Package ‘tvgarch’

April 5, 2024

Type Package
Title Time Varying GARCH Modelling
Version 2.4.2
Date 2024-04-03
Author Susana Campos-Martins [aut, cre], Genaro Sucarrat [ctb]
Maintainer Susana Campos-Martins <scmartins@ucp.pt>
Description Simulation, estimation and inference for univariate and multivariate TV(s)-GARCH(p,q,r)-X models, where s indicates the number and shape of the transition functions, p is the ARCH order, q is the GARCH order, r is the asymmetry order, and 'X' indicates that covariates can be included; see Campos-Martins and Sucarrat (2024) <doi:10.18637/jss.v108.i09>. In the multivariate case, variances are estimated equation by equation and dynamic conditional correlations are allowed. The TV long-term component of the variance as in the multiplicative TV-GARCH model of Amado and Terasvirta (2013) <doi:10.1016/j.jeconom.2013.03.006> introduces non-stationarity whereas the GARCH-X short-term component describes conditional heteroscedasticity. Maximisation by parts leads to consistent and asymptotically normal estimates.

License GPL (>= 2)
Depends R (>= 3.5.0), garchx, zoo, numDeriv

URL https://sites.google.com/site/susanacamposmartins
NeedsCompilation no
Repository CRAN
Date/Publication 2024-04-04 23:30:02 UTC

R topics documented:
tvgarch-package .................................................. 2
coeff.mtvgarch .................................................. 3
coeff.tvgarch .................................................. 6
coeff.tvgarchTest .............................................. 8
combos ......................................................... 11
dccObj .......................................................... 12
Description

Simulation, estimation and inference for univariate and multivariate TV(s)-GARCH(p,q,r)-X models, where s indicates the number and shape of the transition functions, p is the ARCH order, q is the GARCH order, r is the asymmetry order, and 'X' indicates that covariates can be included; see Campos-Martins and Sucarrat (2024) <doi:10.18637/jss.v108.i09>. The TV long-term component, as in the multiplicative TV-GARCH model of Amado and Terasvirta (2013) <doi:10.1016/j.jeconom.2013.03.006>, introduces non-stationarity whereas the GARCH-X short-term component describes conditional heteroscedasticity. Maximisation by parts leads to consistent and asymptotically normal estimates. In the multivariate case, conditional variances are estimated equation by equation and dynamic conditional correlations are allowed.

Details

Package: tvgarch
Type: Package
Version: 2.4.2
Date: 2024-04-03
License: GPL>=2

Author(s)

Susana Campos-Martins, https://sites.google.com/site/susanacamposmartins

Maintainer: Susana Campos-Martins
Contributor: Genaro Sucarrat

References


See Also

tvgarchTest, tvgarch, mtvgarch, tvgarchSim, mtvgarchSim

Examples

set.seed(123)

## Simulate from a TV(1)-GARCH(1,1) model (default):
ySim <- tvgarchSim(n = 1500)

## Test a GARCH(1,1) model against a TV(1)-GARCH(1,1) model:
yTest <- tvgarchTest(y = ySim)
yTest

## Estimate a TV(1)-GARCH(1,1) model (default):
yEst <- tvgarch(y = ySim)
yEst

coef.mtvgarch

Extraction functions for multivariate 'mtvgarch' objects

Description

Extraction functions for objects of class 'mtvgarch'.

Usage

## S3 method for class 'mtvgarch'
coef(object, spec = c("tvgarch", "garch", "tv", "cc"), ...)

## S3 method for class 'mtvgarch'
fitted(object, spec = c("tvgarch", "garch", "tv", "cc"),
as.zoo = TRUE, ...)

## S3 method for class 'mtvgarch'
logLik(object, ...)

## S3 method for class 'mtvgarch'
nobs(object, ...)

## S3 method for class 'mtvgarch'
plot(x, spec = c("tvgarch", "garch", "tv"), ...)

## S3 method for class 'mtvgarch'
predict(object, n.ahead = 10, newxtv = NULL,
          newxreg = NULL, newindex = NULL, n.sim = 5000,
as.zoo = TRUE, verbose = FALSE, ...)

## S3 method for class 'mtvgarch'
print(x, ...)
## S3 method for class 'mtvgarch'
quantile(x, probs = 0.025, type = 7, as.zoo = TRUE, ...)
## S3 method for class 'mtvgarch'
residuals(object, as.zoo = TRUE, ...)
## S3 method for class 'mtvgarch'
summary(object, ...)
## S3 method for class 'mtvgarch'
toLatex(object, digits = 4, ...)
## S3 method for class 'mtvgarch'
vcov(object, spec = c("tvgarch", "garch", "tv", "cc"), ...)

### Arguments

- `object`: an object of class 'mtvgarch'.
- `spec`: specifies whether the function should extract specific results. If "tv", extracts results for the TV component and if "garch" extracts results for the GARCH-X component of TV-GARCH-X model. If "tvgarch", extracts results for TV-GARCH-X model. Only relevant for TV-GARCH-X models. Otherwise, extracts results for GARCH-X models. If "cc", extracts results concerning the conditional correlations. "cc" is not valid in plot.mtvgarch().
- `x`: an object of class 'mtvgarch'.
- `as.zoo`: logical. If TRUE, then the returned result is of class zoo.
- `n.ahead`: integer that determines how many steps ahead predictions should be generated.
- `newxtv`: NULL or vector with the out-of-sample transition variable. If NULL, out-of-sample component g equals intercept.g. Only relevant for TV-GARCH-X models.
- `newxreg`: vector or matrix with the out-of-sample regressor values.
- `newindex`: zoo-index for the out-of-sample predictions. If NULL (default), then 1:n.ahead is used.
- `n.sim`: integer, the number of simulations.
- `verbose`: logical. If TRUE, then the simulations - in addition to the predictions - are returned.
- `probs`: vector of probabilities.
- `type`: integer that determines the algorithm used to compute the quantile, see quantile.
- `digits`: integer, the number of digits in the printed LaTeX code.
- `...`: additional arguments.

### Value

- `coef`: parameter estimates.
- `fitted`: fitted conditional variances and correlations.
- `logLik`: optimised log-likelihood (normal density) values.
- `nobs`: number of observations used in the estimation.
- `plot`: plots of the fitted conditional volatilities.
predict: variance predictions. Column order differs when spillovers are allowed.
print: print of the estimation results.
quantile: fitted quantiles, i.e. the conditional standard deviation times the empirical quantile of the standardised innovations.
residuals: volatility standardised residuals.
summary: summary of estimation results.
vcov: coefficient variance-covariance matrices.

Author(s)
Susana Campos-Martins

References

See Also
mtvgarch, mtvgarchSim, tvgarch, garchx, zoo

Examples
set.seed(12345)

## Simulate from a bivariate CCC-TV(1)-GARCH(1,1) model (default):
mySim <- mtvgarchSim(n = 1500)

## Estimate a CCC-TV(1)-GARCH(1,1) model:
myEst <- mtvgarch(y = mySim)

## Print estimation results:
print(myEst)

## Extract and store conditional variances:
sigma2Est <- fitted(myEst)

## Plot:
plot(myEst)

## Generate predictions:
predict(myEst)
 coef.tvgarch

## S3 method for class 'tvgarch'
coef(object, spec = c("tvgarch", "garch", "tv"), ...)
## S3 method for class 'tvgarch'
fitted(object, spec = c("tvgarch", "garch", "tv"),
as.zoo = TRUE, ...)
## S3 method for class 'tvgarch'
logLik(object, ...)
## S3 method for class 'tvgarch'
nobs(object, ...)
## S3 method for class 'tvgarch'
plot(x, spec = c("tvgarch", "garch", "tv"), ...)
## S3 method for class 'tvgarch'
predict(object, n.ahead = 10, newxtv = NULL,
        newxreg = NULL, newindex = NULL, n.sim = 5000,
        as.zoo = TRUE, verbose = FALSE, ...)
## S3 method for class 'tvgarch'
print(x, ...)
## S3 method for class 'tvgarch'
quantile(x, probs = 0.025, names = TRUE, type = 7,
         as.zoo = TRUE, ...)
## S3 method for class 'tvgarch'
residuals(object, as.zoo = TRUE, ...)
## S3 method for class 'tvgarch'
summary(object, ...)
## S3 method for class 'tvgarch'
toLatex(object, digits = 4, ...)
## S3 method for class 'tvgarch'
vcov(object, spec = c("tvgarch", "garch", "tv"), ...)

Arguments

object

an object of class 'tvgarch'.
spec

specifies whether the function should extract specific results. If "tv", extracts results for the TV component and if "garch" extracts results for the GARCH-X component of TV-GARCH-X model. If "tvgarch", extracts results for TV-GARCH-X model. Only relevant for TV-GARCH-X models. Otherwise, extracts results for GARCH-X models.
coef.tvrgarch

x

as.zoo

n.ahead

newxtv

newxreg

newindex

n.sim

verbose

probs

names

type

digits

Value

c coef: parameter estimates.
f fitted: fitted conditional variance.
logLik: optimised log-likelihood (normal density) value.
nobs: the number of observations used in the estimation.
plot: plot of the fitted conditional volatility.
predict: variance predictions.
print: print of the estimation results.
quantile: fitted quantiles, i.e. the conditional standard deviation times the empirical quantile of the standardised innovations.
residuals: volatility standardised residuals.
summary: summary of estimation results.
vcov: coefficient variance-covariance matrix.

Author(s)

Susana Campos-Martins

References

See Also
tvgarchTest, tvgarch, tvgarchSim, zoo

Examples

set.seed(123)

## Simulate from a TV(1)-GARCH(1,1) model (default):
ySim <- tvgarchSim(n = 1500)

## Estimate a TV(1)-GARCH(1,1) model:
yEst <- tvgarch(y = ySim)

## Print estimation results:
print(yEst)

## Extract and store conditional variances:
sigma2Est <- fitted(yEst)

## Plot:
plot(yEst)

## Generate predictions:
predict(yEst)

c coef.tvgarchTest

Extraction functions for univariate 'tvgarchTest' objects

Description

Extraction functions for objects of class 'tvgarchTest'. Results from the estimation of the model under the null hypothesis, i.e., a GARCH(1,1) model, can be extracted similar to an object of class 'tvgarch' with the exception of functions print.tvgarchTest() and summary.tvgarchTest().

Usage

## S3 method for class 'tvgarchTest'
c coef(object, ...)
## S3 method for class 'tvgarchTest'
fitted(object,
as.zoo = TRUE, ...)
## S3 method for class 'tvgarchTest'
logLik(object, ...)
## S3 method for class 'tvgarchTest'
nobs(object, ...)
## S3 method for class 'tvgarchTest'
predict(yEst)
## S3 method for class 'tvgarchTest'
Arguments

object an object of class 'tvgarchTest'.
x an object of class 'tvgarchTest'.
as.zoo logical. If TRUE, then the returned result is of class zoo.
n.ahead integer that determines how many steps ahead predictions should be generated.
newxreg vector or matrix with the out-of-sample regressor values.
newindex a zoo-index for the out-of-sample predictions. If NULL (default), then 1:n.ahead is used.
n.sim integer, the number of simulations.
verbose logical. If TRUE, then the simulations - in addition to the predictions - are returned.
probs vector of probabilities.
names logical, whether to return names or not.
type integer that determines the algorithm used to compute the quantile, see quantile.
digits integer, the number of digits in the printed LaTeX code.
... additional arguments.

Value

coef: parameter estimates.
fitted: fitted conditional variance.
logLik: optimised log-likelihood (normal density) value.
nobs: the number of observations used in the estimation.
plot: plot of the fitted conditional volatility.
predict: variance predictions.
quantile: fitted quantiles, i.e. the conditional standard deviation times the empirical quantile of the standardised innovations.
residuals: volatility standardised residuals.
summary: summary of test result.
vcov: coefficient variance-covariance matrix.

Author(s)
Susana Campos-Martins

References

See Also
tvgarchTest, tvgarchSim, tvgarch, zoo

Examples
set.seed(123)

## Simulate from a TV(1)-GARCH(1,1) model (default):
ySim <- tvgarchSim(n = 1500)

## Test a GARCH(1,1) model against a TV(1)-GARCH(1,1) model:
yTest <- tvgarchTest(y = ySim)

## Print test and estimation results:
print(yTest)

## Estimated number of locations
summary(yTest)

## Extract and plot estimation results for GARCH(1,1) used in the test:
sigma2Test <- fitted(yTest)
plot(yTest)

## Estimate a TV(s)-GARCH(1,1) model:
s <- summary(yTest)
yEst <- tvgarch(y = ySim, order.g = s)
**combos**

*Compute all combinations of a hierarchy of models of n variables, and enumerate the combinations of the elements of a vector.*

**Description**

combos produces a matrix of combinations of 1 to n variables in ascending order. combinations enumerates the possible combinations of a specified size from the elements of a vector.

**Usage**

```r
combos(n)
combinations(n, r, v = 1:n, set = TRUE, repeats.allowed = FALSE)
```

**Arguments**

- `n`: an integer: the number of variables (combos) or the size of the source vector (combinations)
- `r`: size of the target vectors
- `v`: source vector. Defaults to 1:n
- `set`: logical flag indicating whether duplicates should be removed from the source vector v. Defaults to TRUE.
- `repeats.allowed`: logical flag indicating whether the constructed vectors may include duplicated values. Defaults to FALSE.

**Details**

combos lists hierarchy of all possible combinations of n variables in ascending order, starting with 1 variable, then all combinations of 2 variables, and so on until the one combination with all n variables. It is used by function tvgarch to constrain the size coefficients when s > 1 required to guarantee the variance is positive for all t.

When using combinations, the number of combinations increases rapidly with n and r! To use values of n above about 45, you will need to increase R’s recursion limit. See the expression argument to the options command for details on how to do this. The source code is adapted from the function with the same name in the package gtools. There, it is stated that the code of the function is from an email by Brian D Ripley <ripley@stats.ox.ac.uk> to r-help dated Tue, 14 Dec 1999 11:14:04 +0000 (GMT) in response to Alex Ahgarin <datamanagement@email.com>. Original version was named "subsets" and was Written by Bill Venables

**Value**

- `combos`: a matrix with zeroes in empty elements and 1 in all full elements.
- `combinations`: a matrix where each row contains a vector of length r.
Author(s)
combos by Chris Walsh <cwalsh@unimelb.edu.au>, with modifications by Susana Campos-Martins. Original versions of combinations by Bill Venables <Bill.Venables@cmis.csiro.au>. Extended to handle repeats.allowed by Gregory R. Warnes <greg@warnes.net>.

References

See Also
tvgarch

Examples

combos(3)

combinations(3,2,letters[1:3])
combinations(3,2,letters[1:3],repeats=TRUE)

dccObj

Auxiliary functions

Description
Auxiliary functions used in the estimation of the multivariate TV(s)-GARCH(p,q,r)-X model. Not intended for the average user.

Usage
dccObj(par.dcc, z, sigma2, flag)

Arguments
par.dcc numeric vector containing the ARCH- and GARCH-type coefficients in the dynamic conditional correlations.
z matrix of standardized residuals.
sigma2 matrix of conditional variances.
flag integer. If 0, returns a numeric vector with the values of the objective function; if 1 returns the the value of the objective function; if 2, returns the fitted variance components.

Value
The values of the objective function or fitted dynamic conditional correlations.
**mtvgarch**

**Author(s)**

Susana Campos-Martins

**References**


**See Also**

mtvgarch, fitted.mtvgarch, residuals.mtvgarch

---

**mtvgarch**

*Estimate a multivariate TV-GARCH-X model*

**Description**

Equation by equation estimation of a multivariate multiplicative TV-GARCH-X model with dynamic conditional correlations. For each variance equation, the long-term or unconditional component (TV) and the short-term or conditional variance component (GARCH-X) are estimated separately using maximization by parts, where the iterative algorithm proceeds until convergence. Conditional on the variance estimates, the dynamic conditional correlations are estimated by maximum likelihood.

**Usage**

mtvgarch(y, order.g = c(1, 1), order.h = NULL, order.x = NULL, initial.values = list(), xtv = NULL, xreg = NULL, opt = 2, upper.speed = NULL, tvgarch = FALSE, dcc = FALSE, turbo = TRUE, trace = FALSE)

**Arguments**

- **y**
  - numeric matrix, time series or zoo object.
- **order.g**
  - integer matrix with each row indicating the order.g for each series; number of locations in each transition function of the TV components.
- **order.h**
  - integer matrix with each row indicating the order.h for each series; the first column controls the GARCH order, the second the ARCH order and the third the asymmetry order of the GARCH-X components. If NULL, the default, all series are assumed to follow a GARCH(1,1,0).
- **order.x**
  - NULL or binary matrix indicating which xreg variables should be included as covariates in the GARCH-X components. If provided and xreg is NULL, then the selected volatility spillovers are included as covariates.
initial.values: a list containing the initial parameter values passed on to the optimisation routines (constrOptim for the TV component and nlminb for the GARCH-X component). If list(), the default, then the values are chosen automatically. TV component: intercept.g - NULL or numeric vector, size - NULL or numeric matrix containing the size initial coefficients, speed - NULL or numeric matrix containing the speed initial coefficients, location - NULL or numeric matrix containing the location initial coefficients. GARCH-X component: intercept.h - numeric vector, arch - NULL or numeric matrix containing the ARCH initial coefficients, garch - NULL or numeric matrix containing the GARCH-type initial coefficients, asym - NULL or numeric matrix containing the asymmetry-type initial coefficients, par.xreg - NULL or numeric matrix containing the X-type initial coefficients, and R - initial correlation coefficients.

xtv: NULL or numeric vector, time series or zoo object to include as the transition variable in the TV component. If NULL, calendar time, scaled between 0 and 1, is used as the transition variable.

xreg: numeric vector, time series or zoo object to include as covariates in the GARCH-X component.

opt: integer indicating whether the speed parameter in the TV component should be scaled. If 0, no scaling; if 1, speed/sd(xtv); if 2, exp(speed).

upper.speed: NULL or numeric scalar that sets the upper bound for speed in each transition function. If NULL, the default, the upper bound is 10000 for all transition functions. If numeric scalar, upper.speed is used for all transition functions.

tvgarch: logical. If TRUE, the full parameter set is estimated in one final step as well as the standard errors. If FALSE (default), estimates from last iteration are reported instead.

dcc: logical. If TRUE, dynamic conditional correlations are estimated. If FALSE (default), then the conditional correlations are constant.

turbo: logical. If FALSE (default), then the coefficient variance-covariance is computed during estimation, and the fitted values and residuals are attached to the returned object. If TRUE, then these operations are skipped, and hence estimation is faster. Note, however, that if turbo is set to TRUE, then the coefficient-covariance, fitted values and residuals can still be extracted subsequent to estimation with vcov.mtvgarch(), fitted.mtvgarch() and residuals.mtvgarch(), respectively.

trace: logical. If TRUE all output is printed.

Value
An object of class ‘mtvgarch’.

Author(s)
Susana Campos-Martins

References


See Also
tvgarch, garchx, nlminb, constrOptim

Examples

```r
set.seed(12345)

## Simulate from a bivariate CCC-TV(1)-GARCH(1,1) model (default):
mySim <- mtvgarchSim(n = 1000)

## Estimate a CCC-TV(1)-GARCH(1,1) model (default):
myEst <- mtvgarch(y = mySim)

## Print estimation results:
print(myEst)

## Extract coefficients:
coef(myEst)

## Plot conditional volatilities:
plot(myEst)

## Generate predictions:
predict(myEst)
```

mtvgarchSim

Simulate from a multivariate TV-GARCH-X model

Description

Simulate from a multivariate multiplicative TV(s)-GARCH(p,q,r)-X model.

Usage

```r
mtvgarchSim(n, m = 2, order.g = c(1,1), order.h = c(1,1,0, 1,1,0), order.x = NULL, intercept.g = c(1.2,1), size = c(3,5), speed = c(10,25), location = c(0.5,0.8), intercept.h = c(0.2,0.3), arch = c(0.10,0.05), garch = c(0.80,0.90), asym = NULL, xtv = NULL, xreg = NULL, par.xreg = NULL, R = c(1,0.6,0.6,1), dcc = FALSE, par.dcc = NULL, opt = 0, as.zoo = TRUE, verbose = FALSE, innovations = NULL)
```
Arguments

Arguments:

- `n` integer.
- `m` integer indicating the dimension of the multivariate series.
- `order.g` integer matrix with each row indicating the number of locations in each transition function of the TV components; m rows and max.s columns.
- `order.h` integer matrix with each row indicating the `order.h` for each series; the first column controls the GARCH order, the second the ARCH order and the third the asymmetry order of the GARCH-X components.
- `order.x` NULL or binary matrix indicating which `xreg` variables should be included as covariates in the GARCH-X components. If provided and `xreg` is NULL, volatility spillovers for the selected series are included as covariates.
- `intercept.g` NULL or numeric vector.
- `size` NULL or numeric matrix containing the size coefficients. Only relevant for TV-GARCH models.
- `speed` NULL or numeric matrix containing the speed coefficients. Only relevant for TV-GARCH models.
- `location` NULL or numeric matrix containing the location coefficients; m rows and max.c columns. Only relevant for TV-GARCH models.
- `intercept.h` numeric matrix.
- `arch` NULL or numeric matrix containing the ARCH coefficients.
- `garch` NULL or numeric matrix containing the GARCH-type coefficients.
- `asym` NULL or numeric matrix containing the asymmetry-type coefficients.
- `xtv` NULL or numeric vector, time series or zoo object to include as the transition variable in the TV component. If NULL, calendar time, scaled between 0 and 1, is used as the transition variable. Only relevant for TV-GARCH models.
- `xreg` numeric vector, matrix, time series or zoo object to include as covariates in the GARCH-X component.
- `par.xreg` NULL or numeric matrix containing the covariates initial coefficients.
- `R` matrix of (constant) conditional correlations.
- `dcc` logical. If TRUE, dynamic conditional correlations are estimated. If FALSE (default), then the conditional correlations are constant.
- `par.dcc` numeric vector containing the ARCH- and GARCH-type coefficients in the dynamic conditional correlations.
- `opt` integer indicating whether the speed parameter in the TV component should be scaled. If 0, no scaling; if 1, speed/sd(xtv); if 2, exp(speed). Only relevant for TV-GARCH models.
- `as.zoo` logical. If TRUE, then the returned result is of class zoo.
- `verbose` logical, if TRUE, the conditional variance and the innovations are also returned.
- `innovations` NULL or numeric matrix with the innovations. If NULL, then standard normal innovations are generated with `rnorm`.
Value

An object of class 'zoo' (if as.zoo = TRUE), otherwise a matrix or a list (if verbose = TRUE), with the simulated values.

Author(s)

Susana Campos-Martins

See Also

mtvgarch, tvgarch, garchx, zoo

Examples

set.seed(12345)

## Simulate from a bivariate CCC-TV(1)-GARCH(1,1) model (default): mySim1 <- mtvgarchSim(n = 1500)

## Simulate from a bivariate CCC-TV(1)-GARCH(1,1)-X model
## (with volatility spillovers)
mySim2 <- mtvgarchSim(n = 1500, order.x = c(0,1,1,0), par.xreg = c(0.03,0.04))

Estimate a TV-GARCH-X model

Description

Quasi Maximum Likelihood (ML) estimation of a univariate multiplicative TV(s)-GARCH(p,q,r)-X model, where s indicates the number and the shape of the transition functions, r is the asymmetry order, p is the ARCH order, q is the GARCH order, and 'X' indicates that covariates can be included. Any transition variable, deterministic or stochastic, can be used to drive the transitions between the variance states. The TV long-term component introduces non-stationarity in the variance process, where the GARCH-X short-term component describes conditional heteroscedasticity. Maximization by parts leads to consistent and asymptotically normal estimates.

Usage

tvgarch(y, order.g = 1, order.h = c(1,1,0), xtv = NULL, xreg = NULL, initial.values = list(), opt = 2, upper.speed = NULL, tvgarch = FALSE, turbo = FALSE, trace = FALSE)
Arguments

- **y**
  numeric vector, time series or zoo object.

- **order.g**
  integer vector of length s indicating the number of locations in each transition function of the TV component. Indicates whether a stationary GARCH or a nonstationary GARCH, i.e., TV-GARCH, shall be estimated.

- **order.h**
  integer vector of the form \(c(p,q,r)\). The first entry controls the GARCH order, the second the ARCH order and the third the asymmetry order of the GARCH-X component.

- **initial.values**
  a list containing the initial parameter values passed on to the optimisation routines (constrOptim for the TV component and nlminb for the GARCH-X component). If list(), the default, then the values are chosen automatically. TV component: \(intercept.g - NULL\) or numeric, \(size - NULL\) or numeric vector containing the size initial coefficients, \(speed - NULL\) or numeric vector containing the speed initial coefficients, \(location - NULL\) or numeric vector containing the location initial coefficients. GARCH-X or GARCH-X component of TV-GARCH-X: \(intercept.h\) - numeric, \(arch\) - NULL or numeric vector containing the ARCH initial coefficients, \(garch\) - NULL or numeric vector containing the GARCH-type initial coefficients, \(asym\) - NULL or numeric vector containing the asymmetry-type initial coefficients, and \(par.xreg\) - NULL or numeric vector containing the X-type initial coefficients.

- **xtv**
  NULL or numeric vector, time series or zoo object to include as the transition variable in the TV component. If NULL, calendar time, scaled between 0 and 1, is used as the transition variable. Not relevant for stationary GARCH.

- **xreg**
  numeric vector, time series or zoo object to include as covariates in the GARCH-X component.

- **opt**
  integer indicating whether the speed parameter in the TV component should be scaled. If 0, no scaling; if 1, \(speed/sd(xtv)\); if 2, \(exp(speed)\). Only relevant for TV-GARCH models.

- **upper.speed**
  NULL or numeric scalar that sets the upper bound for speed in each transition function. If NULL, the default, the upper bound is 10000 for all transition functions. If numeric scalar, \(upper.speed\) is used for all transition functions.

- **tvgarch**
  logical. If TRUE, the full parameter set is estimated in one final step as well as the standard errors. If FALSE (default), estimates from last iteration are reported instead.

- **turbo**
  logical. If FALSE (default), then the coefficient variance-covariance is computed during estimation, and the fitted values and residuals are attached to the returned object. If TRUE, then these operations are skipped, and hence estimation is faster. Note, however, that if turbo is set to TRUE, then the coefficient-covariance, fitted values and residuals can still be extracted subsequent to estimation with vcov.tvgarch(), fitted.tvgarch() and residuals.tvgarch(), respectively.

- **trace**
  logical. If TRUE all output is printed when estimating a TV-GARCH.

Value

An object of class 'tvgarch'.
Author(s)
Susana Campos-Martins

References

See Also
garchx, tvgarchSim, nlminb, constrOptim

Examples
set.seed(123)

## Simulate from a TV(1)-GARCH(1,1) model (default):
ySim <- tvgarchSim(n = 1500)

## Estimate a TV(1)-GARCH(1,1) model:
yEst <- tvgarch(y = ySim)

## Print estimation results:
print(yEst)

## Extract coefficients:
coef(yEst)

## Plot conditional volatilities:
plot(yEst)

## Extract log-likelihood:
logLik(yEst)

## Extract and store standardised residuals:
etaEst <- residuals(yEst)

## Generate predictions:
predict(yEst)
Auxiliary functions

Description

Auxiliary functions used in the estimation of the univariate and multivariate TV(s)-GARCH(p.q.r)-X model. Not intended for the average user.

Usage

```r
tv(speed, location, xtv = NULL, n = NULL, opt = 0,
   order.g = NULL, as.zoo = TRUE, verbose = FALSE)
tvObj(par.g, fixed.par.g, xtv, opt, order.g, fixed.h, y, iter0, flag)
garchObj(par.h, xreg, order.h, fixed.g, y, flag)
tvgarchObj(par, fixed.par.g, y, order.g, xtv, opt, iter.fit.h, flag)
```

Arguments

- **speed**: NULL or numeric vector with the values of the speed coefficients.
- **location**: NULL or numeric vector with the values of the location coefficients.
- **xtv**: NULL or numeric vector, time series or zoo object to include as the transition variable in the TV component. If NULL, a continuous variable bounded between 0 and 1 for n observations is constructed and used as the transition variable.
- **n**: integer indicating the number of observations of the continuous transition variable bounded between 0 and 1.
- **opt**: integer, indicates whether the speed parameter in the TV component should be scaled. If 0, no scaling; if 1, speed/sd(xtv); if 2, exp(speed). For function tv(), the default is 0.
- **order.g**: a scalar in tv() and an integer vector of length s in tvObj() indicating the number of locations in each transition function of the TV component. For function tv(), it defaults to NULL.
- **as.zoo**: logical. If TRUE, then the returned result is of class zoo.
- **verbose**: logical, if TRUE, the values of not only the logistic transition function but also the transition variable are returned.
- **par.g**: numeric vector with the values of the parameters in the TV component. If iter0=TRUE, par.g takes the form c(intercept.g, size, speed, location); if iter0=FALSE, then par.g=c(speed, size, location) and the values of the fixed parameters are provided using fixed.par.g).
- **fixed.par.g**: NULL or numeric vector with the values of the parameters fixed in the TV component, i.e., intercept.g.
- **par**: NULL or numeric vector with the values of the parameters in the TV-GARCH-X model.
- **fixed.h**: numeric vector, time series or zoo containing the values of GARCH-X component).
tvgarchSim

Simulate from a univariate TV-GARCH-X model

**Description**

Simulate from a univariate multiplicative TV(s)-GARCH(p,q,r)-X model.

**Usage**

tvgarchSim(n, order.g = 1, order.h = c(1,1,0),
    intercept.g = 1.2, size = 5, speed = 25, location = 0.5, xtv = NULL,
    intercept.h = 0.2, arch = 0.1, garch = 0.8, asym = NULL, xreg = NULL,
    opt = 0, as.zoo = TRUE, verbose = FALSE, innovations = NULL)
Arguments

- **n**: integer.
- **order.g**: integer vector of length s indicating the number of locations in each transition function of the TV component.
- **order.h**: integer vector of the form c(p,q,r). The first entry controls the GARCH order, the second the ARCH order and the third the asymmetry order of the GARCH-X component.
- **intercept.g**: NULL or numeric with the value of the intercept in the TV component.
- **size**: NULL or numeric vector with the values of the size coefficients.
- **speed**: NULL or numeric vector with the values of the speed coefficients.
- **location**: NULL or numeric vector with the values of the location coefficients.
- **xtv**: NULL or numeric vector, time series or zoo object to include as the transition variable in the TV component. If NULL, calendar time, scaled between 0 and 1, is used as the transition variable.
- **opt**: integer indicating whether the speed parameter in the TV component should be scaled. If 0, no scaling; if 1, speed/sd(xtv); if 2, exp(speed).
- **intercept.h**: numeric with the value of the intercept in the GARCH-X component.
- **arch**: NULL or numeric vector with the values of the ARCH-coefficients.
- **garch**: NULL or numeric vector with the values of the GARCH-coefficients.
- **asym**: NULL or numeric vector with the values of the asymmetry-coefficients.
- **xreg**: NULL or numeric vector with the values of the X-term.
- **as.zoo**: logical. If TRUE, then the returned result is of class zoo.
- **verbose**: logical, if TRUE, the conditional variance and innovations are also returned.
- **innovations**: NULL or numeric vector with the innovations. If NULL, then standard normal innovations are generated with rnorm.

Value

An object of class ‘zoo’ (if as.zoo = TRUE), otherwise a vector or a matrix (if verbose = TRUE), with the simulated values.

Author(s)

Susana Campos-Martins

See Also

tvgarch, garchx, zoo
Examples

set.seed(123)

## Simulate from a TV(1)-GARCH(1,1) model (default):
ySim1 <- tvgarchSim(n = 1500)

## Simulate from a TV(2)-GARCH(1,1) model:
ySim2 <- tvgarchSim(n = 1500, order.g = c(1,2), size = c(0.5,-0.4),
speed = c(1.5,2), location = c(0.2, 0.5,0.8))

## Simulate from a GARCH(1,1) model:
ySim3 <- tvgarchSim(n = 1500, order.g = NULL)

## Simulate from a TV(1)-GARCH(1,1,1)-X model:
ySim4 <- tvgarchSim(n = 1500, order.h = c(1,1,1), asym = 0.025, xreg = ySim3^2)

---

**tvgarchTest**  
*Test of a multiplicative time-varying GARCH model*

**Description**

Compute the non-robust and robust Lagrange-Multiplier (LM)-type test statistics for examining the null hypothesis of constant long-term variance, GARCH(1,1), against the alternative of a smoothly changing long-term component, TV-GARCH(1,1).

**Usage**

tvgarchTest(y, xtv = NULL, alpha = 0.05)

**Arguments**

- **y**
  - numeric vector, time series or *zoo* object.
- **xtv**
  - NULL or numeric vector, time series or zoo object to include as the transition variable in the TV component. If NULL, calendar time scaled between 0 and 1 is used as the transition variable.
- **alpha**
  - the significance level.

**Value**

An object of class 'tvgarchTest'.

**Author(s)**

Susana Campos-Martins

**References**

See Also
tvgarch, garch, tvgarchSim

Examples

```r
set.seed(12345)

## Simulate from a TV(1)-GARCH(1,1) model (default):
ySim <- tvgarchSim(n = 1500)

## Test of a TV(1)-GARCH(1,1) model:
yTest <- tvgarchTest(y = ySim)
orderG1 <- summary(yTest)

## Estimate a TV(1)-GARCH(1,1) model:
yEst <- tvgarch(y = ySim, order.g = orderG1)
```
Index

* Econometrics
  coef.mtvgarch, 3
c of. tvgarch, 6
c of. tvgarchTest, 8
dccObj, 12
tvgarchSim, 15
tvgarch, 17
tvgarch-package, 2
tvgarchObj, 20
tvgarchSim, 21
tvgarchTest, 23

* Financial Econometrics
  coef.mtvgarch, 3
c of. tvgarch, 6
c of. tvgarchTest, 8
dccObj, 12
tvgarch, 13
tvgarchSim, 15
tvgarch, 17
tvgarch-package, 2
tvgarchObj, 20
tvgarchSim, 21
tvgarchTest, 23

* Nonlinear Time Series
  coef.mtvgarch, 3
c of. tvgarch, 6
c of. tvgarchTest, 8
dccObj, 12
tvgarch, 13
tvgarchSim, 15
tvgarch, 17
tvgarch-package, 2
tvgarchObj, 20
tvgarchSim, 21
tvgarchTest, 23

* Spillovers
  mtvgarch, 13

* Time Varying Parameter Models
  coef.mtvgarch, 3
c of. tvgarch, 6
c of. tvgarchTest, 8
dccObj, 12
mtvgarch, 13
mtvgarchSim, 15
tvgarch, 17
tvgarch-package, 2
tvgarchObj, 20
tvgarchSim, 21
tvgarchTest, 23

co of. mtvgarch, 3
c of. tvgarch, 6
c of. tvgarchTest, 8
dccObj, 12
mtvgarch, 13
tvgarchSim, 15
tvgarch, 17
tvgarch-package, 2
tvgarchObj, 20
tvgarchSim, 21
tvgarchTest, 23

combinations (combs), 11
combs, 11
constrOptim, 14, 15, 18, 19
dccObj, 12
fitted.mtvgarch, 13
fitted.mtvgarch (coef.mtvgarch), 3
fitted.tvgarch, 21
fitted.tvgarch (coef.tvgarch), 6
fitted.tvgarchTest (coef.tvgarchTest), 8
garchObj (tvgarchObj), 20
garchx, 5, 15, 17, 19, 22, 24
logLik.mtvgarch (coef.mtvgarch), 3
logLik.tvgarch (coef.tvgarch), 6
logLik.tvgarchTest (coef.tvgarchTest), 8
mtvgarch, 3, 5, 13, 13, 17
mtvgarchSim, 3, 5, 15

nlminb, 14, 15, 18, 19
nobs.mtvgarch (coef.mtvgarch), 3
nobs.tvgarch (coef.tvgarch), 6
nobs.tvgarchTest (coef.tvgarchTest), 8
INDEX

plot.mtvgarch (coef.mtvgarch), 3
plot.tvgarch (coef.tvgarch), 6
plot.tvgarchTest (coef.tvgarchTest), 8
predict.mtvgarch (coef.mtvgarch), 3
predict.tvgarch (coef.tvgarch), 6
predict.tvgarchTest (coef.tvgarchTest), 8
print.mtvgarch (coef.mtvgarch), 3
print.tvgarch (coef.tvgarch), 6
print.tvgarchTest (coef.tvgarchTest), 8
quantile, 4, 7, 9
quantile.mtvgarch (coef.mtvgarch), 3
quantile.tvgarch (coef.tvgarch), 6
quantile.tvgarchTest
(coef.tvgarchTest), 8
residuals.mtvgarch, 13
residuals.mtvgarch (coef.mtvgarch), 3
residuals.tvgarch, 21
residuals.tvgarch (coef.tvgarch), 6
residuals.tvgarchTest
(coef.tvgarchTest), 8
rnorm, 16, 22
summary.mtvgarch (coef.mtvgarch), 3
summary.tvgarch (coef.tvgarch), 6
summary.tvgarchTest (coef.tvgarchTest), 8
toLatex.mtvgarch (coef.mtvgarch), 3
toLatex.tvgarch (coef.tvgarch), 6
toLatex.tvgarchTest (coef.tvgarchTest), 8
tv (tvgarchObj), 20
tvgarch, 3, 5, 8, 10, 12, 15, 17, 17, 21, 22, 24
tvgarch-package, 2
tvgarchObj, 20
tvgarchSim, 3, 8, 10, 19, 21, 24
tvgarchTest, 3, 8, 10, 23
tvObj (tvgarchObj), 20
vcov.mtvgarch (coef.mtvgarch), 3
vcov.tvgarch (coef.tvgarch), 6
vcov.tvgarchTest (coef.tvgarchTest), 8
zoo, 4, 5, 7–10, 13, 16–18, 20–23