

# Package ‘tvgarch’

October 14, 2022

**Type** Package

**Title** Time Varying GARCH Modelling

**Version** 2.2

**Date** 2021-12-11

**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 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. 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 Ter{\a}svirta (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, hier.part

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

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-12-21 10:30:02 UTC

## R topics documented:

tvgarch-package . . . . .	2
coef.mtvgarch . . . . .	3
coef.tvgarch . . . . .	5
coef.tvgarchTest . . . . .	8
dccObj . . . . .	10
garchObj . . . . .	11
mtvgarch . . . . .	12

mtvgarchSim . . . . .	14
tvgarch . . . . .	16
tvgarchSim . . . . .	18
tvgarchTest . . . . .	20

<b>Index</b>	<b>22</b>
--------------	-----------

---

tvgarch-package	<i>Time Varying GARCH Modelling</i>
-----------------	-------------------------------------

---

## 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. 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 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: 2.2  
 Date: 2021-12-11  
 License: GPL>=2

## Author(s)

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

Maintainer: Susana Campos-Martins

Contributor: Genaro Sucarrat

## 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	<i>Extraction functions for multivariate 'mtvgarch' objects</i>
---------------	---

---

## Description

Extraction functions for objects of class 'mtvgarch'.

## Usage

```

## S3 method for class 'mtvgarch'
coef(object, spec = c("sigma2", "tv", "garch", "cc"), ...)
## S3 method for class 'mtvgarch'
fitted(object, spec = c("sigma2", "tv", "garch", "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("sigma2", "tv", "garch"), ...)
## 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("sigma2", "tv", "garch", "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 "sigma2", extracts results for TV-GARCH-X model. Only relevant for TV-GARCH-X models. Otherwise, extracts results for GARCH-X model. 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 <a href="#">zoo</a> .
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 <a href="#">quantile</a> .
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

Cristina Amado and Timo Teräsvirta (2013) Modelling volatility by variance decomposition, *Journal of Econometrics* 175, 142-153. Christian Francq and Jean-Michel Zakoïan (2016) Estimating multivariate volatility models equation by equation, *J. R. Stat. Soc. Ser. B Stat. Methodol* 78, 613-635.

## See Also

[mtvgarch](#), [mtvgarchSim](#), [tvgarch](#), [garchx](#), [zoo](#)

## Examples

```
set.seed(123)

## 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

*Extraction functions for univariate 'tvgarch' objects*

---

## Description

Extraction functions for objects of class 'tvgarch'.

## Usage

```
## S3 method for class 'tvgarch'
coef(object, spec = c("sigma2", "tv", "garch"), ...)
## S3 method for class 'tvgarch'
fitted(object, spec = c("sigma2", "tv", "garch"),
        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("sigma2", "tv", "garch"), ...)
## 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("sigma2", "tv", "garch"), ...)

```

### 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 "sigma2", extracts results for TV-GARCH-X model. Only relevant for TV-GARCH-X models. Otherwise, extracts results for GARCH-X model.
x	an object of class 'tvgarch'.
as.zoo	logical. If TRUE, then the returned result is of class <a href="#">zoo</a> .
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	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 <a href="#">quantile</a> .
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.
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**

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**

[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)
```

---

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'
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'
plot(x, ...)
## S3 method for class 'tvgarchTest'
predict(object, n.ahead = 10, newxreg = NULL,
  newindex = NULL, n.sim = 5000,
  as.zoo = TRUE, verbose = FALSE, ...)
## S3 method for class 'tvgarchTest'
print(x, ...)
## S3 method for class 'tvgarchTest'
quantile(x, probs = 0.025, names = TRUE, type = 7,
  as.zoo = TRUE, ...)
## S3 method for class 'tvgarchTest'
residuals(object, as.zoo = TRUE, ...)
## S3 method for class 'tvgarchTest'
summary(object, ...)
## S3 method for class 'tvgarchTest'
toLatex(object, digits = 4, ...)
## S3 method for class 'tvgarchTest'
vcov(object, ...)
```

## 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 <a href="#">zoo</a> .
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 <a href="#">quantile</a> .
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**

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**

[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)

```

---

 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.

**Author(s)**

Susana Campos-Martins

**See Also**[mtvgarch](#), [fitted.mtvgarch](#), [residuals.mtvgarch](#)

garchObj

*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**

```
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)
```

**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(size,location) and the values of the fixed parameters are provided using fixed.par.g).

<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 FALSE, 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, times eries or zoo 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](#)

---

mtvgarch

*Estimate a multivariate TV-GARCH-X model*

---

**Description**

Equation by equation estimation of a multivariate multiplicative TV-GARCH-X model with dnamic 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, dcc = FALSE,
turbo = TRUE, trace = FALSE)
```

**Arguments**

<code>y</code>	numeric matrix, time series or <code>zoo</code> object.
<code>order.g</code>	integer matrix with each row indicating the order.g for each series; number of locations in each transition function of the TV components.
<code>order.h</code>	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).
<code>order.x</code>	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.
<code>initial.values</code>	a list containing the initial parameter values passed on to the optimisation routines ( <code>constrOptim</code> for the TV component and <code>nlsminb</code> 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 vector, <code>size</code> - NULL or numeric matrix containing the size initial coefficients, <code>speed</code> - NULL or numeric matrix containing the speed initial coefficients, <code>location</code> - NULL or numeric matrix containing the location initial coefficients. GARCH-X component: <code>intercept.h</code> - numeric vector, <code>arch</code> - NULL or numeric matrix containing the ARCH initial coefficients, <code>garch</code> - NULL or numeric matrix containing the GARCH-type initial coefficients, <code>asym</code> - NULL or numeric matrix containing the asymmetry-type initial coefficients, <code>par.xreg</code> - NULL or numeric matrix containing the X-type initial coefficients, and <code>R</code> - initial correlation coefficients.
<code>xtv</code>	NULL or numeric vector, time series or <code>zoo</code> 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 <code>zoo</code> 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>dcc</code>	logical. If TRUE, dynamic conditional correlations are estimated. If FALSE (default), then the conditional correlations are constant.
<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 turbo is set to TRUE, then the coefficient-covariance, fitted values and residuals can still be extracted subsequent to estimation with <code>vcov.mtvgarch()</code> , <code>fitted.mtvgarch()</code> and <code>residuals.mtvgarch()</code> , respectively.
<code>trace</code>	logical. If TRUE all output is printed.

**Value**

An object of class 'mtvgarch'.

**Author(s)**

Susana Campos-Martins

## References

Cristina Amado and Timo Teräsvirta (2013) Modelling volatility by variance decomposition, *Journal of Econometrics* 175, 142-153. Christian Francq and Jean-Michel Zakoïan (2016) Estimating multivariate volatility models equation by equation, *J. R. Stat. Soc. Ser. B Stat. Methodol* 78, 613-635. Robert F. Engle (2002) Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models, *Journal of Business and Economic Statistics* 20, 339-350.

## See Also

[tvgarch](#), [garchx](#), [nlminb](#), [constrOptim](#)

## Examples

```
set.seed(123)

## 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

```
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**

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 <code>zoo</code> .
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 <code>rnorm</code> .

**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**

```
## 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))
```

---

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**

*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.



<code>order.h</code>	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.
<code>initial.values</code>	a list containing the initial parameter values passed on to the optimisation routines ( <code>constrOptim</code> for the TV component and <code>nminb</code> 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 or GARCH-X component of TV-GARCH-X: <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. Not relevant for stationary GARCH.
<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> . Only relevant for TV-GARCH models.
<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 when estimating a TV-GARCH.

**Value**

An object 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)

## Plot conditional volatilities:
plot(yEst)

## Extract log-likelihood:
logLik(yEst)

## Extract and store standardised residuals:
etaEst <- residuals(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 = 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 <a href="#">zoo</a> .
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 <a href="#">rnorm</a> .

**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**

<code>y</code>	numeric vector, time series or <a href="#">zoo</a> object.
<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>alpha</code>	the significance level.

**Value**

An object of class 'tvgarchTest'.

**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:
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
  - [coef.tvgarch](#), 5
  - [coef.tvgarchTest](#), 8
  - [dccObj](#), 10
  - [garchObj](#), 11
  - [mtvgarchSim](#), 14
  - [tvgarch](#), 16
  - [tvgarch-package](#), 2
  - [tvgarchSim](#), 18
  - [tvgarchTest](#), 20
- \* **Financial Econometrics**
  - [coef.mtvgarch](#), 3
  - [coef.tvgarch](#), 5
  - [coef.tvgarchTest](#), 8
  - [dccObj](#), 10
  - [garchObj](#), 11
  - [mtvgarch](#), 12
  - [mtvgarchSim](#), 14
  - [tvgarch](#), 16
  - [tvgarch-package](#), 2
  - [tvgarchSim](#), 18
  - [tvgarchTest](#), 20
- \* **Nonlinear Time Series**
  - [coef.mtvgarch](#), 3
  - [coef.tvgarch](#), 5
  - [coef.tvgarchTest](#), 8
  - [dccObj](#), 10
  - [garchObj](#), 11
  - [mtvgarch](#), 12
  - [mtvgarchSim](#), 14
  - [tvgarch](#), 16
  - [tvgarch-package](#), 2
  - [tvgarchSim](#), 18
  - [tvgarchTest](#), 20
- \* **Spillovers**
  - [mtvgarch](#), 12
- \* **Time Varying Parameter Models**
  - [coef.mtvgarch](#), 3
  - [coef.tvgarch](#), 5
  - [coef.tvgarchTest](#), 8
  - [dccObj](#), 10
  - [garchObj](#), 11
  - [mtvgarch](#), 12
  - [mtvgarchSim](#), 14
  - [tvgarch](#), 16
  - [tvgarch-package](#), 2
  - [tvgarchSim](#), 18
  - [tvgarchTest](#), 20
- [coef.mtvgarch](#), 3
- [coef.tvgarch](#), 5
- [coef.tvgarchTest](#), 8
- [constrOptim](#), 13, 14, 17, 18
- [dccObj](#), 10
- [fitted.mtvgarch](#), 11
- [fitted.mtvgarch \(coef.mtvgarch\)](#), 3
- [fitted.tvgarch](#), 12
- [fitted.tvgarch \(coef.tvgarch\)](#), 5
- [fitted.tvgarchTest \(coef.tvgarchTest\)](#), 8
- [garchObj](#), 11
- [garchx](#), 5, 14, 16, 18, 19, 21
- [logLik.mtvgarch \(coef.mtvgarch\)](#), 3
- [logLik.tvgarch \(coef.tvgarch\)](#), 5
- [logLik.tvgarchTest \(coef.tvgarchTest\)](#), 8
- [mtvgarch](#), 2, 5, 11, 12, 16
- [mtvgarchSim](#), 2, 5, 14
- [nlminb](#), 13, 14, 17, 18
- [nobs.mtvgarch \(coef.mtvgarch\)](#), 3
- [nobs.tvgarch \(coef.tvgarch\)](#), 5
- [nobs.tvgarchTest \(coef.tvgarchTest\)](#), 8
- [plot.mtvgarch \(coef.mtvgarch\)](#), 3
- [plot.tvgarch \(coef.tvgarch\)](#), 5

plot.tvgarchTest (coef.tvgarchTest), 8  
predict.mtvgarch (coef.mtvgarch), 3  
predict.tvgarch (coef.tvgarch), 5  
predict.tvgarchTest (coef.tvgarchTest),  
8  
print.mtvgarch (coef.mtvgarch), 3  
print.tvgarch (coef.tvgarch), 5  
print.tvgarchTest (coef.tvgarchTest), 8  
  
quantile, 4, 6, 9  
quantile.mtvgarch (coef.mtvgarch), 3  
quantile.tvgarch (coef.tvgarch), 5  
quantile.tvgarchTest  
(coef.tvgarchTest), 8  
  
residuals.mtvgarch, 11  
residuals.mtvgarch (coef.mtvgarch), 3  
residuals.tvgarch, 12  
residuals.tvgarch (coef.tvgarch), 5  
residuals.tvgarchTest  
(coef.tvgarchTest), 8  
rnorm, 15, 19  
  
summary.mtvgarch (coef.mtvgarch), 3  
summary.tvgarch (coef.tvgarch), 5  
summary.tvgarchTest (coef.tvgarchTest),  
8  
  
toLatex.mtvgarch (coef.mtvgarch), 3  
toLatex.tvgarch (coef.tvgarch), 5  
toLatex.tvgarchTest (coef.tvgarchTest),  
8  
tv (garchObj), 11  
tvgarch, 2, 5, 7, 9, 12, 14, 16, 16, 19, 21  
tvgarch-package, 2  
tvgarchSim, 2, 7, 9, 18, 18, 21  
tvgarchTest, 2, 7, 9, 20  
tvObj (garchObj), 11  
  
vcov.mtvgarch (coef.mtvgarch), 3  
vcov.tvgarch (coef.tvgarch), 5  
vcov.tvgarchTest (coef.tvgarchTest), 8  
  
zoo, 4–7, 9, 11–13, 15, 16, 19, 20