norm"),
                             do.skew = c(FALSE,FALSE), do.mix = FALSE, do.shape.ind = FALSE)

# calculate state probabilities at  $T + 1$ 
Plast = MSGARCH::Plast(spec = spec, theta = spec$theta0, y = sp500ret)
```

pred	<i>Predictive density function at $T + 1$.</i>
------	---

Description

Method returning the predictive probability density at $T + 1$ of a vector of points.

Usage

```
pred(spec, x, theta, y, log = TRUE)
```


Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
x	Vector (of size N) of point to be evaluated
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.
log	Boolean indicating if the log-density is returned. (Default: log = TRUE)

Details

If a matrix of MCMC posterior draws estimates is given, the Bayesian predictive density is calculated.

Value

(Log-)Predictive density at $T + 1$ of x (vector of size N).

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# generate random draws
x = rnorm(100)

# run pred method on random draws
pred = MSGARCH::pred(spec = spec, x = x, theta = spec$theta0, y = sp500ret, log = TRUE)
```

Pstate

State probabilities filtering function.

Description

Method returning the filtered state probabilities.

Usage

```
Pstate(spec, theta, y)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually.

Value

Filtered state probabilities of class MSGARCH_RND (array of size $T \times M \times K$). The class MSGARCH_RND contains the plot method.

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

# compute the filtered state probabilities
Pstate = MSGARCH::Pstate(spec = spec, theta = spec$theta0, y = sp500ret)

plot(Pstate)
```

risk

Value-at-Risk And Expected-shortfall functions at $T + 1$.

Description

Method returning the Value-at-Risk and Expected-shortfall at $T + 1$.

Usage

```
risk(spec, theta, y, level = c(0.95, 0.99), ES = TRUE)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
theta	Vector (of size d) or matrix (of size $M \times d$) of parameter estimates.
y	Vector (of size T) of observations.
level	Vector (of size R) of Value-at-risk and Expected-shortfall levels. (Default: level = c(0.95, 0.99))
ES	Boolean indicating if Expected-shortfall is also calculated. (Default: ES = TRUE)

Details

If a matrix of MCMC posterior draws estimates is given, the Bayesian Value-at-Risk and Expected-shortfall are calculated.

Value

A list containing one or two components:

- VaR : Value-at-Risk at $T + 1$ at the choosen levels (vector of size R).
- ES : Expected-shortfall at $T + 1$ at the choosen levels (vector of size R).

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
set.seed(123)

# compute the Value-at-Risk and Expected-shortfall
risk = MSGARCH::risk(spec = spec, theta = spec$theta0, y = sp500ret,
                     level = c(0.95, 0.99), ES = TRUE)
```

rnd	<i>Random draws simulation method at $T + 1$.</i>
-----	--

Description

Method returning random draws at $T + 1$.

Usage

```
rnd(spec, n, theta, y, do.state = TRUE)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
n	Number of random draws to be generated.
theta	Vector (of size d) or matrix (of size $M \times d$) of parameter estimates.
y	Vector (of size T) of observations.
do.state	Boolean indicating if the simulated state are also output. (Default: <code>do.state = TRUE</code>)

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually.

Value

A list of class MSGARCH_RND containing one or two components:

- `draws`: vector (of size n) or matrix (of size $M \times n$) of simulated draws at $T + 1$.
- `state`: vector (of size n) or matrix (of size $M \times n$) of simulated states at $T + 1$. The state value appear only if `do.state = TRUE`.

The MSGARCH_RND class contains the `summary` and `plot` method.

Examples

```
# load data
data("sp500ret")

# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# generate random draws
rnd = MSGARCH::rnd(spec = spec, n = 1000, theta = spec$theta0, y = sp500ret, do.state = TRUE)

plot(rnd)

summary(rnd)
```

sim	<i>Process simulation method.</i>
-----	-----------------------------------

Description

Method returning a simulated process.

Usage

```
sim(spec, n, theta, burnin = 500, do.state = TRUE)
```

Arguments

spec	Model specification of class MSGARCH_SPEC created with create.spec .
n	Simulation length.
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
burnin	(integer >= 0) Burnin period discarded (first simulation draws). (Default: burnin = 500)
do.state	Boolean indicating if the simulated state are also output. (Default: do.state = TRUE)

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually.

Value

A list of class MSGARCH_SIM containing one or two components.

- draws: vector (of size n) or matrix (of size M x n) of simulated draws.
- state: vector (of size n) or matrix (of size M x n) of simulated states. The state value appear only if do.state = TRUE.

The MSGARCH_SIM class contains the plot method.

Examples

```
# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

set.seed(123)

# generate process
sim = MSGARCH::sim(spec = spec, n = 1000, theta = spec$theta0, burnin = 500, do.state = TRUE)

plot(sim)
```

sp500ret

*Standard and poors 500 closing Value log return***Description**

The S&P500 index closing value log return from 1987-03-10 to 2009-01-30 from Yahoo Finance <https://ca.finance.yahoo.com/>.

Usage

```
data("sp500ret")
```

Format

Vector containing 5,523 observations.

Source

Yahoo Finance <https://ca.finance.yahoo.com/>

unc.vol

*Unconditional volatility of each regime.***Description**

Method returning the unconditional volatility of the process in each state.

Usage

```
unc.vol(spec, theta)
```

Arguments

spec Model specification of class MSGARCH_SPEC created with [create.spec](#).
theta Vector (of size d) or matrix (of size M x d) of parameter estimates.

Details

If a matrix of parameter estimates is given, each parameter estimates is evaluated individually.

Value

Unconditional volatility (vector of size K or matrix of size M x K) of each regime.

Examples

```
# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
                             do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)

# compute the unconditional volatility in each regime
unc.vol = MSGARCH::unc.vol(spec = spec, theta = spec$theta0)
```

Index

*Topic **datasets**

sp500ret, [21](#)

AIC, [3](#), [10](#), [11](#)

BIC, [4](#), [10](#), [11](#)

cdf, [5](#), [5](#), [7](#)

create.spec, [5](#), [6](#), [9](#), [11–21](#)

DIC, [8](#), [10](#)

fit.bayes, [3](#), [4](#), [7](#), [8](#), [9](#)

fit.mle, [3](#), [4](#), [7](#), [10](#)

ht, [7](#), [12](#)

kernel, [7](#), [13](#)

MSGARCH (MSGARCH-package), [2](#)

MSGARCH-package, [2](#)

pdf, [7](#), [14](#), [14](#)

pit, [7](#), [15](#)

Plast, [7](#), [16](#)

pred, [7](#), [16](#)

Pstate, [7](#), [17](#)

risk, [7](#), [18](#)

rnd, [7](#), [19](#)

sim, [7](#), [20](#)

sp500ret, [21](#)

unc.vol, [7](#), [21](#)