

Package ‘tvgarch’

February 5, 2021

Type Package

Title Time Varying GARCH Modelling

Version 1.0

Date 2021-01-28

Author Susana Campos-Martins [aut, cre], Genaro Sucarrat [ctb]

Maintainer Susana Campos-Martins <susana.martins@nuffield.ox.ac.uk>

Description Simulation, estimation and inference for 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. The TV long-term component, as in the multiplicative TV-GARCH model of Amado and Teräsvirta (2013) <doi:10.1016/j.jeconom.2013.03.006>, introduces non-stationarity in the variance process, where 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, hier.part, matrixStats, numDeriv, zoo

URL <https://sites.google.com/site/susanacamposmartins/>

NeedsCompilation no

Repository CRAN

Date/Publication 2021-02-05 09:00:02 UTC

R topics documented:

coef.tvgarch	2
garchObj	4
tvgarch	5
tvgarchSim	7
tvgarchTest	9

Index 11

coef.tvgarch *Extraction functions for univariate 'tvgarch' objects*

Description

Extraction functions for objects of class 'tvgarch'

Usage

```
## S3 method for class 'tvgarch'
coef(object, spec = NULL, ...)
## S3 method for class 'tvgarch'
fitted(object, spec = NULL, as.zoo = TRUE, ...)
## S3 method for class 'tvgarch'
logLik(object, ...)
## S3 method for class 'tvgarch'
nobs(object, ...)
## S3 method for class 'tvgarch'
predict(object, n.ahead = 10, newxtv = NULL, newxreg = NULL,
         newindex = NULL, n.sim = 5000, 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'
toLatex(object, digits = 4, ...)
## S3 method for class 'tvgarch'
vcov(object, spec = NULL, ...)
```

Arguments

object	an object of class 'tvgarch'
spec	specifies whether the function should extract specific results. If "tv", extracts results from the TV (long-term) component; if "garch" extracts results from the GARCH-X (short-term). If NULL, extracts results from full model
x	an object of class 'tvgarch'
as.zoo	logical. If TRUE, then the returned result is of class <code>zoo</code>
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 <code>g</code> equals <code>intercept.g</code>
newxreg	vector or matrix with the out-of-sample regressor values
newindex	zoo-index for the out-of-sample predictions. If NULL (default), then <code>1:n.ahead</code> 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:	numeric vector containing parameter estimates
fitted:	fitted conditional variance
logLik:	log-likelihood (normal density)
nobs:	the number of observations used in the estimation
predict:	a vector with the predictions (verbose=FALSE), or a matrix with both the predictions and the simulations (verbose=TRUE)
print:	print of the estimation results
quantile:	the fitted quantiles, i.e. the conditional standard deviation times the empirical quantile of the standardised innovations
residuals:	standardised residuals
vcov:	coefficient variance-covariance matrix

Author(s)

Susana Campos-Martins

References

Cristina Amado and Timo Teräsvirta (2013) Modelling volatility by variance decomposition, *Journal of Econometrics* 175, 142-153. Cristina Amado and Timo Teräsvirta (2014) Modelling changes in the unconditional variance of long stock return series, *Journal of Empirical Finance* 25, 15-35.

See Also

[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 coefficients:
coef(yEst)

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

## Extract log-likelihood:
logLik(yEst)

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

## Extract variance-covariance matrix:
vcov(yEst)

## Generate predictions:
predict(yEst)

```

garchObj

Auxiliary functions

Description

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

Usage

```

tv(speed, location, xtv, opt, order.g)
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)

```

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, calendar time, scaled between 0 and 1, is used as the transition variable
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).
order.g	integer vector of length s indicating the number of locations in each transition function of the TV component

<code>par.g</code>	numeric vector with the values of the parameters in the TV component. If <code>iter0=TRUE</code> , <code>par.g</code> takes the form <code>c(intercept.g,size,speed,location)</code> ; if <code>iter0=FALSE</code> , then <code>par.g=c(size,location)</code> and the values of the fixed parameters are provided using <code>fixed.par.g</code>)
<code>fixed.par.g</code>	NULL or numeric vector with the values of the parameters fixed in the TV component of form <code>c(intercept.g,speed)</code>
<code>fixed.h</code>	numeric vector, time-series or <code>zoo</code> containing the values of GARCH-X component)
<code>y</code>	numeric vector, time-series or <code>zoo</code> object
<code>iter0</code>	logical. If <code>FALSE</code> , some parameters in the TV component are fixed during the iterative estimation
<code>flag</code>	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
<code>par.h</code>	numeric vector with the values of the parameters in the GARCH-X component
<code>order.h</code>	integer vector of the form <code>c(p,q,r)</code> . The first entry controls the GARCH order, the second the ARCH order and the third the asymmetry order of the GARCH-X component
<code>xreg</code>	numeric vector, time-series or <code>zoo</code> object to include as covariates in the GARCH-X component
<code>fixed.g</code>	numeric vector, time-series or <code>zoo</code> containing the values of TV component)

Value

The values of the objective function or fitted variance components

Author(s)

Susana Campos-Martins

See Also

[tvgarch](#), [fitted.tvgarch](#), [residuals.tvgarch](#)

tvgarch

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, turbo = FALSE, trace = FALSE)
```

Arguments

<code>y</code>	numeric vector, time-series or zoo object
<code>order.g</code>	integer vector of length <code>s</code> indicating the number of locations in each transition function of the TV component
<code>order.h</code>	integer vector of the form <code>c(p,q,r)</code> . The first entry controls the GARCH order, the second the ARCH order and the third the asymmetry order of the GARCH-X component
<code>initial.values</code>	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 <code>list()</code> , the default, then the values are chosen automatically. TV component: <code>intercept.g</code> - NULL or numeric, <code>size</code> - NULL or numeric vector containing the size initial coefficients, <code>speed</code> - NULL or numeric vector containing the speed initial coefficients, <code>location</code> - NULL or numeric vector containing the location initial coefficients. GARCH-X component: <code>intercept.h</code> - numeric, <code>arch</code> - NULL or numeric vector containing the ARCH initial coefficients, <code>garch</code> - NULL or numeric vector containing the GARCH-type initial coefficients, <code>asym</code> - NULL or numeric vector containing the asymmetry-type initial coefficients, and <code>par.xreg</code> - NULL or numeric vector containing the X-type initial coefficients
<code>xtv</code>	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
<code>xreg</code>	numeric vector, time-series or zoo object to include as covariates in the GARCH-X component
<code>opt</code>	integer indicating whether the speed parameter in the TV component should be scaled. If 0, no scaling; if 1, <code>speed/sd(xtv)</code> ; if 2, <code>exp(speed)</code> .
<code>turbo</code>	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 <code>turbo</code> is set to TRUE, then the coefficient-covariance, fitted values and residuals can still be extracted subsequent to estimation with <code>vcov.tvgarch</code> , <code>fitted.tvgarch</code> and <code>residuals.tvgarch</code> , respectively
<code>trace</code>	logical. If TRUE all output is printed

Value

A list of class 'tvgarch'

Author(s)

Susana Campos-Martins

References

Cristina Amado and Timo Teräsvirta (2013) Modelling volatility by variance decomposition, *Journal of Econometrics* 175, 142-153. Cristina Amado and Timo Teräsvirta (2014) Modelling changes in the unconditional variance of long stock return series, *Journal of Empirical Finance* 25, 15-35.

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)

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

## Extract log-likelihood:
logLik(yEst)

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

## Extract variance-covariance matrix:
vcov(yEst)

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

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 = 0.8, speed = 2, location = 0.5, xtv = NULL,
           intercept.h = 0.2, arch = 0.1, garch = 0.8, asym = NULL, xreg = NULL,
           opt = 2, 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
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

A numeric vector or matrix 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)-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, trace = TRUE)
```

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
trace	logical, if TRUE all output is printed

Value

The number of location parameters in the single transition function

Author(s)

Susana Campos-Martins

References

Cristina Amado and Timo Teräsvirta (2017) Specification and testing of multiplicative time-varying GARCH models with applications, *Econometric Reviews* 36:4, 421-446.

See Also

[tvgarch](#), [garchx](#), [tvgarchSim](#)

Examples

```
set.seed(123)

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

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

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

Index

* Econometrics

coef.tvgarch, 2
garchObj, 4
tvgarch, 5
tvgarchSim, 7
tvgarchTest, 9

* Financial Econometrics

coef.tvgarch, 2
garchObj, 4
tvgarch, 5
tvgarchSim, 7
tvgarchTest, 9

* Nonlinear Time Series

coef.tvgarch, 2
garchObj, 4
tvgarch, 5
tvgarchSim, 7
tvgarchTest, 9

* Time-Varying Parameter Models

coef.tvgarch, 2
garchObj, 4
tvgarch, 5
tvgarchSim, 7
tvgarchTest, 9

coef.tvgarch, 2
constrOptim, 6, 7

fitted.tvgarch, 5
fitted.tvgarch(coef.tvgarch), 2

garchObj, 4
garchx, 7, 8, 10

logLik.tvgarch(coef.tvgarch), 2

nlminb, 6, 7
nobs.tvgarch(coef.tvgarch), 2

predict.tvgarch(coef.tvgarch), 2
print.tvgarch(coef.tvgarch), 2

quantile, 3

quantile.tvgarch(coef.tvgarch), 2

residuals.tvgarch, 5

residuals.tvgarch(coef.tvgarch), 2

rnorm, 8

toLatex.tvgarch(coef.tvgarch), 2

tv(garchObj), 4

tvgarch, 3, 5, 5, 8, 10

tvgarchSim, 3, 7, 7, 10

tvgarchTest, 9

tvObj(garchObj), 4

vcov.tvgarch(coef.tvgarch), 2

zoo, 2, 3, 5, 6, 8, 9