Package 'MSGARCH'

July 22, 2016

Type Package

Title Markov-Switching GARCH models

Version 0.08

Date 2016-06-11

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,
RcppArmadillo,
adaptMCMC,
DEoptim,
nloptr

LinkingTo Rcpp, RcppArmadillo

Depends RcppArmadillo

Suggests knitr,
rmarkdown

VignetteBuilder knitr

RoxygenNote 5.0.1

R topics documented:

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 provided 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 model. This flexibility can come with problematic since the user must take care when optimizing since the optimizer is not always guaranteed to converge. As option, we provide the f.kernel function which allows the user to develop his own optimization process if none of the optimization options that the package provides is adequate.

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/.

f.aic — *Compute Akaike information criterion (AIC).*

Description

Compute Akaike information criterion (AIC).

Usage

```
f.aic(theta, y, spec)
```

Arguments

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

Value

AIC values (scalar or vector of size M).

References

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

Examples

```
data("sp500ret")

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

aic = spec$f.aic(theta = spec$theta0, y = sp500ret, spec = spec)
```

f.bic — *Compute Bayesian information criterion (BIC).*

Description

Compute Bayesian information criterion (BIC).

Usage

```
f.bic(theta, y, spec)
```

Arguments

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

Value

BIC values (scalar or vector of size M).

References

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

Examples

```
data("sp500ret")

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

bic = spec$f.bic(theta = spec$theta0, y = sp500ret, spec = spec)
```

---

f.cdf *Cumulative density function at T + 1.*

Description

Function inside a specification returning the cumulative density of a vector of points.

Usage

```
f.cdf(x, theta, y, log = TRUE)
```

Arguments

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 `f.cdf` function uses the last variance estimate by filtering.

Value

Cumulative density or log-density of the points x (vector of size N or matrix of size M x N).

Examples

```
data("sp500ret")

spec = MSGARCH::f.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)

x = rnorm(100)

cdf = spec$f.cdf(x = x, theta = spec$theta0, y = sp500ret, log = FALSE)
```

---

f.create.spec *Specification creation*

Description

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

Usage

```
f.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. (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 specification 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 containing functions and variables related to the created specification.
The list contains:

Variables:

- `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 parameter are regime-independent.
- `K` : Number of regimes.
- `sigma0` : Default parameters variance-covariance matrix (of size K x K) used during Bayesian esimation.
- `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 function bound.
- `inequb` : Vector (of size d) of upper inequality function 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 distribuion parameters.
- `do.init` : Boolean indicating the default `do.init` argument.
- `label` : Vector (of size d) of parameters label.
- `name` : Vector (of size K) of specification name.

Functions:

- `f.sim` : Simulation function.
- `f.ht` : Conditional variance in each regime.
- `f.kernel` : Kernel function.
- `f.unc.vol` : Unconditional variance in each regime.
- `f.pred` : Predictive density function.
- `f.pit` : Probability Integral Transform at T + 1.
- `f.risk` : Value-at-Risk And Expected-Shortfall functions.
- `f.rnd` : Simulation function at T + 1.
- `f.pdf` : Probability density function at T + 1.
- `f.cdf` : Cumulative density function at T + 1.
- `f.Pstate` : State probabilities filtering function.
- `f.Plast` : State probabilities at T + 1.

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

Haas, M. Mittnik, S. & Paolella, 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

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

`f.estim.bayes` *Bayesian estimation.*

Description

Function that performs Bayesian estimation of a `MSGARCH` specification on a set of observations.

Usage

```
f.estim.bayes(y, spec, ctr = list())
```

Arguments

`y`	Vector (of size T) of observations.
`spec`	Model specification created with `f.create.spec`.
`ctr`	A list of control parameters. The control parameters have three components: • `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.burn + N.mcmc`. The Bayesian estimation uses the R package `adaptMCMC` (Andreas, 2012) which implements the adaptive sampler of Vihola (2012).

Value

A list containing two variables:

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

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

```
data("sp500ret")

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

theta = spec$f.estim.bay(y = sp500ret, spec = spec,
                         ctr = list(N.burn = 100,N.mcmc = 1000, N.thin = 1))
```

`f.estim.mle` *ML estimation.*

Description

Function that performs Maximum Likelihood estimation of a `MSGARCH` specification on a set of observations.

Usage

```
f.estim.mle(y, spec, ctr = list())
```

Arguments

y	Vector (of size T) of observations.
spec	Model specification created with f.create.spec.
ctr	List of control parameters. The control parameters have two components to it: • theta0 : Starting parameters (vector of size d). If no starting parameters is provided, the default starting parameters of the specification are used. • do.init : Boolean indicating if there is a pre-optimization with the R package DEoptim (Ardia et al., 2011). (default: do.init = FALSE)

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 containing two variables:

- theta : Optimal parameters (vector of size d).
- l_likelihood : log-likelihood of y given the optimal parameters.

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

```
data("sp500ret")

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

theta = MSGARCH::f.estim.mle(y = sp500ret, spec = spec,  ctr = list(do.init = TRUE))
```

---

f.ht *Conditional variance in each regime.*

Description

Function inside a specification returning the conditional variance of each regime.

Usage

```
f.ht(theta, y)
```

Arguments

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

Condititional variance time serie (array of size T + 1 x M x K) for each regime.

Examples

```
data("sp500ret")

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

ht = spec$f.ht(theta = spec$theta0, y = sp500ret)
```

---

f.kernel *Kernel function.*

Description

Function inside a specification returning the kernel value of a vector of observations.

Usage

```
f.kernel(theta, y, log = TRUE)
```

Arguments

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 prior(θ) + L(y|θ) 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

Kernel or 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

```
data("sp500ret")

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

kernel = spec$f.kernel(theta = spec$theta0, y = sp500ret, log = TRUE)
```

---

f.pdf *Probability density function at T + 1.*

Description

Function inside a specification returning the probability density of a vector of points.

Usage

```
f.pdf(x, theta, y, log = TRUE)
```

Arguments

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 f.pdf function uses the last variance estimate by filtering.

Value

Probability density or log-density of the points x (vector of size N or matrix of size M x N).

Examples

```
data("sp500ret")

spec = MSGARCH::f.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)

x = rnorm(100)

pdf = spec$f.pdf(x = x, theta = spec$theta0, y = sp500ret, log = TRUE)
```

`f.pit` *Probability Integral Transform at T + 1.*

Description

Function inside a specification returning the predictive Probability integral transform (PIT).

Usage

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

Arguments

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. (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 of the points x or Normal variate derived from the Probability integral transform of x (vector of size N).

Examples

```
data("sp500ret")

spec = MSGARCH::f.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)

x = rnorm(100)
```

```
pit = spec$f.pit(x = x theta = spec$theta0, y = sp500ret, do.norm = FALSE)
```

`f.Plast` *State probabilities at T + 1.*

Description

Function inside a specification returning the state probabilities at T + 1.

Usage

```
f.Plast(theta, y)
```

Arguments

`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

```
data("sp500ret")

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

Plast = spec$f.Plast(theta = spec$theta0, y = sp500ret)
```

`f.pred` *Predictive density function.*

Description

Function inside a specification returning the predictive probability density of a vector of points.

Usage

```
f.pred(x, theta, y, log = TRUE)
```

Arguments

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

Predictive density or log-density of x (vector of size N).

Examples

```
data("sp500ret")

spec = MSGARCH::f.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)

x = rnorm(100)

pred = spec$f.pred(x = x, theta = spec$theta0, y = sp500ret, log = TRUE)
```

---

f.Pstate *State probabilities filtering function.*

Description

Function inside a specification returning the filtered state probabilities.

Usage

```
f.Pstate(theta, y)
```

Arguments

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 (array of size T x M x K).

Examples

```
data("sp500ret")

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

Pstate  = spec$f.Pstate(theta = spec$theta0, y = sp500ret)
```

---

f.risk *Value-at-Risk And Expected-shortfall functions.*

Description

Function inside a specification returning the Value-at-risk and Expected-shortfall.

Usage

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

Arguments

theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.
level	Vector (of size A) 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 the choosen levels (vector of size A).
- ES : Expected-shortfall at the choosen levels (vector of size A).

Examples

```
data("sp500ret")

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

risk = spec$f.risk(theta = spec$theta0, y = sp500ret,level = c(0.95,0.99), ES = TRUE)
```

f.rnd *Simulation function at T + 1.*

Description

Function inside a specification returning random draws at T + 1.

Usage

```
f.rnd(n, theta, y, do.state = FALSE)
```

Arguments

n	Number of random draws to be generated.
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates.
y	Vector (of size T) of observations.
do.state	Boolean indicating if the simulated state are also output. (default: do.state = FALSE)

Details

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

Value

A list containing one or two components:

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

Examples

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

rnd = spec$f.rnd(n = 1000, theta = theta, y = sp500ret, do.state = TRUE)
```

f.sim *Simulation function.*

Description

Function inside a specification returning a simulated process.

Usage

```
f.sim(n, theta, burnin = 500, do.state = FALSE)
```

Arguments

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: log = TRUE)

Details

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

Value

A list 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.

Examples

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

y = spec$f.sim(n = 1000, theta = spec$theta0, burnin = 500, do.state = TRUE)
```

---

f.unc.vol *Unconditional variance of each regime.*

Description

Function inside a specification returning the unconditional variance of the process in each state.

Usage

```
f.unc.vol(theta)
```

Arguments

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

Examples

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

unc.vol = spec$f.unc.vol(theta = spec$theta0)
```

---

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 5523 observations.

Source

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

Index