



Modeling Nonstationary Financial Volatility with the R Package `tvgarch`

Susana Campos-Martins 

Catholic University of Portugal
University of Oxford

Genaro Sucarrat 

BI Norwegian Business School

Abstract

Certain events can make the structure of volatility of financial returns to change, making it nonstationary. Models of time-varying conditional variance such as generalized autoregressive conditional heteroscedasticity (GARCH) models usually assume stationarity. However, this assumption can be inappropriate and volatility predictions can fail in the presence of structural changes in the unconditional variance. To overcome this problem, in the time-varying (TV-)GARCH model, the GARCH parameters are allowed to vary smoothly over time by assuming not only the conditional but also the unconditional variance to be time-varying. In this paper, we show how useful the R package `tvgarch` (Campos-Martins and Sucarrat 2023) can be for modeling nonstationary volatility in financial empirical applications. The functions for simulating, testing and estimating TV-GARCH-X models, where additional covariates can be included, are implemented in both univariate and multivariate settings.

Keywords: `tvgarch`, financial volatility, nonstationary GARCH models, time-varying parameter models.

1. Introduction

In the autoregressive conditional heteroscedasticity (ARCH) class of models proposed by Engle (1982), the variable of interest u_t is decomposed multiplicatively as

$$u_t = \sigma_t \eta_t, \quad \sigma_t > 0, \quad \eta_t \sim iid(0, 1), \quad t = 1, \dots, n, \quad (1)$$

where σ_t^2 is the conditional variance of u_t , η_t is a real-valued standardized innovation with mean zero and unit variance (e.g., the standard normal), and n is the sample size. In Engle (1982), u_t was the error term of a dynamic linear regression of inflation. So the conditional standard deviation σ_t was the uncertainty of the inflation forecast conditional on the past.

However, it is in finance that ARCH models have become the most popular. There, u_t is financial return, either raw or demeaned (i.e., u_t has mean zero), and the conditional variance σ_t^2 is a measure of the variability or volatility of return conditional on the past. Specifications within the ARCH class has proved particularly useful to model volatility clustering, a common characteristic of financial returns. In [Engle and Russell \(1998\)](#), it was noted that ARCH models can also be used to model non-negative variables, say, the trading volume of financial assets, the duration between financial trades, and so on. This interpretation of ARCH models is known as the multiplicative error model (MEM), see [Brownlees, Cipollini, and Gallo \(2012\)](#) for a survey. In what follows, we will focus exclusively on the financial return interpretation of u_t .

Stationary and nonstationary volatility

Define the conditional variance (i.e., volatility) as $\sigma_t^2 = E(u_t^2 | \mathcal{F}_{t-1})$, where \mathcal{F}_{t-1} is a suitable conditioning set (typically the information available up to and including $t - 1$). In standard approaches, the conditional variance process $\{\sigma_t^2\}$ is stationary. This implies that the unconditional variance $E(\sigma_t^2)$ is constant over time. History in general, however, and recent events in particular, show that financial volatility is not necessarily stationary. Events such as the 2008 global financial crisis and, more recently, the COVID-19 pandemic, can ultimately make the structure (e.g., baseline level or persistence) of financial volatility to change, making it nonstationary. This has important implications for statistical inference and financial decision making. [Lamoureux and Lastrapes \(1990\)](#), [Mikosch and Stărică \(2004\)](#), and [Hillebrand \(2005\)](#), for example, showed that ignoring changes in the unconditional volatility can lead to spurious persistence and long-memory effects. Next, these distortions may affect risk estimation (e.g., [Andreou and Ghysels 2008](#)), asset allocation (e.g., [Pettenuzzo and Timmermann 2011](#)), and estimates of the equity premium (e.g., [Pastor and Stambaugh 2001](#)), to mention but a few.

The most common volatility model is the stationary version of the first order generalized ARCH (GARCH) model, i.e., the standard GARCH(1, 1), which was proposed by [Bollerslev \(1986\)](#):

$$\sigma_t^2 = \omega + \alpha u_{t-1}^2 + \beta \sigma_{t-1}^2, \quad \omega, \alpha > 0, \quad 0 \leq \beta < 1. \quad (2)$$

The GARCH is the volatility analogue of the autoregressive moving average (ARMA) model. (In fact, as is well-known, the standard GARCH(1, 1) admits an ARMA(1, 1) representation with heteroscedastic noise, see [Francq and Zakoian 2019](#), p. 18.) Empirical comparisons suggest the standard GARCH(1, 1) fares well in many financial applications (see e.g., [Hansen and Lunde 2005](#)). However, numerous extensions have been proposed to account for the additional characteristics typically found in financial returns (see e.g., [Francq and Zakoian 2019](#) for a survey). This includes, amongst other, higher order models, asymmetry or “leverage” (good news and bad news affect volatility differently and so have a different predictive ability) and covariates. Most extensions assume the volatility process $\{\sigma_t^2\}$ is stationary.

Notwithstanding the usefulness and success of stationary GARCH models in describing the dynamic structure of financial volatility clustering, they will be misspecified if volatility is nonstationary. A remedy to this shortcoming is the time-varying (TV) GARCH model of [Amado and Teräsvirta \(2013, 2017\)](#), i.e., the key ingredient of the `tvgarch` package. In TV-GARCH models the conditional variance is decomposed multiplicatively as

$$\sigma_t^2 = g_t h_t, \quad g_t, h_t > 0, \quad (3)$$

where $\{g_t\}$ is a non-stochastic process and h_t is a “rescaled” stationary GARCH specification, e.g.,

$$h_t = \omega + \alpha\phi_{t-1}^2 + \beta h_{t-1}^2 \quad \text{where} \quad \phi_t^2 \equiv u_t^2/g_t. \quad (4)$$

Often, g_t is interpreted as the “long-term” component and h_t is interpreted as the “short-term” component. Henceforth, we will also refer to g_t as the TV component and h_t as the GARCH component. The $\sigma_t^2 = g_t h_t$ is still the conditional variance, but the unconditional volatility $E(\sigma_t^2) = g_t E(\phi_t^2)$ is now time-varying if g_t changes over time. Of course, the stationary GARCH is obtained as a special case when g_t is constant. In Amado and Teräsvirta (2013, 2017), the TV component is made up of logistic transition functions. This means g_t changes smoothly over time. However, abrupt changes can suitably be characterized by setting the speed of transition parameter sufficiently high, see Section 2.1. For a survey of multiplicative decompositions of volatility, see Amado, Silvennoinen, and Teräsvirta (2019).

A survey of software

Many publicly available softwares for statistics and econometrics contain modules for stationary volatility modeling. The most general and well-known R packages available for simulation, estimation and forecasting of GARCH models include the **fGarch** (Wuertz, Chalabi, Setz, Maechler, and Boshnakov 2024), **rugarch** (Galanos 2023) and **garchx** (Sucarrat 2021) packages for univariate series, and the **rmgarch** (Galanos 2022) package for multivariate series. Popular R packages for estimating stationary stochastic volatility models include **stochvol** and **factorstochvol** (Hosszejni and Kastner 2021) for, respectively, the univariate and the multivariate case. The recent R package **tvgarch** (Campos-Martins and Sucarrat 2023) implements the nonstationary TV-GARCH model, and is intended for the simulation, testing and modeling of nonstationary volatility in financial empirical applications. This package is on the Comprehensive R Archive Network (CRAN) since January 28, 2021. Even though other R or commercial packages are available for the modeling of volatility nonstationarity, to the best of our knowledge, **tvgarch** is the first, both inside and outside the R universe, to offer nonstationary volatility modeling with smooth transition logistic functions.

A summary of the main features of each R package available for modeling nonstationary volatility, namely **MSGARCH** (Ardia, Bluteau, Boudt, Catania, and Trottier 2019), **segM-Garch** (Cho and Korkas 2019) and **tvgarch**, is provided in Table 1. The novelties in **tvgarch**, which distinguishes it from the other two packages, are also highlighted. They differ from **tv-garch** in several ways. When the transition between volatility regimes is rather abrupt and assumed to be governed by a latent discrete Markov process, this is called the Markov-switching GARCH model (see, e.g., Hamilton and Susmel 1994). This model can be implemented using the R package **MSGARCH** (Ardia *et al.* 2019). Additionally to the way the nonstationary TV component is modeled, the key differences between the two packages lie in the modeling of the stationary GARCH component. The package **tvgarch** assumes a fairly flexible GARCH component by allowing the user to specify higher orders for the ARCH and GARCH terms (in addition to leverage effects) and to add explanatory variables. If the speed of transition turns out to be zero (meaning no transition exists and so volatility is stationary and its parameters remain constant), the different models collapse into their stationary GARCH counterparts. The GARCH component of either a stationary or nonstationary GARCH model is estimated by calling the **garchx()** function of the R package **garchx** (Sucarrat 2021), on which the package **tvgarch** depends. Also, because estimation is carried out by pseudo or quasi maximum likelihood (QML) in the package **tvgarch**, there is no need to specify a distribution for the

	<code>MSGARCH</code>	<code>segMGarch</code>	<code>tvgarch</code>
Nonstationary volatility	✓	✓	✓
Abrupt transitions	✓	✓	✓
Smooth transitions			✓
Exogenous transition variables			✓
QML (robust to misspecified density)		✓	✓
Higher ARCH and GARCH orders		✓	✓
Asymmetry	✓		✓
Higher asymmetry order			✓
Covariates (X)			✓
Test for nonstationarity		✓	✓
Multivariate		✓	✓

Table 1: A feature-based comparison of the R packages `MSGARCH` (Ardia *et al.* 2019), `segMGarch` (Cho and Korkas 2019) and `tvgarch` (Campos-Martins and Sucarrat 2023) for modeling nonstationary volatility with GARCH estimation.

error term as is the case for the package `MSGARCH`. Finally, in a Markov-switching GARCH model, the volatility forecasts are quickly adjusted to changes in the unconditional volatility. In the smooth transition setting, because the transitions between volatility states are ruled by a transition variable, the way volatility forecasts are computed will depend on both the speed of transition and/or whether this variable is deterministic (e.g., time) or stochastic (e.g., past or realized volatility). Another R package that allows nonstationary volatility is `segMGarch` (Cho and Korkas 2019), which provides a segmentation algorithm for detecting change-points in high-dimensional GARCH processes. It allows for multiple points of change, which are common to a subset or all of the component time series in either/both conditional variances or/and conditional correlations. A similar segmentation approach is proposed by Amado and Teräsvirta (2014) for testing and estimating nonstationary volatility using the TV-GARCH model. The unconditional variance of a long stock return series is modeled by dividing the time series into non-overlapping segments each comprising at least of 1500 observations, applying the strategy to them and combining the results into a single model.

Another strongly related nonstationary volatility model, which is also based on a multiplicative decomposition of the conditional variance, is the spline GARCH model (Engle and Rangel 2008). It allows the unconditional variance to change smoothly over time, but the transition is rather described as an exponential quadratic spline function of time. The closest the spline GARCH model can be to the TV-GARCH is when time (scaled between zero and one) is selected as the transition variable in the logistic transition function. In this setting, both models are able to describe dominant changes in the (unconditional) volatility of a time series of financial returns. The spline GARCH model can be implemented with the module `G@RCH` (Laurent 2018) of `OxMetrics` (Doornik 2018), where the number of knots (and of equally spaced intervals over the sample period) has to be specified by the user. In the module, additional explanatory variables are also allowed.

As it turns out, models such as the TV-GARCH, where structural changes in the unconditional variance are allowed, also help in explaining the long-range dependence and the integrated GARCH effect of financial return series (Mikosch and Stărică 2004; Amado and Teräsvirta 2013). Volatility models of strong persistence such as the integrated GARCH (Engle and

Bollerslev 1986) or the fractionally integrated GARCH (Baillie, Bollerslev, and Mikkelsen 1996) can be implemented with the module **G@RCH** (Laurent 2018) of **OxMetrics** (Doornik 2018) or the R package **rugarch** (Galanos 2023).

Main functions

For most users, the main functions of the package **tvgarch** are `tvgarch()`, `tvgarchTest()` and `mtvgarch()`. The function `tvgarch()` estimates a univariate TV-GARCH model, whereas `mtvgarch()` estimates a multivariate TV-GARCH model. The function `tvgarchTest()` implements a test procedure that tries to identify the number of changes in the unconditional volatility. It is important to carry out this procedure *before* estimating a TV-GARCH model, since the parameters of a TV-GARCH model are not identified if one tries to estimate more changes in the unconditional volatility than there actually is. Section 3 contains a more detailed overview of the main functions of **tvgarch**.

On reproducibility

Estimation in our package is via numerical optimization procedures. So users may experience that their results differ from the ones reported in this paper, even if the same code is used. This may be due to differences in hardware, OS, R version, package versions or the setting in `options()`. The results in this paper have been obtained using MS Windows Pro 64-bit version 10.0.19045, R 4.3.1 (R Core Team 2024), **tvgarch** 2.4 (Campos-Martins and Sucarrat 2023), **garchx** 1.5 (Sucarrat 2021), **numDeriv** 2016.8-1.1 (Gilbert and Varadhan 2019), **dc-cmidas** 0.1.0 (Candila 2024) on a Lenovo ThinkPad X390 Yoga with an Intel i7-8565U CPU at 1.80GHz chip. All packages are available on CRAN.

Organization of paper

The remainder of this paper is organized as follows. The next section describes univariate and multivariate TV-GARCH models (Sections 2.1–2.4), and gives an overview of aspects related to estimation (Section 2.5). Section 3 provides an overview of the main functions of the package **tvgarch**. In Section 4, we show how to simulate either a univariate or a multivariate time series with nonstationary volatility using the R package **tvgarch**. Section 5 contains an application to real data. Specifically, an illustration of the testing and estimation strategies for modeling nonstationary financial volatility is given. This is done first for a single series of financial returns and then for a multivariate time series, where not only the conditional variances but also the conditional correlations of returns are estimated. Finally, Section 6 concludes.

2. The TV-GARCH model

The TV-GARCH model of Amado and Teräsvirta (2013, 2017) is the key ingredient of the **tvgarch** package. To recall, the univariate TV-GARCH model specifies the financial return u_t (raw or demeaned) as

$$u_t = \sigma_t \eta_t, \quad \sigma_t > 0, \quad \eta_t \sim iid(0, 1), \quad \sigma_t^2 = g_t h_t, \quad t = 1, \dots, n,$$

where σ_t^2 is the conditional variance, η_t is a zero mean and unit variance innovation, g_t is

the non-stochastic TV component, h_t is the rescaled GARCH component and n is the sample size, see Equations 1–4 above.

2.1. The TV component, g_t

The non-stochastic TV component g_t takes the simplest form when it is described by a single logistic transition function, and a single transition. We will refer to this as the TV(1) specification:

$$\begin{aligned} g_t &= \delta_0 + \delta_1 G(\gamma, c; t/n), & 0 < \delta_0, & \quad -\delta_0 < \delta_1, \\ G(\gamma, c; t/n) &= \frac{1}{1 + \exp\{-\gamma(t/n - c)\}}, & 0 < \gamma, & \quad c \in (0, 1). \end{aligned}$$

The TV(1) is thus characterized by the four parameters $\boldsymbol{\vartheta}^g = (\delta_0, \delta_1, \gamma, c)'$. The parameter δ_1 is the total transition size, since the logistic function G varies between 0 and 1. In the special case where the transition size is zero, i.e., $\delta_1 = 0$, then g_t is constant and equal to δ_0 . In this case γ and c are not identified in estimation, since any pair of admissible values (γ, c) is compatible with $g_t = \delta_0$ for all t . It is therefore important to identify the number of transitions before estimation, see Section 5.2.1. Notice that the parameter restriction $-\delta_0 < \delta_1$ ensures $0 < g_t$ for all t . The c is the mid-point of the location of transition, and is expressed as a fraction of the sample. For example, if $c = 0.5$ and $n = 100$, then the mid-point of the location of transition is $t = c \cdot n = 50$. Locations of transition are expressed as fractions due to the asymptotic estimation theory that we rely on, see Amado and Teräsvirta (2013). The γ is the speed of transition: The higher, the faster and more abrupt. In fact, as $\gamma \rightarrow \infty$, the function G tends to a step function with the step at c . The lower γ is, the slower the transition is. Notice that time enters as scaled time in G , i.e., $t/n \in (0, 1]$. Again, this is due to the asymptotic estimation theory. More generally, t/n is a transition variable governing the change in the unconditional variance. In principle, any non-stochastic continuous transition variable $s_t \in [0, 1]$ can be used. So $s_t = t/n$ is a special case. The left panel of Figure 1 plots the TV(1) for transition speeds $\gamma = 10, 25, 50$ when $\delta_0 = \delta_1 = 1$ and $s_t = t/n$.

The number of locations of transition in g_t can be increased in three ways. The first restricts transition sizes and transition speeds to be equal across locations:

$$\begin{aligned} g_t &= \delta_0 + \delta_1 G(\gamma, \mathbf{c}; s_t), & 0 < \delta_0 + \delta_1, \\ G(\gamma, \mathbf{c}; s_t) &= \left(1 + \exp\left\{-\gamma \prod_{l=1}^{\tau} (s_t - c_l)\right\}\right)^{-1}, & c_1 \leq \dots \leq c_k. \end{aligned} \quad (5)$$

Here, the number of transitions is τ , so this is a TV(τ) specification. The parameters to be estimated are $\boldsymbol{\vartheta}^g = (\delta_0, \delta_1, \mathbf{c})'$. The right panel of Figure 1 plots an example with two locations, $\mathbf{c} = (0.25, 0.75)'$, for transition speeds $\gamma = 10, 25, 50$ when $s_t = t/n$.

The second way the number of locations of transition can be increased, allows the transition sizes and transition speeds to differ across locations:

$$\begin{aligned} g_t &= \delta_0 + \sum_{j=1}^k \delta_j G_j(\gamma_j, c_j; s_t), & 0 < \delta_0 + \sum_{j=1}^k \delta_j, \\ G(\gamma_j, c_j; s_t) &= \frac{1}{1 + \exp\{-\gamma_j(s_t - c_j)\}}. \end{aligned}$$

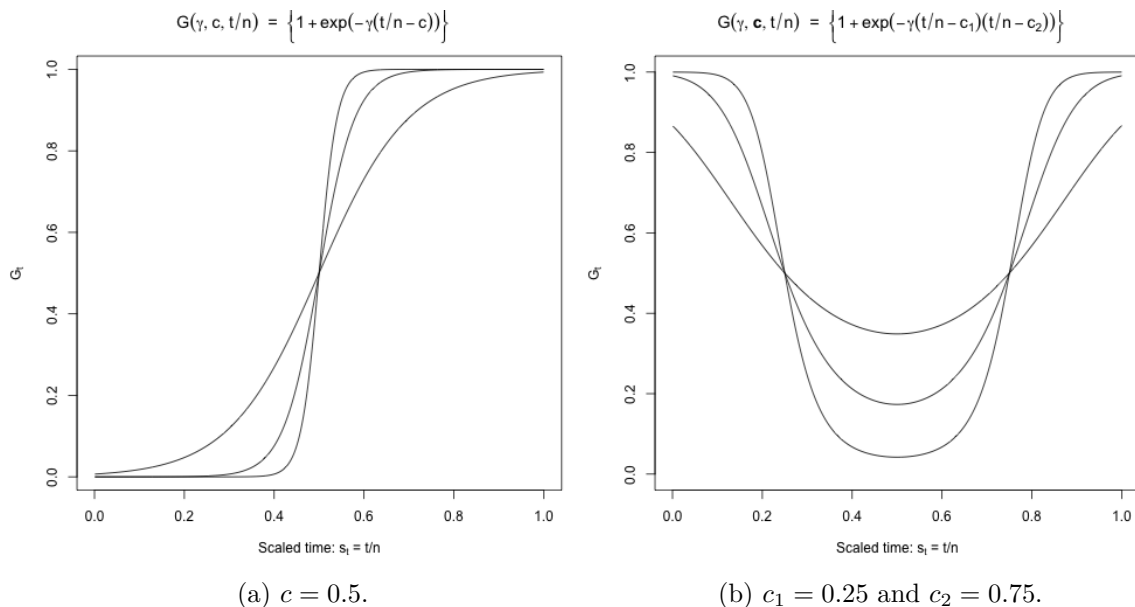


Figure 1: The logistic function with one and two locations of transition for different values of the speed of transition, i.e., $\gamma = 10, 25, 50$.

The number of transitions is k , i.e., one transition in each logistic function. The parameters to be estimated are now $\boldsymbol{\vartheta}^g = (\boldsymbol{\delta}', \boldsymbol{\gamma}', \mathbf{c}')'$, where $\boldsymbol{\delta} = (\delta_0, \delta_1, \dots, \delta_k)'$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_k)'$ and $\mathbf{c} = (c_1, \dots, c_k)'$. Notice that, here, a more complicated set of parameter restrictions on the δ_j 's are needed to ensure $0 < g_t$ for all t .

The third way the number of locations of transition can be increased, combines the first two ways:

$$g_t = \delta_0 + \sum_{j=1}^k \delta_j G_j(\gamma_j, \mathbf{c}_j; s_t), \quad 0 < \delta_0 + \sum_{j=1}^k \delta_j, \quad (6)$$

$$G_j(\gamma_j, \mathbf{c}_j; s_t) = \left(1 + \exp \left\{ -\gamma_j \prod_{l=1}^{\tau_j} (s_t - c_{jl}) \right\} \right)^{-1}. \quad (7)$$

So the number of logistic functions is k , and the number of transitions is $\sum_{j=1}^k \tau_j$.

2.2. The rescaled GARCH component, h_t

To recall, the first order rescaled GARCH(1, 1) specification is given by

$$h_t = \omega + \alpha \phi_{t-1}^2 + \beta h_{t-1}^2 \quad \text{where} \quad \phi_t^2 \equiv u_t^2 / g_t,$$

see Equation 4. The parameter $0 < \omega$ is the intercept, $0 < \alpha$ is the ARCH parameter and $0 \leq \beta$ is the GARCH parameter. Further restrictions can be imposed on α and β to guarantee the stationarity of $\{h_t\}$. In particular, weak stationarity holds if and only if $\alpha + \beta < 1$.

While the GARCH(1, 1) specification is widely used, it is well-known that it is not satisfactory in the presence of asymmetry or “leverage” (asymmetry is often attributed to financial

leverage). That is, volatility tends to be higher after negative returns. Usually, such asymmetry is accommodated by including the product of the lagged indicator term $\mathbb{1}(\phi_{t-1} < 0)$ and ϕ_{t-1}^2 , where $\mathbb{1}(A)$ is equal to 1 when A holds and zero otherwise. This leads to the rescaled asymmetric GARCH(1, 1):

$$h_t = \omega + \alpha\phi_{t-1}^2 + \beta h_{t-1}^2 + \lambda\phi_{t-1}^2\mathbb{1}(\phi_{t-1} < 0).$$

We will also refer to this specification as the GARCH(1,1,1).

First order models are often sufficient to describe the dynamics in empirical applications. When they do not suffice, then more general GARCH(p, q, r) models can be specified:

$$h_t = \omega + \sum_{j=1}^q \alpha_j \phi_{t-j}^2 + \sum_{j=1}^p \beta_j h_{t-j} + \sum_{j=1}^r \lambda_j \phi_{t-j}^2 \mathbb{1}(\phi_{t-j} < 0).$$

The p , q and r refer to the GARCH, ARCH and asymmetry order, respectively.

Finally, to accommodate the possible influence of stationary covariates, a GARCH(p, q, r)-X model can be specified, where ‘X’ indicates the inclusion of stationary covariates:

$$h_t = \omega + \sum_{j=1}^q \alpha_j \phi_{t-j}^2 + \sum_{j=1}^p \beta_j h_{t-j} + \sum_{j=1}^r \lambda_j \phi_{t-j}^2 \mathbb{1}(\phi_{t-j} < 0) + \boldsymbol{\zeta}' \mathbf{x}_t. \quad (8)$$

To ensure h_t is positive for every t , each entry of the $v \times 1$ covariate vector \mathbf{x}_t must be non-negative for all t . Also, the following parameter restrictions must hold: $\omega > 0$, $\alpha_j \geq 0, j = 1, \dots, q$, $\beta_j \geq 0, j = 1, \dots, p$, $\lambda_j \geq 0, j = 1, \dots, r$ and $\zeta_j \geq 0, j = 1, \dots, v$. These parameter restrictions are implemented inside the function `tvgarch()` during estimation, and cannot be altered by the user.

2.3. The TV-GARCH(p, q, r)-X model

At its most general, the TV component in Equations 6–7 is combined with the GARCH(p, q, r)-X component in Equation 8 to form the TV-GARCH(p, q, r)-X model. In most cases, however, a simple specification with few terms is sufficient for the problem at hand. For this reason, the default of the function `tvgarch()` is to estimate the TV(1)-GARCH(1,1) specification:

$$g_t = \delta_0 + \frac{\delta_1}{1 + \exp(-\gamma(t/T - c))}, \quad h_t = \omega + \alpha\phi_{t-1}^2 + \beta h_{t-1}.$$

In Section 5 we discuss how estimation of this and other specifications proceed in practice.

2.4. The multivariate TV-GARCH(p, q, r)-X

In conditional correlation GARCH models, the conditional covariances are usually assumed to have a multiplicative decomposition, where the conditional variances are modeled separately from the conditional correlations. See [Bauwens, Laurent, and Rombouts \(2006\)](#) for a survey on multivariate GARCH modeling.

In this class of models, the conditional variances and/or the conditional correlations can be further multiplicatively decomposed when processes are nonstationary. Much attention has been dedicated to the decomposition of nonstationary conditional variances into short- and

long-term variance components, which can then be modeled separately. For a recent survey on GARCH models with multiplicative decomposition of conditional variances (and conditional correlations), we refer to [Amado *et al.* \(2019\)](#).

The general conditional correlation GARCH model can be described as follows. Let $\mathbf{u}_t = (u_{1t}, \dots, u_{mt})'$ denote the vector of financial returns (raw or demeaned). Throughout, we assume $\mathbb{E}(\mathbf{u}_t | \mathcal{F}_{t-1}) = \mathbf{0}_m$, where $\mathbf{0}_m$ denotes an $m \times 1$ zero vector and \mathcal{F}_{t-1} contains the information available until time $t - 1$. Henceforth, our focus is on the conditional covariance matrix of \mathbf{u}_t , \mathbf{H}_t , where $\mathbf{H}_t \equiv \mathbb{E}(\mathbf{u}_t \mathbf{u}_t' | \mathcal{F}_{t-1})$. We assume \mathbf{H}_t exists and is positive definite for all t .

The conditional covariance matrix of \mathbf{u}_t , \mathbf{H}_t , is multiplicatively decomposed as follows:

$$\mathbf{H}_t = \mathbf{D}_t \mathbf{G}_t \mathbf{P}_t \mathbf{G}_t \mathbf{D}_t, \quad (9)$$

where $\mathbf{D}_t = \text{diag} \{h_{1t}^{1/2}, \dots, h_{mt}^{1/2}\}$ is a diagonal matrix containing the rescaled conditional standard deviations, where h_{it} is assumed to follow a GARCH(p_i, q_i, r_i)- X_i structure as defined in Equation 8, and $\mathbf{G}_t = \text{diag} \{g_{1t}^{1/2}, \dots, g_{mt}^{1/2}\}$ is a diagonal matrix of the positive-valued and deterministic functions, g_{it} , describing the TV component in Equations 6–7. In the GARCH-X component, p_i indicates the ARCH order, q_i the GARCH order, r_i the asymmetry order, and X_i that covariates are also included in the i th variance equation. \mathbf{P}_t is a positive definite matrix that contains the time-varying conditional correlations $\rho_{ijt}, i, j = 1, \dots, m$, with $\rho_{ijt} = 1$ when $i = j$.

In the dynamic conditional correlation GARCH model of [Engle \(2002\)](#), the conditional correlations of financial returns are computed as a function of the past information and given by the dynamic matrix process

$$\mathbf{Q}_t = (1 - \theta_1 - \theta_2) \bar{\mathbf{Q}} + \theta_1 \boldsymbol{\eta}_{t-1} \boldsymbol{\eta}_{t-1}' + \theta_2 \mathbf{Q}_{t-1}, \quad (10)$$

where $\theta_1 > 0$, $\theta_2 > 0$, $\theta_1 + \theta_2 < 1$ and $\bar{\mathbf{Q}}$ is the unconditional correlation matrix of the volatility standardized residuals, $\boldsymbol{\eta}_t$. The parameter restrictions are imposed to ensure positive definiteness of \mathbf{Q}_t , which ensure that also the time-varying conditional correlations \mathbf{P}_t are also positive definite. To obtain valid correlation coefficients, the correlation matrix is rescaled as follows:

$$\mathbf{P}_t = \text{diag}(\mathbf{Q}_t)^{-1/2} \mathbf{Q}_t \text{diag}(\mathbf{Q}_t)^{-1/2}. \quad (11)$$

A special case is nested in this model when conditional correlations are time-invariant, i.e., $\mathbf{P}_t \equiv \mathbf{P}$ in which case the model becomes the constant conditional correlation GARCH model ([Bollerslev 1990](#)). [Francq and Zakoian \(2019, Chapter 10\)](#) and [Amado *et al.* \(2019, Section 5\)](#) contains surveys of multivariate GARCH models.

In practice, testing and estimation of the variance component of the conditional covariance matrix are carried out equation by equation ([Francq and Zakoian 2012, 2016](#)). Estimation of the correlation component is then carried out conditionally on the estimated variances completing the two-step estimation procedure; see [Engle \(2002\)](#).

2.5. Estimation

In the univariate case, estimation of the parameters in g_t and h_t is implemented with `tvrgarch()`. On an average contemporary laptop, estimation of a TV(1)-GARCH(1, 1) model with $n \approx$

2000 typically takes 1–5 seconds. Estimation is via quasi maximum likelihood (QML). The word “quasi” means estimation is by means of maximum likelihood (ML) based on a normal density, but estimates are nevertheless consistent (subject to suitable regularity conditions) even if the density of the innovations $\{\eta_t\}$ is not normal. Due to the numerical complexity of the problem, estimation is implemented with the iterative two step method derived by Amado and Teräsvirta (2013), which is an adaption of the maximization by parts method of Song, Fan, and Kalbfleisch (2005). Let $\boldsymbol{\vartheta} = (\boldsymbol{\vartheta}^h, \boldsymbol{\vartheta}^g)'$ denote the full set of parameters, where $\boldsymbol{\vartheta}^h$ are the parameters of the GARCH component h_t , and $\boldsymbol{\vartheta}^g$ are the parameters of the TV component g_t . In each iteration i , the parameter $\boldsymbol{\vartheta}^g$ is estimated in the first step, and then the parameter $\boldsymbol{\vartheta}^h$ is estimated in a second step. Next, the estimation results from iteration i are used for the next iteration $i + 1$. This iterative procedure continues until convergence. In each iteration, the first step calls the function `constrOptim()` internally, and the second step calls the function `garchx()` from the package `garchx` internally, which invokes `nlmminb()`. To ensure the estimation procedure proceeds correctly from one component to another, and from one iteration to the next, user-specification of the arguments in `constrOptim()` and `garchx()` is limited. Currently, only the arguments `initial.values` (a numeric vector) and `opt` (an integer equal to 0, 1 or 2) are available to control the numerical optimization procedure. The former argument can be used to user-specify the starting values of the parameters. The latter argument controls the scaling of the speed parameters $\gamma_j, j = 1, \dots, k$. This is useful, since their estimates can converge towards very large values, causing numerical problems when computing the standard errors. To overcome this, the speed parameters can be scaled. If `opt = 0`, no scaling is applied. This is the default for the `tv()` function and the functions for simulating TV-GARCH processes, namely the `tvgarchSim()` and `mtvgarchSim()` functions. When `opt = 1`, then the speed of transition, γ , is replaced by $\gamma/\text{sd}(s_t)$, where `sd` denotes the sample standard deviation. Finally, if `opt = 2`, $\exp(\gamma)$ is used in place of γ . To avoid having the speed of transition converging to a very large number, `opt = 2` is the default. Notice also that, internally, the speed parameters are by default bounded from above by 250 when `opt = 0` or `opt = 1`, and by $\ln 250$ when `opt = 2`. This can be changed via the `upper.speed` argument in `tvgarch()`. A sketch of the theory behind the estimation of a univariate nonstationary GARCH model is provided in Appendices A and B.

In the multivariate case, estimation of the parameters is implemented with `mtvgarch()`. Estimation is carried out with the equation by equation approach of Francq and Zakoian (2012) using the univariate estimation procedure described above. In other words, each equation entails a call to `tvgarch()` internal to `mtvgarch()`. Next, after the estimation of the parameters of the conditional variances, the standardized residuals are used to estimate the dynamic conditional correlation (DCC) model of Engle (1982). This is optional and can be controlled via the argument `dcc` in `mtvgarch()`. See Appendix C for more details on the dynamic conditional correlations.

3. An overview of the package

For the average user, the main functions of the `tvgarch` package are `tvgarch()`, `tvgarchTest()` and `mtvgarch()`. The function `tvgarch()` estimates a univariate TV-GARCH model and returns an object of class “`tvgarch`”. The function `tvgarchTest()` implements a test procedure to identify how many transitions or changes there are in the unconditional volatility. In empirical applications, it is important to carry out this test *before* estimating a TV-GARCH

model, since the parameters of the TV-GARCH model are not identified if one tries to estimate more transitions than there actually is. The function `mtvgarch()` estimates a multivariate TV-GARCH model and returns an object of class `"mtvgarch"`. Note that the function `tvgarchTest()` can also be used prior to the estimation of a multivariate TV-GARCH model. Some users will be interested in simulating from TV-GARCH models. To this end, the functions `tv()`, `tvgarchSim()` and `mtvgarchSim()` can be used. The function `tv()` can be used to simulate from the non-stochastic component g_t , i.e., the dynamics of the time-varying unconditional volatility. This can be useful to obtain a detailed idea of how the different parameters affect the transition dynamics. The function `tvgarchSim()` simulates from a univariate TV-GARCH, whereas `mtvgarch()` simulates from a multivariate TV-GARCH model. Here follows a summary of the main characteristics of these functions:

- `tvgarch()`: Estimate a univariate TV-GARCH model.

Main arguments: A univariate time series y – e.g., an object of class `"vector"`, `"ts"` or `"zoo"` (Zeileis and Grothendieck 2005) – an integer vector `order.g` that specifies the general g_t component in Equations 6–7 and an integer vector `order.h` that specifies the GARCH(p, q, r) component h_t in Equation 8. Additional covariates can be included via the `xreg` argument.

Value: A list of class `"tvgarch"`.

S3 methods: `coef()`, `fitted()`, `logLik()`, `nobs()`, `plot()`, `predict()`, `print()`, `quantile()`, `residuals()`, `summary()`, `toLatex()`, `vcov()`. Type `help(coef.tvgarch)` for detailed documentation of the methods.

- `tvgarchTest()`: A test procedure to identify the number of transitions.

Main arguments: A time series y (e.g., an object of class `"vector"`, `"ts"` or `"zoo"`) and the significance level `alpha` (the default is 0.05) to be used in the procedure.

Value: A list of class `"tvgarchTest"`. The entry named `"order.g"` contains the number of transitions suggested by the test procedure. The entry named `"mat"` provides the details of the non-robust test procedure, whereas the entry named `"matRob"` contains the details the robust version. To recall, “robust” means robust to non-normality of the innovations, i.e., the η_t 's.

S3 methods: Same as those of `tvgarch()`, type `help(coef.tvgarchTest)` for detailed documentation. Note that the values returned from these methods are always those of a GARCH(1,1) specification estimated under the assumption that the unconditional volatility is constant.

- `mtvgarch()`: Estimate a multivariate TV-GARCH model.

Main arguments: A multivariate time series y (e.g., an object of class `"matrix"`, `"ts"` or `"zoo"`), a matrix `order.g` in which row i specifies the g_{it} component of equation i , a matrix `order.h` in which row i specifies the h_{it} component of equation i , and a logical `dcc` that determines whether constant (`FALSE`, the default) or time-varying dynamic (`TRUE`) conditional correlations should be fitted.

Value: A list of class `"mtvgarch"`.

S3 methods: Same as those of `tv`garch(), type `help(coef.mtv`garch) for detailed documentation of the methods.

- `tv()`: Simulate the dynamics of the component g_t , i.e., unconditional volatility.

Main arguments: The number of observations `n`, a numeric vector `location` with the scaled locations (i.e., values between 0 and 1) of the transitions and a numeric vector `speed` with the transition speeds.

Value: A time-series of class "zoo" (default) ordered by scaled time. Optionally, by setting `as.zoo = FALSE`, the returned series is of class "vector".
- `tv`garchSim(): Simulate from a univariate TV-GARCH model.

Main arguments: The number of observations `n`, the arguments that specify the g_t component (`intercept.g`, `size`, `speed` and `location`), and the arguments that specifies the GARCH model, i.e., the h_t component (`intercept.h`, `arch`, `garch`, `asym`, `xreg`).

Value: A univariate time-series of class "zoo" (default). For more detailed output, set `verbose = TRUE`.
- `mtv`garchSim(): Simulate from a multivariate TV-GARCH model.

Main arguments: The number of observations `n`, the arguments that specify the g_t component (`intercept.g`, `size`, `speed` and `location`), the arguments that specifies the GARCH model, i.e., the h_t component (`intercept.h`, `arch`, `garch`, `asym`, `xreg`), and the arguments that specify the conditional correlations (`R`, `dcc`, `dcc.par`).

Value: A multivariate time-series of class "zoo" (default). For more detailed output, set `verbose = TRUE`.

4. Simulation from TV-GARCH models

Simulating TV-GARCH processes can be very useful. For example, to study the finite sample properties of estimators or test statistics, simulated data are needed. We may also be interested in specific cases (e.g., a multivariate TV-GARCH model with very high correlations), which can be difficult to find empirically, or making predictions (e.g., when constructing confidence intervals using parametric bootstrapping).

Nonstationary GARCH processes can be simulated with the `tv`garchSim() function for a univariate times series or the `mtv`garchSim() function for an m -dimensional series, in which case the simulated univariate models constitute a nonstationary vector GARCH from correlated innovations.

4.1. Simulating the TV component

To simulate the values of the logistic transition function in Equation 6, the auxiliary function `tv()` of *tv*garch can be used. The transition variable is defined by the `xtv` argument. When `xtv = NULL`, the transition variable is a proxy for time, scaled between 0 and 1, for a total of `n` observations. When `xtv = NULL`, the argument `n` has to be provided. To construct the

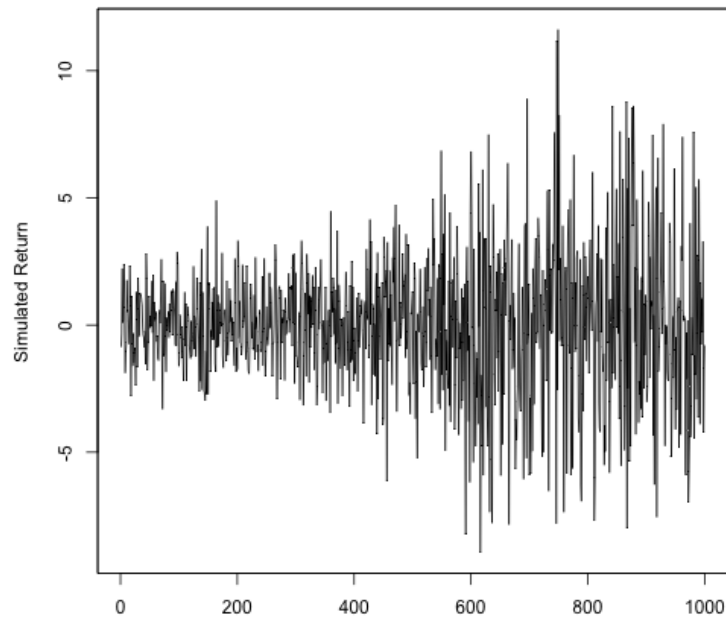


Figure 2: The time series of returns simulated from a TV(1)-GARCH(1,1) model.

transition variable `xtv` when `xtv = NULL` for 1000 observations, the following code can be used:

```
R> n <- 1000
R> stime <- seq(1:n)/n
```

This is similar to the code in the `tv()` function used to construct the transition variable `xtv` when `xtv = NULL`. The variable `xtv` can be returned if the argument `verbose = TRUE`. The following code can be used to generate the values of the logistic function in Equation 5 with $\gamma = 10$ and a single location of transition with the mid-point of change at $c = 0.5$:

```
R> G1t <- tv(speed = 10, location = 0.5, n = 1000)
```

By default, the values returned by `tv()` is a vector of class "zoo", but this can be changed via the `zoo` argument. Graph (a) in Figure 1 contains a plot of the values, together with plots of `speed = 25` and `speed = 50`. The higher the `speed`, the faster or more abrupt transition. If instead two locations of change are assumed at $c_1 = 0.25$ and $c_2 = 0.75$, then the following code produces such logistic function values:

```
R> G2t <- tv(speed = 10, location = c(0.25, 0.75), n = 1000)
```

Graph (b) in Figure 1 contains a plot of these values, together with plots of `speed = 25` and `speed = 50`. Again, the higher the `speed`, the faster or more abrupt transitions.

4.2. Simulating from a univariate TV-GARCH model

The simplest TV-GARCH model is simulated by default. The following code allows us to obtain a time series with 1000 observations whose conditional variance is simulated from a TV(1)-GARCH(1,1) model and to store the observations in an object called `uSim`:

```
R> set.seed(123)
R> uSim <- tvgarchSim(n = 1000)
```

The simulated series of returns are visualized in Figure 2, from where it can be seen that the simulated returns become more volatile in the second half of the simulated sample (this path is obtained by assuming one transition whose mid-point of change coincides with the median observation).

By default, the independent innovations, $\eta_t, t = 1, \dots, n$, are random draws from a standard normal distribution. This can be changed through the `innovations` argument of the `tvgarchSim()` function, where the innovations are supplied as an $n \times 1$ vector. In the default setting, `opt = 0` and because the transition variable is not specified (i.e., `xTV = NULL`), $s_t \equiv t/n$. The arguments of `tvgarchSim()` can be described as follows:

- `order.g` is an integer vector of length k that indicates the number of locations in each logistic transition function of the TV component, recall Equations 6–7. The default, `order.g = 1`, means the conditional variance of the time series is nonstationary with a single transition function and a single location parameter. By contrast, if `order.g = c(1,2)`, then the TV component contains one location in the first logistic function, and two locations in the second logistic function. That is, a total of three transitions.
- `order.h` is an integer vector of length 3 that indicates the GARCH, ARCH and asymmetry order for each series, i.e., `order.h = (p, q, r)`, recall Equation 8. The default is `order.h = c(1,1,0)`, i.e., a standard GARCH(1,1) without asymmetry. To produce asymmetric effects, the third item (the asymmetry order), which is zero by default, needs to be changed to, e.g., `order.h = c(1,1,1)`.
- `intercept.g` is a scalar, which indicates the value of the intercept in the TV component, δ_0 . By default, `intercept.g = 1.2`. If, for instance, the time series were stationary, then there would be of course no need to specify the value of the intercept in the TV component nor any of the following three arguments.
- `size` is a vector of length k that contains the values of the size coefficients, $\delta_j, j = 1, \dots, k$. By default, `size = 5`.
- `speed` is a vector that contains the values of the speed coefficients, $\gamma_j, j = 1, \dots, k$. The default setting is `speed = 25`. For identification of the nonstationary variance model, $\gamma > 0$. However, the range of possible values for this parameter is different according to the choice of `opt`. For instance, when `opt = 2`, small negative values of the speed of transition are possible.
- `location` is a vector that contains the values of the locations of transitions, $c_{jl}, j = 1, \dots, k, l = 1, \dots, k_j$. The mid-point of change in the default model (i.e., with a single transition function) occurs at $c_{11} = 0.5$ and can be coded by setting `location = 0.5`. If more than one location parameter and/or transition function, the location values should be provided for each transition in a single vector such that $c_{j1} \leq c_{j2} \leq \dots \leq c_{jk_j}, j = 1, \dots, k$. As an example, when `order.g = 2` and `location = c(0.2,0.8)`, a TV(2) component is simulated whose logistic transition function is symmetric around $(c_{12} - c_{11})/2 = (0.8 - 0.2)/2 = 0.3$.

- `intercept.h` is a scalar, which indicates the value of the intercept in the GARCH component, ω . By default, `intercept.h` = 0.2.
- `arch` is a vector of length q that contains the ARCH coefficients used in the simulations, $\alpha_j, j = 1, \dots, q$. The default value is given by `arch` = 0.1.
- `garch` is a vector of length p that contains the values of the GARCH coefficients, $\beta_j, j = 1, \dots, p$. By default, this is set to `garch` = 0.8.
- `asym` is a vector of length r that contains the values of the asymmetry parameters, $\lambda_j, j = 1, \dots, r$. By default, no asymmetric effects are assumed, i.e., `asym` = NULL. If instead we want the conditional variance to show asymmetric effects (meaning the `order.h` has been changed such that $r \neq 0$), then the r values of the asymmetry coefficients should be provided using the `asym` argument.
- `xreg` is a vector of length n that contains the impacts of the stationary covariates. The default is NULL, i.e., no regressors.
- `opt` an integer (0, 1 or 2) that indicates whether the speed parameter in the TV component should be scaled, and if so how (more details in the help pages). The default is 0, i.e., no scaling.

The default values above are chosen according to the results usually found in financial empirical applications. Naturally, these can and should be adjusted to the empirical problem at hand.

4.3. Simulating from a multivariate TV-GARCH model

The first argument of `mtvgarchSim()` is the number of observations to simulate n , and the second argument is the number of time series to be simulated m . By default, a bivariate process is simulated, i.e., $m = 2$. Also by default, both conditional variances are nonstationary and follow a TV(1)-GARCH(1, 1) model. By contrast, the simulated conditional correlations are constant under the default, but can have the dynamic structure of Equations 10–11 if the `dcc` argument of the `mtvgarchSim()` function is switched to TRUE by the user. A combination of stationary and nonstationary variance equations with or without leverage effects and/or covariates is possible by user-specifying the other arguments in `mtvgarchSim()`. For example, the following code simulates a bivariate time series, where the first follows a nonstationary TV(1)-GARCH(1, 1) model and the second a nonstationary TV(1)-GARCH(1, 1, 1) model (or TV(1)-GJR(1)-GARCH(1, 1) following the literature on GARCH models with asymmetric effects introduced by [Glosten, Jagannathan, and Runkle 1993](#)), and stores the simulated series in an object called `mSim`:

```
R> mSim <- mtvgarchSim(n = 1000, order.h = c(1,1,0,1,1,1), asym = 0.01)
```

It is straightforward to change the `innovations` argument in `mtvgarchSim()`, in which case an $n \times m$ matrix of innovations can be provided rather than assuming random draws from a standard normal. Other relevant arguments are explained below and more examples are provided. For notational convenience, the first subscript identifies the time series. If only two subscripts are shown, then the second subscript identifies the parameter. Whenever three

subscripts are necessary (if more than one location parameter is assumed), then the second subscript identifies the transition function while the third is used to identify the parameter.

- **order.g** is an $m \times \max(\{k_i\}_{i=1}^m)$ integer matrix (a vector can also be provided in which case the `matrix()` function with the argument `byrow = TRUE` is then applied) that indicates the number of locations of transition in each logistic transition function of the TV components. By default, **order.g** = `c(1, 1)`. This means the conditional variance of each time series is nonstationary with a single transition function with one location parameter. If, for instance, **order.g** = `c(1,0)`, this means the conditional variance of the first time series remains the same as in the default model, but the conditional variance of the second one is now stationary.
- **order.h** is an $m \times 3$ matrix of integers where the first column controls the GARCH order, the second the ARCH order and the third the asymmetry order of each GARCH-X component. By default, a standard GARCH(1,1) process is assumed for each times series. In the code example above, where **order.h** = `c(1,1,0,1,1,1)`, the conditional variance of the second time series contains asymmetric effects, but not the first conditional variance.
- **intercept.g** is a vector of length m that contains the values of the intercepts in the TV components. By default, the intercepts in each TV(1) component are set to $\delta_{10} = 1.2$ and $\delta_{20} = 1$ (i.e., **intercept.g** = `c(1.2, 1)`). If, for instance, the second time series were stationary, then there would be no need to specify the value of the intercept in the TV component of the second series.
- **size** is an $m \times \max(\{k_i\}_{i=1}^m)$ numeric matrix which provides the values of the size coefficients, $\delta_{ij}, i = 1, \dots, m, j = 1, \dots, \max(\{k_i\}_{i=1}^m)$. By setting **size** = `c(3, 5)`, the size for the first (and single) transition is $\delta_{11} = 3$ and $\delta_{21} = 5$ for, respectively, the first and second series.
- **speed** is similar to **size** and provides the values of the speed coefficients, $\gamma_{ij}, i = 1, \dots, m, j = 1, \dots, \max(\{k_i\}_{i=1}^m)$. By setting **speed** = `c(10, 25)`, the speed of transition in the first (and single) transition of the first time series, $\gamma_{11} = 10$ and for the second time series, $\gamma_{21} = 25$.
- **location** is an $m \times \max(\{k_i\}_{i=1}^m)$ matrix that includes the values of the location of transition in each transition function. To get some insight on the form of this matrix, consider a single transition function for both times series with two locations in the first and one in the second. To simulate a bivariate series with nonstationary variance equations of that form, **location** should be a 2×2 matrix with a zero value assigned to the second location of transition in the logistic transition function of the second series. Whenever a location of transition does not apply, the value attributed should be zero (even though the value will not be used, in practice). The mid-point of change for the example above with one location of transition is the default, i.e., $c_{11} = 0.5$ for the first series and $c_{21} = 0.8$ for the second series.
- **intercept.h** is a vector of length m that contains the values of the intercepts in the GARCH component, $\omega_i, i = 1, \dots, m$. By default, **intercept.h** = `c(0.2,0.3)`.

- **arch** is the $m \times \max(\{q_i\}_{i=1}^m)$ matrix containing the ARCH coefficients used in the simulations, $\alpha_{ij}, j = 1, \dots, q_j$. The default values are given by **arch** = `c(0.1, 0.05)`
- **garch** is the $m \times \max(\{p_i\}_{i=1}^m)$ matrix containing the values of the GARCH coefficients, $\beta_{ij}, j = 1, \dots, p_j$. By default, this is set to **garch** = `c(0.8, 0.9)`.
- **asym** is an $m \times \max(\{r_i\}_{i=1}^m)$ matrix containing the values of asymmetry parameters, $\lambda_{ij}, j = 1, \dots, r_j$. Because only the second series assumes asymmetry effects of news on the conditional variance, **asym** = `0.01`.
- **R** is the matrix of constant conditional correlations, $\rho_{ijt}, i, j = 1, \dots, m$, with $\rho_{ijt} = 1$ when $i = j$. The unique pairwise correlation between the two time series is, by default, assumed to be 0.6. This is simply coded as **R** = `c(1, 0.6, 0.6, 1)`.

To make use of other arguments which allow us to include covariates in the model, let us first generate a set of two covariates. The following code generates 1000 observations for two random variables with mean zero and unit variance, squares them and stores them in an object called **mX**:

```
R> mX <- matrix(rnorm(1000*2)^2, 1000, 2)
```

xreg is the argument that provides the set of stationary covariates. Next, the following code simulates a similar multivariate TV-GARCH-X model but now with two covariates in the GARCH-X component of the first time series. The coefficients for each covariate are, respectively, $\zeta_{11} = 0.01$ and $\zeta_{12} = 0.02$. The simulated series are then stored in an object called **mSim2**:

```
R> mSim2 <- mtvgarchSim(n = 1000, order.x = c(1,0,0,1), xreg = mX,
+   par.xreg = c(0.01, 0.02))
```

In some applications, there may be the case that past information driving the volatility of one time series helps to predict the variance of other series. Volatility spillovers occur when past squared returns (ARCH) and/or variances (GARCH) of a series affect the variance of another return series. Since estimation in the package **tv****garch** is performed equation by equation, volatility spillovers are allowed only through the ARCH terms, i.e., past squared returns. Provided that the argument **xreg** = `NULL`, the default, and **order.x**, an $m \times m$ binary matrix, is non-zero, volatility spillovers can be accommodated in the GARCH-X component as additional covariates. In that setting, the covariates in the GARCH-X component include lagged squared returns of the multivariate data. Note that, when assuming volatility spillovers, only first-order lagged returns, variances and asymmetry terms are possible (i.e., maximum **order.h** = `(1,1,1)`). The covariates $\mathbf{x}_t = \phi_{t-1}^2$ will then be used to account for the volatility spillovers. The covariates to be included in the GARCH-X component of each time series are selected by the non-zero entries of the **order.x** argument. (Note that the diagonal elements of **par.xreg** should all be zero, even though they are never used in practice.) If **order.x** has non-zero off-diagonal elements and **xreg** = `NULL`, then volatility spillovers are included as covariates multiplied by the coefficients provided in **par.xreg**. This argument is an $m \times m$ numeric matrix containing the coefficients of the covariates, in this case, the cross-series information. Volatility spillovers are allowed in the multivariate **mtvgarchSim()** and **mtvgarch()** functions.

As an example, consider a bivariate TV-GARCH-X model. Suppose the first-order lagged squared standardized returns of the second series affect the GARCH-X component of the first series. Moreover, assume that the spillover coefficient, i.e., the parameter associated to variable $\phi_{2,t-1}^2$ in the GARCH-X component of the first series, ζ_{12} , is equal to 0.02. The GARCH(1,1)-X component for the first time series thus has the form

$$h_{1t} = 0.2 + 0.1\phi_{1,t-1}^2 + 0.8h_{1,t-1} + 0.02\phi_{2,t-1}^2.$$

The following code simulates such bivariate time series with (uni-directional) volatility spillovers and stores it in an object called `mSim3`:

```
R> mSim3 <- mtvgarchSim(n = 1000, order.x = c(0,1,0,0), par.xreg = 0.02,
+   verbose = TRUE)
```

By changing the logical argument `verbose`, the object `mSim` stores other information and simulated series (e.g., the innovations). It is possible to check which variables are included in each GARCH-X component and all the parameter values used in the simulations. For instance, considering the example above and switching `verbose = TRUE`, the GARCH-X parameters used to simulate the two time series with volatility spillovers can be printed by typing `mSim3$par.h`:

```
      intercept arch1 garch1 lag(y2^2)
y1          0.2  0.10   0.8   0.02
y2          0.3  0.05   0.9   0.00
```

5. Empirical illustration

This section contains three parts. The first, Section 5.1, introduces the data we use for our illustrations. Next, Section 5.2 provides a detailed illustration of the volatility modeling of a single return series. Finally, Section 5.3 illustrates the estimation of a multivariate model, both its volatilities and its dynamic conditional correlations.

5.1. The data

The financial data we use for our illustrations are from the datasets `ftse100`, `sp500` and `nasdaq` in the R package `dccmidas` (Candila 2024). The datasets contain the daily open and close prices – as well as their realized variances (computed using intra-day 5-minute returns, see (Andersen and Bollerslev 1998)) – of the three major stock indices FTSE 100, S&P 500 and NASDAQ. For simplicity, we merge the closing prices of the three stocks so they all cover the same sample period, and we remove rows with missing values. To do this, we use the following code, which also saves the three series in a time series object called `stocks_p`:

```
R> library("dccmidas")
R> stocks_p <- merge(ftse100[, "close_price"], sp500[, "close_price"],
+   join = "left")
R> stocks_p <- merge(stocks_p, nasdaq[, "close_price"], join = "left")
R> stocks_p <- na.omit(stocks_p)
R> colnames(stocks_p) <- c("ftse100", "sp500", "nasdaq")
```

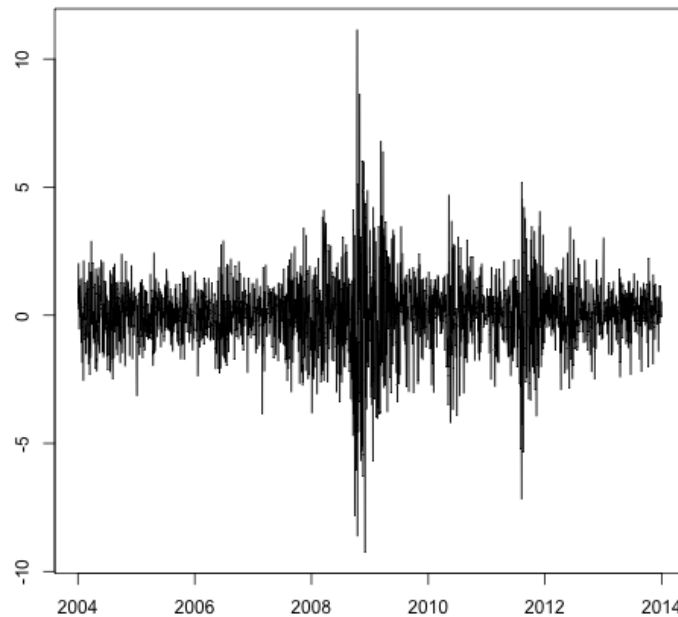


Figure 3: The daily log-returns (in percent) on the NASDAQ index.

For very long sample periods, more than one structural change or shift is to be expected in the unconditional variance of returns. The modeling strategy of [Amado and Teräsvirta \(2013, 2017\)](#), where the estimation and testing procedures are applied to the full sample, works well for a small number of transitions. When the parameters in the GARCH model are very unstable making the unconditional variance change often, a solution to guarantee all transitions are captured is to divide the series into smaller periods, applying the strategy to them and combining the results into a single model; see [Amado and Teräsvirta \(2014\)](#). To keep the illustration simple, we only consider the period around the 2008 global financial crisis (i.e., excluding the observations around the *dot.com* bubble and the recent COVID-19 pandemic). We thus consider the daily prices between December 31, 2003 and December 31, 2013. This is done as follows:

```
R> stocks_p <- stocks_p[index(stocks_p) >= "2003-12-31 GMT", ]
R> stocks_p <- stocks_p[index(stocks_p) < "2014-01-01 GMT", ]
```

Finally, the following code computes and saves the log-returns in % (i.e, multiplied by 100):

```
R> stocks_r <- (diff(log(stocks_p)) * 100)[-1, ]
```

where the first observation is dropped for a new total of 2466 observations.

5.2. Modeling a single series of stock returns

To illustrate the modeling of a univariate time series, we focus on the daily log-returns of the `nasdaq` index. The code to select this series can be

```
R> nasdaq_r <- stocks_r[, "nasdaq"]
```

The series of daily log-returns on the NASDAQ index, `nasdaq_r`, is plotted in Figure 3. Two characteristics are clear from the figure. The first is the presence of volatility clustering: large (small) returns tend to be followed by large (small) returns, which provides a clear indication that the conditional variance is not constant. The second characteristic is that the returns are systematically more volatile during the 2008 global financial crisis. This may suggest that also the unconditional variance is not constant. The larger returns will be reflected in higher volatility during this crisis period, compared to the periods at the beginning and end of the sample period, where volatility is expected to be rather low and stable. The global financial crisis may have not only caused the conditional variance to change, but also the unconditional variance.

Identifying the number of transitions

Before estimating a TV-GARCH model, we first need to identify the number of transitions or changes in unconditional variance. This is important, since the estimator of the TV-GARCH model is not identified (asymptotically) if it specifies more transitions than there actually is (i.e., assumption AG2 in Amado and Teräsvirta 2013 does not hold). The function `tvgarchTest()` implements a procedure proposed by Amado and Teräsvirta (2017) to identify the number of transitions. The procedure consists of a sequence of Lagrange-multiplier (LM) tests. Under the initial null hypothesis, the unconditional variance is constant and a stationary GARCH model with constant parameters is sufficient to describe the dynamics of the conditional variance. Under the alternative, the unconditional variance is time-varying and should be modeled as a smoothly changing function of the transition variable used to perform the test. The results from `tvgarchTest()` can thus be used to guide the choice of the number of transitions in specifying a suitable TV-GARCH model. More details about the test is provided in Appendix D.

The following code implements the identification procedure for the `nasdaq` return series, `nasdaq_r`, using scaled time as the transition variable (default):

```
R> tvgarchTestObj <- tvgarchTest(nasdaq_r)
```

which returns an object of class "tvgarchTest". To extract the number of transitions suggested by the procedure, use function `summary()`. To extract the estimation results of the model under the initial null hypothesis, i.e., the stationary GARCH(1,1) model, extraction functions are available similar to those for objects of class "tvgarch". See *Post estimation* below for more details. By typing `tvgarchTestObj`, the GARCH(1,1) estimation results, which include the estimated parameters, the standard errors and the optimized log-likelihood value, and all test results are all printed as follows:

```
Date: Tue Jul  4 10:39:44 2023
Testing GARCH(1,1), i.e., the model under H0, against
  TV-GARCH(1,1), i.e., the model under H1:

Estimation results for model under H0:
Model: GARCH(1,1)

              intercept.h      arch1      garch1
Estimate:      0.0213933 0.0714765 0.9139542
```

Std. Error: 0.0065884 0.0113774 0.0125204

Log-likelihood: -3816.104

Transition variable in TV-GARCH(1,1): time

Results from the Non-Robust TR² Test:

	NonRobTR2	p-value
H0:B3=B2=B1=0	10.7801	0.0130
H03:B3=0	3.3314	0.0680
H02:B2=0 B3=0	7.2128	0.0072
H01:B1=0 B3=B2=0	0.2467	0.6194

Results from the Robust TR² Test:

	RobTR2	p-value
H0:B3=B2=B1=0	10.8985	0.0123
H03:B3=0	3.5003	0.0614
H02:B2=0 B3=0	7.1331	0.0076
H01:B1=0 B3=B2=0	0.7243	0.3947

Single

No. of locations (alpha = 0.05) 2

The results above suggest the unconditional variance is not constant, and that it should be modeled by two transitions.

Both versions of the TR² test are reported, namely the non-robust and the robust test statistics. Recall, here, “robust” means robustness to non-normality of the innovations, i.e., the η_t 's. For each of the two versions, the first result reported corresponds to testing the hypothesis that all three coefficients are zero and the subsequent results from testing the hypothesis that each coefficient is zero conditionally on the other coefficients being also zero. See Appendix D for more details. Henceforth, we focus on the robust results.

The significance level is set via the `alpha` argument of `tvgarchTest()`. By default, this is 0.05. When the p value from the general test exceeds `alpha`, the test fails to reject the null hypothesis \mathbb{H}_0 , which means the transition variable selected provides no evidence for variance nonstationarity. The function returns a zero as there is no transition and so no locations of transition. If, however, the p value is less than the selected `alpha`, \mathbb{H}_0 can be rejected providing evidence for nonstationary variance according to the selected transition variable defined by `xtv`. The observed p value from this general misspecification test is 0.0123, which is smaller than `alpha`. We therefore reject the general null hypothesis that all coefficients in Equation 12 are zero at the 5% confidence level. This result thus provides evidence in favor of time-varying parameters and so that variance is nonstationary. Conditionally on the rejection of \mathbb{H}_0 , the `tvgarchTest()` function tests and prints the sequence of nested hypotheses in Equation 13 to determine the number of locations of transition in the logistic transition function. The selection rule for k_1 is based on the lowest p value. If \mathbb{H}_{03} is rejected with a lower (higher) p value than \mathbb{H}_{01} , then $k_1 = 3$ ($k_1 = 1$), which is the value printed as output. If, however,

\mathbb{H}_{02} is rejected with a lower p value than the other two, then $k_1 = 2$. In this empirical example, the lowest p value is 0.0076, which corresponds to two locations of transition. The `tvgarchTest()` returns the number of locations of transition in the single logistic transition function for a transition variable defined by the argument `xtv`. We now proceed with the modeling of the conditional variance by estimating a TV(2)-GARCH-X model, where the structure of the GARCH-X component has to be determined. Information criteria and/or visual inspection of the series can be useful in selecting the appropriate conditional variance model.

Some additional notes on the testing strategy. Notice that the test statistics are computed conditional on the choice of the transition variable, s_t , meaning that rejection of the constancy hypothesis is valid for a particular transition variable. By default, `xtv = NULL` meaning $s_t = t/n$ and the logistic transition function is deterministic. By changing the `xtv` argument, any continuous transition variable can be used, including stochastic variables.

Estimation

In the `tvgarch()` function, the `order.h` argument specifies the GARCH(p, q, r) part of Equation 8, and `order.g` specifies the g_t component to be estimated. Both arguments have the same structure as in `tvgarchSim()`, see Section 4.2.

The test procedure above suggests there are two transitions. To estimate a TV(2)-GARCH(1, 1) model in which the transition size and speed are the same at each location, recall Equation 5, simply set `order.g` equal to 2:

```
R> tvgarchEst <- tvgarch(y = nasdaq_r, order.g = 2)
```

The estimation results are thus saved in an object of class "tvgarch" called `tvgarchEst`, and typing `tvgarchEst` prints:

```
Date: Tue Jul 4 10:42:45 2023
Model: TV(2)-GARCH(1,1)
Method: maximization by parts
No. of iterations: 49
No. of observations: 2466

* TV specification (long-term component) *

Optimization: linearly-constrained
Transition variable: time

intercept: 3.433546 (fixed from first iteration)

              size1    speed1 location1 location2
Estimate:    -2.134989  5.5214574  0.3602065  0.790451
Std. Error:   0.062759  0.5822143  0.0069976  0.004645

* GARCH specification (short-term component) *
```



```
Optimization: box-constrained
Message (nlminb): relative convergence (4)
```

```
            intercept      arch1      garch1
Estimate:    0.0137868 0.0647036 0.9159871
Std. Error:  0.0041438 0.0110393 0.0131253
```

```
Log-likelihood: -3812.191
```

Post estimation

The iterative optimization algorithm for fitting the TV(2)-GARCH(1,1) model reached convergence after 49 iterations, but notice that this and the other estimation result may vary slightly across operating systems, software versions and other configurational settings (recall the section on reproducibility in the Introduction). The estimated parameters and corresponding standard errors are printed for each component of the conditional variance. In the TV component, the estimate of the intercept is $\hat{\delta}_0 = 3.433546$, and the estimate of the size parameter is $\hat{\delta}_1 = -2.134989$. To help understand how the estimated variance from the TV component evolves over time, it is useful to have a look at the estimated logistic transition function plotted in the top left graph of Figure 4. The estimated deterministic component, \hat{g}_t , is the blue or smoother of the lines in the top right graph of the figure. At the beginning of the sample, it is equal to $\hat{\delta}_0 + \hat{\delta}_1 = 1.298557$ (i.e., when $\hat{G}_t = 1$). Then it starts to increase some months before the most acute phase of the global financial crisis. It remains at this higher level (approximately equal to $\hat{\delta}_0$) during the crisis until it eventually evolves in a descending way towards the initial level.

To date the transitions, the following code extracts the closest possible value of the transition variable `xtv` used in the estimation called `stime2` to the estimated first location of transition, \hat{c}_{11} :

```
R> coef.g <- coef(tvgarchEst, spec = "tv")
R> c11 <- coef.g["location1"]
R> stime2 <- tvgarchEst$xtv
R> stime2[which.min(abs(stime2 - c11))]
```

which returns:

```
2007-08-09
0.3600973
```

The first transition has been estimated to have a mid-point of change (the change being an increase in variance) on 9 August 2007. This is the day the mortgage crisis went global (Barr 2017). The second location of transition, \hat{c}_{12} , corresponds to 21 November 2011, the mid-point of a decrease in the variance towards the initial level. Thereafter, the variance remains low and fairly stable.

The estimated `speed` parameter is relatively high meaning the transition (i.e., break) is fairly abrupt or fast. By default, this parameter is scaled in the estimation, i.e., `opt = 2`. To obtain the estimated value of γ_1 , we can simply compute $\hat{\gamma}_1 = \exp(5.5214574)$, which is

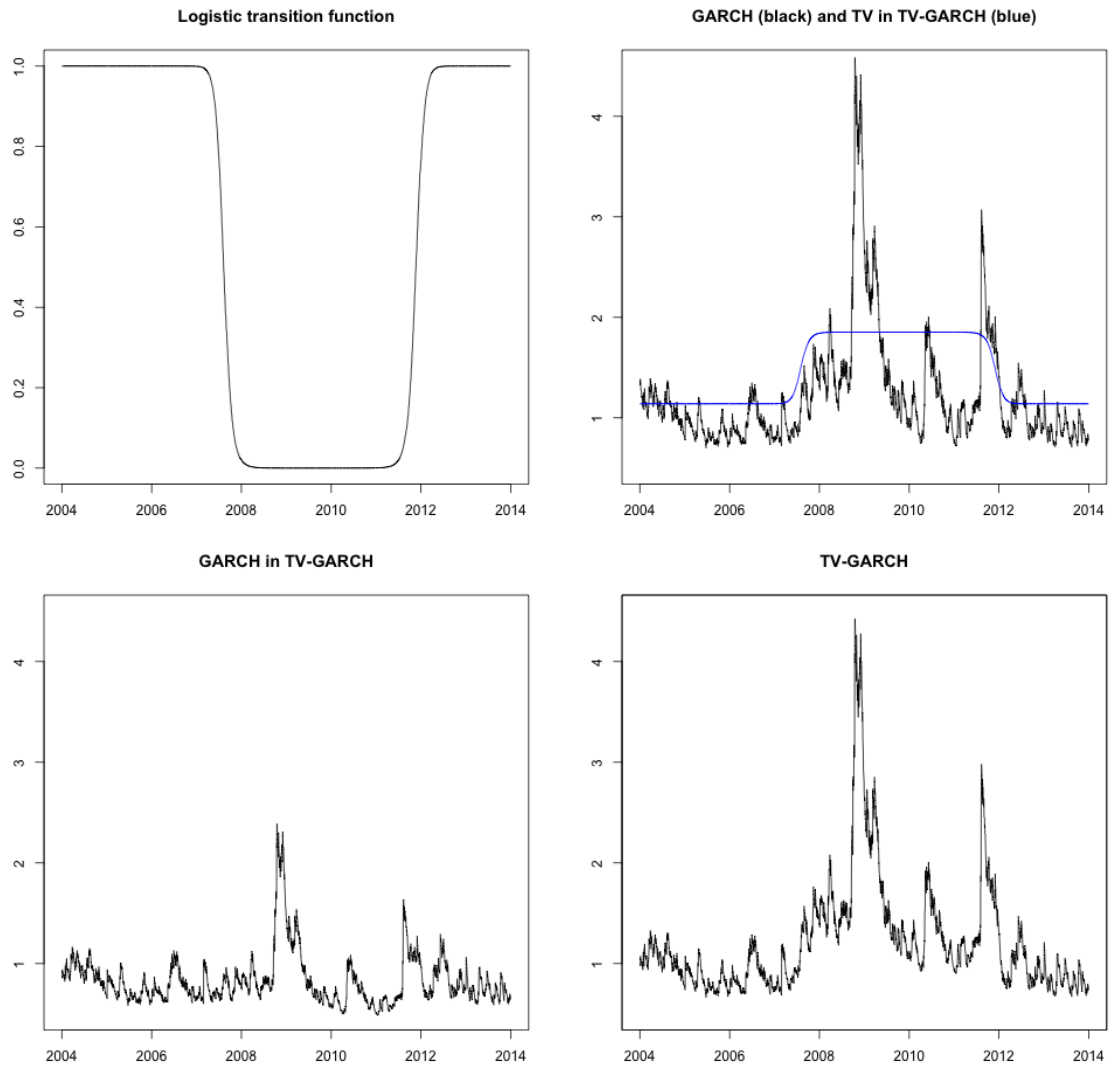


Figure 4: The estimated logistic transition function (upper left) and conditional standard deviation (lower right) from a TV(2)-GARCH(1,1) model as well as its TV component (upper right, blue) and the GARCH component (lower left). For comparison, the conditional standard deviation from a stationary GARCH(1,1) model is also shown (upper right, black). All plots for the NASDAQ index returns.

approximately equal to the default upper bound, i.e., 250. By changing the `opt` argument of function `tvgarch()`, we can opt for no scaling if `opt = 0`. Nevertheless, results (not shown here to save space) are similar. The default upper bound of 250 is due to numerical considerations, but can be changed by the user via the `upper.speed` argument in `tvgarch()`.

To plot the estimated conditional standard deviation (and/or its components), we can use the extraction function `plot.tvgarch()`. Applying it on an object of class "tvgarch" returns three plots with the estimated conditional standard deviations, i.e., from the TV-GARCH model and its two components, where `spec = "tvgarch"` is the default. To get individual plots instead, type

```
R> plot(tvgarchEst, spec = "tv", ylim = c(0.5,4.5), main = "")
R> plot(tvgarchEst, spec = "garch", ylim = c(0.5,4.5))
R> plot(sqrt(fitted(tvgarchEst)), ylim = c(0.5,4.5), type = "l",
+       ylab = "", xlab = "")
R> title("TV-GARCH")
```

where we opt for plots to have all the same scale. These are shown in Figure 4 below.

The conditional standard deviation estimated assuming a TV(2)-GARCH(1, 1) model is plotted in the lower right panel. For comparison, the conditional standard deviation estimated assuming a stationary GARCH(1, 1) model (black) is plotted alongside the conditional standard deviation from the TV component of the TV(2)-GARCH(1, 1) model (blue) in the upper right panel. The TV component looks similar to a smoothed conditional mean of the estimated GARCH component. The baseline variance appears to have persistently increased months before 2008 and remained at this higher level during the global financial crisis. It eventually returned to the level observed before the crisis. The estimated conditional standard deviation from the GARCH component of the TV(2)-GARCH(1, 1) model plotted in the lower left panel of Figure 4 is obtained by standardizing the raw returns by the square root of the estimated TV component. As can be observed, volatility appears more stable. Ideally, a statistical test should be used to test this post estimation. In particular, to test whether additional transition functions are necessary. Such tests are not yet available in **tv**garch, but other post estimation analyses are possible as outlined below.

To report the results in equation form, the extraction function `toLatex()` can be used. Simply type `toLatex(tvgarchEst)` to obtain the following after compilation of the L^AT_EX code:

$$\hat{h}_t = 0.0138 + 0.0647 \frac{y_{t-1}^2}{\hat{g}_{t-1}} + 0.9160 \hat{h}_{t-1}$$

(0.0041) (0.0110) (0.0131)

$$\hat{g}_t = 3.4335 - 2.1350 \hat{G}_1(5.5215, 0.3602, 0.7905; t/n)$$

(-) (0.0628) (0.5822) (0.0070) (0.0046)

Log-likelihood: - 3812.1914 n = 2466

Additional extraction functions include:

- `coef(tvgarchEst)`: Coefficient estimates. The argument `spec` can be used to specify whether the function should only extract the optimal parameters in the TV component (i.e., by choosing `spec = "tv"`) or in the GARCH-X component (in which case, `spec = "garch"`). For all coefficient estimates, `spec = "tv`garch". Only relevant for TV-GARCH-X models.
- `fitted(tvgarchEst)`: Fitted conditional variance, $\hat{\sigma}_t^2, t = 1, \dots, n$, by default, i.e., when `spec = "tv`garch". To extract only the TV component, \hat{g}_t , choose `spec = "tv"`. For the GARCH-X component, \hat{h}_t , select `spec = "garch"` instead.
- `logLik(tvgarchEst)`: Optimized log-likelihood value.
- `nobs(tvgarchEst)`: Number of observations used in the estimation.
- `plot(tvgarchEst)`: Plots of the estimated conditional standard deviation. If `spec = "tv"`, it plots the conditional standard deviation from the TV component in the (nonstationary) TV-GARCH-X model alongside the conditional standard deviation estimated

assuming a (stationary) GARCH-X model for comparison. If `spec = "garch"` instead, then the conditional standard deviation from the GARCH-X component in the TV-GARCH-X model is plotted. If `spec = "tvgarch"`, the two aforementioned plots can be obtained alongside the (full) conditional standard deviation from the TV-GARCH-X model.

- `predict(tvgarchEst)`: Predictions of the conditional variance. When conditional variance is nonstationary, the user can provide the new values for the transition variable using the argument `newxtv`. If left unchanged, then the predicted TV component of the conditional variance will simply be equal to its `intercept`, i.e., $\hat{g}_{t+h} = \hat{\delta}_0$, where h denotes the period ahead. The predicted values for the GARCH component are obtained similarly to those obtained by using the extraction function `predict.garchx()` in package `garchx`; see [Sucarrat \(2021\)](#). When explanatory variables are included, the new values must be provided using the `newxreg` argument.
- `print(tvgarchEst)`: Print of estimation results (parameters and standard errors).
- `quantile(tvgarchEst)`: Fitted quantile(s), the default corresponds to 97.5% value-at-risk. This function for objects of class "tvgarch" is similar to function `quantile.garchx()` in `garchx`; see [Sucarrat \(2021\)](#).
- `residuals(tvgarchEst)`: Volatility standardized residuals.
- `summary(tvgarchEst)`: A summary of the estimation.
- `toLatex(tvgarchEst)`: \LaTeX print of result (equation form).
- `vcov(tvgarchEst)`: Full coefficient-covariance when `spec = "tvgarch"`. The argument `spec` can be used to specify whether the function should only extract the covariance matrix of the parameters in the TV component (i.e., by choosing `spec = "tv"`) or in the GARCH-X component (in which case, `spec = "garch"`).

5.3. Modeling multivariate stock returns

As an empirical application of the function `mtvgarch()`, we consider a multivariate time series comprising `ftse100`, `sp500` and `nasdaq`. For modeling the conditional variances, we implement the procedure described in the previous section equation by equation. For this reason, we will omit the variance results for `nasdaq` and focus on the two other stock return series in this section before we proceed to the estimation of the conditional correlations. The series of closing prices are converted into log-returns for all indices.

We first test the constancy of the parameters in the GARCH component for every constituent series using the function `tvgarchTest()`. This can be done with the following loop, where the number of locations of parameters in the TV component is stored for each series in a vector called `stocks_k`:

```
R> m <- ncol(stocks_r)
R> stocks_k <- matrix(0, m, 1)
R> for (i in 1:m) {
+   stocks_k[i] <- summary(tvgarchTest(stocks_r[, i]))
+ }
```

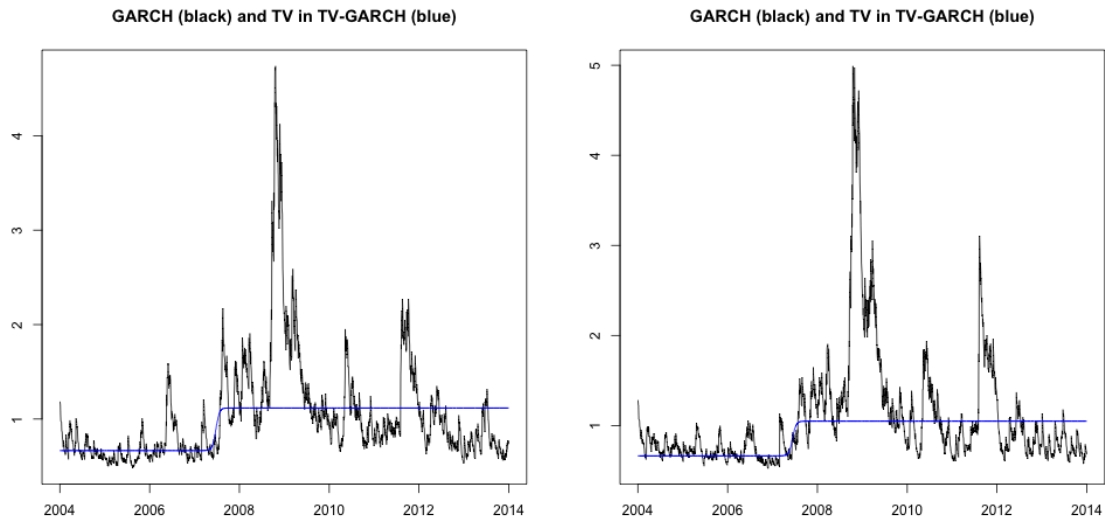


Figure 5: The estimated volatility from a stationary GARCH(1, 1) model (black) and the TV component from the TV(1)-GARCH(1, 1) model (blue) for the FTSE 100 (left) and the SP 500 (right) index returns.

```
R> rownames(stocks_k) <- colnames(stocks_r)
R> colnames(stocks_k) <- "k1"
```

To check the number of location parameters in the single transition function of each series, type `stocks_k`, which gives

```
      k1
ftse100 1
sp500    1
nasdaq   2
```

According to the test results, the stationary GARCH model is misspecified for all time series and a TV-GARCH model would be a more appropriate specification for the conditional variance. Specifically, the test results suggest a TV(1)-GARCH(1, 1) model should be estimated for `ftse100` and `sp500`, and a TV(2)-GARCH(1, 1) model for `nasdaq`. The different TV-GARCH models are estimated, equation by equation, followed by the constant conditional correlations and the results are saved in an object of class "mtvgarch" called `ccctvgarchEst` using the code:

```
R> ccctvgarchEst <- mtvgarch(y = stocks_r, order.g = stocks_k)
```

Similar to the univariate case, we can plot the estimated conditional standard deviation using `plot.mtvgarch()`. This function calls the equivalent function for the univariate case, i.e., `plot.tvgarch()` for each object of class "tvgarch" (m objects will be created within `mtvgarch()`). By using the following,

```
R> plot(ccctvgarchEst, spec = "tv")
```

we obtain three plots, one for each times series of stock returns in the multivariate sample. By choosing `spec = "tv"`, each plot shows the estimated conditional standard deviation from a stationary GARCH(1, 1) model (black) and the TV component of the TV(1)-GARCH(1, 1) model (blue). Figure 5 presents the plots for `ftse100` (left) and `sp500` (right), where the plot for `nasdaq` has been omitted as this one is already shown in the upper panel of Figure 4.

Similar to the univariate case, to print the results from estimating a multivariate TV-GARCH-X model with `mtvgarch()`, we can simply type `ccctvgarchEst`, use `print.mtvgarch()` or `summary.mtvgarch()`. The estimation results of each constituent stock return series is printed together with the constant or dynamic conditional correlation results (depending on the logical argument `dcc` of `mtvgarch()`). Alternatively, by typing

```
R> toLatex(ccctvgarchEst)
```

we obtain the following estimated variance equations in L^AT_EX format:

$$\begin{aligned} \hat{h}_{1t} &= \underset{(0.0080)}{0.0223} + \underset{(0.0163)}{0.0895} \frac{y_{1,t-1}^2}{\hat{g}_{1,t-1}} + \underset{(0.0201)}{0.8921} \hat{h}_{1,t-1} \\ \hat{g}_{1t} &= \underset{(-)}{0.4429} + \underset{(0.0486)}{0.7906} \hat{G}_1(5.5215; 0.3482; t/n) \\ &\text{Log-likelihood: } -3323.4869 \quad n = 2466 \\ \hat{h}_{2t} &= \underset{(0.0075)}{0.0266} + \underset{(0.0121)}{0.0783} \frac{y_{2,t-1}^2}{\hat{g}_{2,t-1}} + \underset{(0.0131)}{0.9015} \hat{h}_{2,t-1} \\ \hat{g}_{2t} &= \underset{(-)}{0.4401} + \underset{(0.0474)}{0.6587} \hat{G}_1(5.5215; 0.3448; t/n) \\ &\text{Log-likelihood: } -3407.4655 \quad n = 2466 \\ \hat{h}_{3t} &= \underset{(0.0041)}{0.0138} + \underset{(0.0110)}{0.0647} \frac{y_{3,t-1}^2}{\hat{g}_{3,t-1}} + \underset{(0.0131)}{0.9160} \hat{h}_{3,t-1} \\ \hat{g}_{3t} &= \underset{(-)}{3.4335} - \underset{(0.0628)}{2.1350} \hat{G}_1(5.5215; 0.3602, 0.7905; t/n) \\ &\text{Log-likelihood: } -3812.1914 \quad n = 2466 \end{aligned}$$

Constant conditional correlations:

	ftse100	sp500	nasdaq
ftse100	1	0.568	0.5284
sp500	0.568	1	0.9373
nasdaq	0.5284	0.9373	1

Table 2: Constant conditional correlations.

For comparison, stationary GARCH models could also be estimated, equation by equation, by changing argument `order.g = NULL`. (For brevity, we do not report these results.)

To estimate dynamic conditional correlations, simply change the `dcc` argument as follows, saving the results in an object called `dcctvgarchEst`:

```
R> dcctvgarchEst <- mtvgarch(stocks_r, order.g = stock_k, dcc = TRUE)
```

By typing `dcctvgarchEst`, the estimation results for the dynamic conditional correlations (omitting the estimation results for the conditional variances, which are equal to those shown previously) are:

```
*****
* The Volatility Component *
*****
```

[omitted to save space]

```
*****
* The Correlation Component *
*****
```

Dynamic Conditional Correlations

Q matrix

```
          ftse100  sp500  nasdaq
ftse100  1.0000  0.5680  0.5284
sp500    0.5680  1.0000  0.9373
nasdaq   0.5284  0.9373  1.0000
```

Coefficients

```
          alpha  beta
Estimate:  0.0150 0.9821
Std. Error: 0.0046 0.0062
```

Log-likelihood: -7341.978

The estimated correlation coefficients of the $\bar{\mathbf{Q}}$ matrix in Equation 10 is simply the constant conditional correlation matrix, which is obtained by computing the empirical correlation matrix of the volatility standardized residuals in the objective function `dccObj()`. The empirical correlation matrix in an object of class "mtvgarch" corresponds to the correlation matrix $\mathbf{P}_t \equiv \mathbf{P}$ if constant conditional correlations are assumed or to $\bar{\mathbf{Q}}$ in Equations 10–11 when dynamic conditional correlations are assumed instead.

The estimated parameters `alpha` and `beta` correspond to the parameters θ_1 and θ_2 in Equation 10. As usually found in empirical applications, the autoregressive coefficient `beta` is much larger than `alpha`. Also, the dynamic conditional correlations appear to be very persistent. In fact, correlation persistence is very close to unity ($\hat{\theta}_1 + \hat{\theta}_2 = 0.9970769$).

The estimated constant and dynamic conditional correlations can be obtained and stored in an object called, respectively, CCC and DCC by using the extraction function `fitted()` as follows:

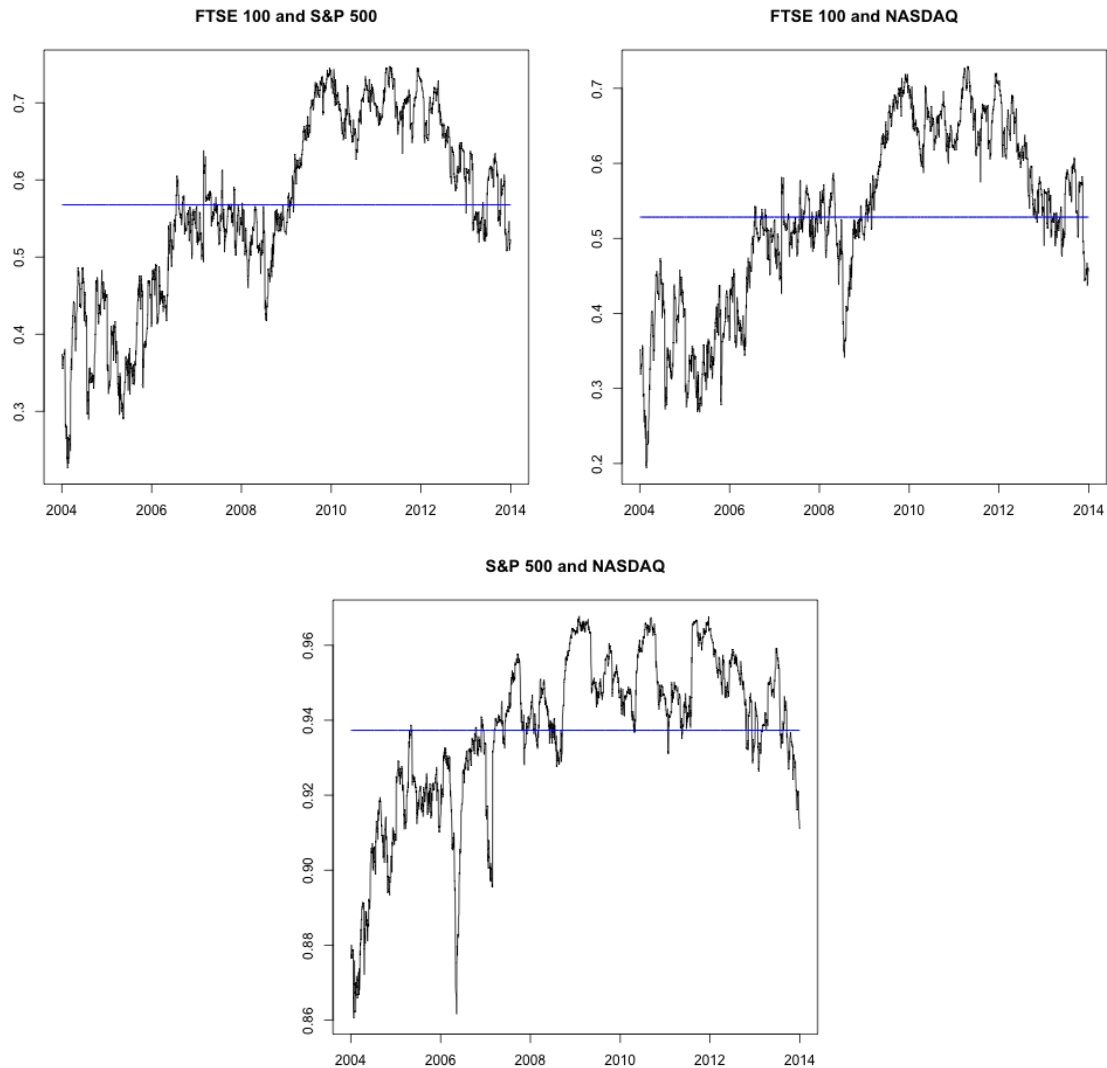


Figure 6: The estimated dynamic conditional correlations from the DCC-TV-GARCH model of stock index returns. For comparison, the constant conditional correlations (blue) from the CCC-TV-GARCH model are also shown.

```
R> CCC <- fitted(ccctvgarchEst, spec = "cc")
R> DCC <- fitted(dcctvgarchEst, spec = "cc")
```

for objects `ccctvgarchEst` and `dcctvgarchEst`. These are plotted in Figure 6 (black). For comparison, we also plot the estimated constant conditional correlations (used as the baseline correlations in the function `dccObj()`, the function called when computing the dynamic conditional correlations). It is interesting to visualize how the dynamic conditional correlations not only are persistent but also how they fluctuate around their long-term mean (as measured by the constant (un)conditional correlations).

To compare the performance of each model, i.e., with constant and dynamic conditional correlations, we can use the optimized log-likelihood value reported for each object of class `"mtvgarch"`. This value is -7462.079 when we assume constant conditional correlations

and -7341.978 for the dynamic conditional correlation model. The higher log-likelihood value for the DCC-TV-GARCH model compared to the CCC-TV-GARCH indicates higher performance, which is expected given the former assumes a more flexible correlation structure. Nonetheless, the DCC-TV-GARCH model also includes two extra parameters, namely θ_1 and θ_2 , which need to be estimated. Simple comparison of the optimized log-likelihood values can thus be misleading.

To properly compare the two models, the Bayesian information criterion (BIC) can be used. The lowest BIC corresponds to the best fit among the models being compared. The BIC can be computed according to the following code for the two models::

```
R> BIC(logLik(ccctvgarchEst)$ccc)
R> BIC(logLik(dcctvgarchEst)$dcc)
```

The BICs are 15095.99 and 14871.4, respectively, for the CCC-TV-GARCH and the DCC-TV-GARCH models, respectively. This supports our observation that assuming dynamic conditional correlations would provide a better fit to the data.

6. Conclusions

Stationary volatility models can be inappropriate and volatility predictions can fail in the presence of structural changes in the unconditional variance. To overcome this situation, in the TV-GARCH model, the standard GARCH parameters are allowed to vary smoothly over time according to a linear combination of logistic transition functions.

Simulating TV-GARCH processes can be very useful. With the the R package **tvgarch**, a nonstationary GARCH process is simulated with the `tvgarchSim()` function for univariate and the `mtvgarchSim()` function for an m -dimensional series, respectively. We also show how useful **tvgarch** is for modeling nonstationary volatility in financial empirical applications. Using daily stock returns, the results from the tests of constant unconditional variance using the `tvgarchTest()` function provide strong evidence for nonstationary volatility. To estimate the multivariate TV-GARCH model, we use the `mtvgarch()` function. In the first step, the conditional variances are modeled equation by equation with the univariate `tvgarch()` function and then, in the second step, the conditional correlations are estimated conditionally on the estimates obtained in the first step. In practice, the univariate `tvgarch()` function estimates a TV-GARCH model with a single transition function in the TV component and invokes the `garchx()` function from package **garchx** to estimate the parameters in the GARCH component. By assuming a single transition function, financial volatility appears to increase during the 2008 global financial crisis, making its structure to change either temporarily (two transitions) or permanently (one transition) for the sample period analyzed. Several extraction functions allow us to analyze the results after estimating a TV-GARCH model, including to obtain the necessary data to compare different model specifications.

References

- Amado C, Silvennoinen A, Teräsvirta T (2019). “Models with Multiplicative Decomposition of Conditional Variances and Correlations.” In J Chevallier, S Goutte, D Guerreiro, S Saglio, B Sanhaji (eds.), *Financial Mathematics, Volatility and Covariance Modelling*, volume 2, pp. 217–260. Routledge, Milton.

- Amado C, Teräsvirta T (2013). “Modelling Volatility by Variance Decomposition.” *Journal of Econometrics*, **175**, 142–153. doi:10.1016/j.jeconom.2013.03.006.
- Amado C, Teräsvirta T (2014). “Modelling Changes in the Unconditional Variance of Long Stock Return Series.” *Journal of Empirical Finance*, **25**, 15–35. doi:10.1016/j.jempfin.2013.09.003.
- Amado C, Teräsvirta T (2017). “Specification and Testing of Multiplicative Time-Varying GARCH Models with Applications.” *Econometric Reviews*, **36**, 421–446. doi:10.1080/07474938.2014.977064.
- Andersen TG, Bollerslev T (1998). “Answering the Skeptics: Yes, Standard Volatility Models Do Provide Accurate Forecasts.” *International Economic Review*, **39**, 885–905. doi:10.2307/2527343.
- Andreou E, Ghysels E (2008). “Quality Control for Structural Credit Risk Models.” *Journal of Econometrics*, **146**, 364–375. doi:10.1016/j.jeconom.2008.08.013.
- Ardia D, Bluteau K, Boudt K, Catania L, Trottier D (2019). “Markov-Switching GARCH Models in R: The **MSGARCH** Package.” *Journal of Statistical Software*, **91**(4), 1–38. doi:10.18637/jss.v091.i04.
- Baillie RT, Bollerslev T, Mikkelsen HO (1996). “Fractionally Integrated Generalized Autoregressive Conditional Heteroskedasticity.” *Journal of Econometrics*, **74**, 3–30. doi:10.1016/s0304-4076(95)01749-6.
- Barr C (2017). “Aug. 9, 2007: The Day the Mortgage Crisis Went Global.” *The Wall Street Journal*. August 9, 2017, URL <https://www.wsj.com/articles/aug-9-2007-the-day-the-mortgage-crisis-went-global-1502271004>.
- Bauwens L, Laurent S, Rombouts JVK (2006). “Multivariate GARCH Models: A Survey.” *Journal of Applied Econometrics*, **21**, 79–109. doi:10.1002/jae.842.
- Bollerslev T (1986). “Generalized Autoregressive Conditional Heteroskedasticity.” *Journal of Econometrics*, **31**, 307–327. doi:10.1016/0304-4076(86)90063-1.
- Bollerslev T (1990). “Modelling the Coherence in Short-Run Nominal Exchange Rates: A Multivariate Generalized ARCH Model.” *Review of Economics and Statistics*, **72**, 498–505. doi:10.2307/2109358.
- Brownlees CT, Cipollini F, Gallo GM (2012). “Multiplicative Error Models.” In L Bauwens, C Hafner, S Laurent (eds.), *Handbook of Volatility Models and Their Applications*, pp. 223–247. John Wiley & Sons, New Jersey.
- Campos-Martins S, Sucarrat G (2023). **tvgarch**: *Time Varying GARCH Modelling*. R package version 2.4.2, URL <https://CRAN.R-project.org/package=tvgarch>.
- Candila V (2024). **dccmidas**: *A Package for Estimating DCC-Based Models in R*. R package version 0.1.2, URL <https://CRAN.R-project.org/package=dccmidas>.
- Cho H, Korkas K (2019). **segMGarch**: *Multiple Change-Point Detection for High-Dimensional GARCH Processes*. R package version 1.2, URL <https://CRAN.R-project.org/package=segMGarch>.

- Doornik JA (2018). *An Introduction to OxMetrics 8*. London. URL <https://www.doornik.com/>.
- Engle RF (1982). “Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation.” *Econometrica*, **50**, 987–1007. doi:10.2307/1912773.
- Engle RF (2002). “Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroskedasticity Models.” *Journal of Business & Economic Statistics*, **20**, 339–350. doi:10.1198/073500102288618487.
- Engle RF, Bollerslev T (1986). “Modelling the Persistence of Conditional Variances.” *Econometric Reviews*, **5**, 1–50. doi:10.1080/07474938608800095.
- Engle RF, Rangel JG (2008). “The Spline GARCH Model for Low Frequency Volatility and Its Global Macroeconomic Causes.” *Review of Financial Studies*, **21**, 1187–1222. doi:10.1093/rfs/hhn004.
- Engle RF, Russell JR (1998). “Autoregressive Conditional Duration: A New Model of Irregularly Spaced Transaction Data.” *Econometrica*, **66**, 1127–1162. doi:10.2307/2999632.
- Francq C, Zakoïan JM (2012). “QML Estimation of a Class of Multivariate Asymmetric GARCH Models.” *Econometric Theory*, **28**, 179–206. doi:10.1017/s0266466611000156.
- Francq C, Zakoïan JM (2016). “Estimating Multivariate Volatility Models Equation by Equation.” *Journal of the Royal Statistical Society B*, **78**, 613–635. doi:10.1111/rssb.12126.
- Francq C, Zakoïan JM (2019). *GARCH Models*. 2nd edition. John Wiley & Sons, New York.
- Galanos A (2022). *rmgarch: Multivariate GARCH Models*. R package version 1.3-9, URL <https://CRAN.R-project.org/package=rmgarch>.
- Galanos A (2023). *rugarch: Univariate GARCH Models*. R package version 1.5-1, URL <https://CRAN.R-project.org/package=rugarch>.
- Gilbert P, Varadhan R (2019). *numDeriv: Accurate Numerical Derivatives*. R package version 2016.8-1.1, URL <https://CRAN.R-project.org/package=numDeriv>.
- Glosten LR, Jagannathan R, Runkle DE (1993). “On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks.” *Journal of Finance*, **48**, 1779–1801. doi:10.21034/sr.157.
- Hamilton JD, Susmel R (1994). “Autoregressive Conditional Heteroskedasticity and Changes in Regime.” *Journal of Econometrics*, **64**, 307–333. doi:10.1016/0304-4076(94)90067-1.
- Hansen PR, Lunde A (2005). “A Forecast Comparison of Volatility Models: Does Anything Beat a GARCH(1,1)?” *Journal of Applied Econometrics*, **20**, 873–889. doi:10.1002/jae.800.
- Hillebrand E (2005). “Neglecting Parameter Changes in GARCH Models.” *Journal of Econometrics*, **129**, 121–138. doi:10.1016/j.jeconom.2004.09.005.

- Hosszejni D, Kastner G (2021). “Modeling Univariate and Multivariate Stochastic Volatility in R with **stochvol** and **factorstochvol**.” *Journal of Statistical Software*, **100**(12), 1–34. doi:10.18637/jss.v100.i12.
- Lamoureux CG, Lastrapes WD (1990). “Persistence in Variance, Structural Change, and the GARCH Model.” *Journal of Business & Economic Statistics*, **8**, 225–234. doi:10.1080/07350015.1990.10509794.
- Laurent S (2018). *Estimating and Forecasting ARCH Models Using G@RCH 8*. London. **OxMetrics** version 8.
- Luukkonen R, Saikkonen P, Teräsvirta T (1988). “Testing Linearity against Smooth Transition Autoregressive Models.” *Biometrika*, **75**, 491–499. doi:10.1093/biomet/75.3.491.
- Mikosch T, Stărică C (2004). “Non-Stationarities in Financial Time Series, the Long-Range Dependence and the IGARCH Effects.” *Review of Economics and Statistics*, **86**, 378–384. doi:10.1162/003465304323023886.
- Pastor L, Stambaugh RF (2001). “The Equity Premium and Structural Breaks.” *The Journal of Finance*, **56**, 1207–1239. doi:10.1111/0022-1082.00365.
- Pettenuzzo D, Timmermann A (2011). “Predictability of Stock Returns and Asset Allocation under Structural Breaks.” *Journal of Econometrics*, **164**, 60–78. doi:10.1016/j.jeconom.2011.02.019.
- R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Song P, Fan Y, Kalbfleisch JD (2005). “Maximization by Parts in Likelihood Inference.” *Journal of the American Statistical Association*, **100**, 1145–1158. doi:10.1198/016214505000000204.
- Sucarrat G (2021). “**garchx**: Flexible and Robust GARCH-X Modelling.” *The R Journal*, **13**(1), 276–291. doi:10.32614/rj-2021-057.
- Wooldridge JM (1990). “A Unified Approach to Robust, Regression-Based Specification Tests.” *Econometric Theory*, **6**, 17–43. doi:10.1017/s0266466600004898.
- Wooldridge JM (1991). “On the Application of Robust, Regression-Based Diagnostics to Models of Conditional Means and Conditional Variances.” *Journal of Econometrics*, **47**, 5–46. doi:10.1016/0304-4076(91)90076-p.
- Wuertz D, Chalabi Y, Setz T, Maechler M, Boshnakov GN (2024). **fGarch**: *Rmetrics – Autoregressive Conditional Heteroskedastic Modelling*. R package version 4033.92, URL <https://CRAN.R-project.org/package=fGarch>.
- Zeileis A, Grothendieck G (2005). “**zoo**: S3 Infrastructure for Regular and Irregular Time Series.” *Journal of Statistical Software*, **14**(6), 1–27. doi:10.18637/jss.v014.i06.

A. Estimating the nonstationary conditional variance

Before describing the estimation strategy, it is convenient to introduce some notation. The full set of variance parameters is denoted by $\boldsymbol{\vartheta} = (\boldsymbol{\vartheta}^h, \boldsymbol{\vartheta}^g)'$, where $\boldsymbol{\vartheta}^h$ denotes the set of parameters in the GARCH-X component, h_t , and $\boldsymbol{\vartheta}^g$ denotes the set of parameters in TV component, g_t .

The `tvgarch()` function can be used to estimate the parameters in $\boldsymbol{\vartheta}$ by quasi maximum likelihood. Assume the conditional variance

$$\sigma_t^2(\boldsymbol{\vartheta}) = h_t(\boldsymbol{\vartheta}^h, \boldsymbol{\vartheta}^g)g_t(\boldsymbol{\vartheta}^g).$$

Then, $\hat{\boldsymbol{\vartheta}}$, the estimator of $\boldsymbol{\vartheta}$, can be obtained as the solution of

$$\arg \max L_n(\boldsymbol{\vartheta}) = \sum_{t=1}^n \ell_t(\boldsymbol{\vartheta}),$$

where

$$\ell_t(\boldsymbol{\vartheta}) = \kappa - \frac{1}{2} \left\{ \ln [h_t(\boldsymbol{\vartheta}^h, \boldsymbol{\vartheta}^g)] + \ln [g_t(\boldsymbol{\vartheta}^g)] + \frac{u_t^2}{h_t(\boldsymbol{\vartheta}^h, \boldsymbol{\vartheta}^g)g_t(\boldsymbol{\vartheta}^g)} \right\}$$

is the log-likelihood function for u_t assuming normality. Notice, however, that the estimation is robust to a misspecified density.

To estimate the parameters in the nonstationary conditional variance, whose components are defined in Equations 6–7 and Equation 8, maximization by parts is applied with the `tvgarch()` function. In practice, $\hat{\boldsymbol{\vartheta}}^h$ and $\hat{\boldsymbol{\vartheta}}^g$ are obtained iteratively until convergence. Maximization by parts for the estimation of TV-GARCH models is described in Appendix B. The parameters in the GARCH-X component of either a stationary or nonstationary GARCH model are estimated by the `garchx()` function of the R package `garchx` (Sucarrat 2021), on which the package `tvgarch` depends; see also Sucarrat (2021). For inference, we rely on the asymptotic distribution of the estimator $\hat{\boldsymbol{\vartheta}}$ being normal. As an illustration, in Section 5.2, the `tvgarch()` function is applied to real data, namely a time series of returns on the NASDAQ index.

B. Maximization by parts

To estimate the nonstationary conditional variance of a financial time series, it is convenient to estimate the parameters in each component separately; see Amado and Teräsvirta (2013) for further details on the estimation. The vectors $\boldsymbol{\vartheta}^h$ and $\boldsymbol{\vartheta}^g$ include the parameters in the GARCH-X and TV components, respectively. For the most general variance model, these correspond to Equation 8 and Equations 6–7, respectively. We will use the notation (\cdot) to enumerate the iteration. The maximization by parts is then done iteratively until convergence as follows:

1. By setting $h_t^{(0)} = 1$, $\hat{\boldsymbol{\vartheta}}_{(0)}^g$ is obtained as the solution of

$$\arg \max L_n(\boldsymbol{\vartheta}_{(0)}^g) = \sum_{t=1}^n \ell_t(\boldsymbol{\vartheta}_{(0)}^g),$$

where

$$\ell_t(\boldsymbol{\vartheta}_{(0)}^g) = \kappa - \frac{1}{2} \left\{ \ln [g_t(\boldsymbol{\vartheta}_{(0)}^g)] + \frac{u_t^2}{g_t(\boldsymbol{\vartheta}_{(0)}^g)} \right\}.$$

$\hat{g}_t^{(0)} \equiv \hat{g}_t(\hat{\boldsymbol{\vartheta}}_{(0)}^g)$ and $\hat{\phi}_t^{(0)} = \frac{u_t}{\sqrt{\hat{g}_t^{(0)}}}$ are both computed. The parameter δ_0 is fixed to

$\hat{\delta}_0^{(0)}$ from this iteration onwards. This parameter is fixed for identifiability purposes, see [Amado and Teräsvirta \(2013\)](#), and [Amado and Teräsvirta \(2017, p. 423\)](#).

2. $\hat{\boldsymbol{\vartheta}}_{(1)}^h$ is obtained as the solution of

$$\arg \max L_n(\boldsymbol{\vartheta}_{(1)}^h, \hat{\boldsymbol{\vartheta}}_{(0)}^g) = \sum_{t=1}^n \ell_t(\boldsymbol{\vartheta}_{(1)}^h, \hat{\boldsymbol{\vartheta}}_{(0)}^g),$$

where

$$\ell_t(\boldsymbol{\vartheta}_{(1)}^h, \hat{\boldsymbol{\vartheta}}_{(0)}^g) = \kappa - \frac{1}{2} \left\{ \ln [h_t(\boldsymbol{\vartheta}_{(1)}^h, \hat{\boldsymbol{\vartheta}}_{(0)}^g)] + \frac{[\hat{\phi}_t^{(0)}]^2}{h_t(\boldsymbol{\vartheta}_{(1)}^h, \hat{\boldsymbol{\vartheta}}_{(0)}^g)} \right\}.$$

$\hat{h}_t^{(1)} \equiv h_t(\hat{\boldsymbol{\vartheta}}_{(1)}^h, \hat{\boldsymbol{\vartheta}}_{(0)}^g)$ and $\hat{\psi}_t^{(1)} = \frac{u_t}{\sqrt{\hat{h}_t^{(1)}}}$ are both computed.

3. $\boldsymbol{\vartheta}^g$ is re-estimated to obtain $\hat{\boldsymbol{\vartheta}}_{(1)}^g$ as the solution of

$$\arg \max L_n(\hat{\boldsymbol{\vartheta}}_{(1)}^h, \boldsymbol{\vartheta}_{(1)}^g) = \sum_{t=1}^n \ell_t(\hat{\boldsymbol{\vartheta}}_{(1)}^h, \boldsymbol{\vartheta}_{(1)}^g),$$

where

$$\ell_t(\hat{\boldsymbol{\vartheta}}_{(1)}^h, \boldsymbol{\vartheta}_{(1)}^g) = \kappa - \frac{1}{2} \left\{ \ln [g_t(\boldsymbol{\vartheta}_{(1)}^g)] + \frac{[\hat{\phi}_t^{(1)}]^2}{g_t(\boldsymbol{\vartheta}_{(1)}^g)} \right\}.$$

$\hat{g}_t^{(1)} \equiv g_t(\hat{\boldsymbol{\vartheta}}_{(1)}^g)$ and $\hat{\phi}_t^{(1)} = \frac{u_t}{\sqrt{\hat{g}_t^{(1)}}}$ are both updated.

4. Steps 2 and 3 are iterated until convergence. In each iteration $j > 1$, $\hat{\boldsymbol{\vartheta}}_{(j)}^h$ and $\hat{\boldsymbol{\vartheta}}_{(j)}^g$ are obtained, and $\hat{h}_t^{(j)} \equiv h_t(\hat{\boldsymbol{\vartheta}}_{(j)}^h, \hat{\boldsymbol{\vartheta}}_{(j-1)}^g)$ and $\hat{g}_t^{(j)} \equiv g_t(\hat{\boldsymbol{\vartheta}}_{(j)}^g)$ are updated.

C. Estimating the conditional correlations

In order to estimate the conditional correlation in multivariate financial time series, the estimation of the conditional variance is repeated for each time series of residuals, $u_{it}, i =$

\dots, m , and the vector of volatility standardized residuals $\hat{\boldsymbol{\eta}}_t = (\hat{\eta}_{1t}, \dots, \hat{\eta}_{mt})'$ is constructed, where

$$\hat{\eta}_{it} = \frac{u_{it}}{\sqrt{\hat{h}_{it}(\hat{\boldsymbol{\vartheta}}_i^h, \hat{\boldsymbol{\vartheta}}_i^g) \hat{g}_{it}(\hat{\boldsymbol{\vartheta}}_i^g)}}.$$

The set of all parameters is denoted by $\boldsymbol{\theta} = (\boldsymbol{\vartheta}'_1, \dots, \boldsymbol{\vartheta}'_m, \boldsymbol{\rho}')'$, where $\boldsymbol{\rho}$ contains the correlation parameters. The composition of the parameter vector $\boldsymbol{\rho}$ depends on the structure of the conditional correlation matrix \mathbf{P}_t . When the conditional correlations are time-invariant, i.e., $\mathbf{P}_t \equiv \mathbf{P}$, $\boldsymbol{\rho}$ includes all the unique pairwise correlations of the m -dimensional time series. When conditional correlations follow the dynamic structure of Equations 10–11, $\boldsymbol{\rho}$ includes all the unique pairwise correlations of the unconditional correlation matrix \mathbf{Q}_t and the two additional parameters governing their dynamics, namely θ_1 and θ_2 . In practice, only θ_1 and θ_2 are estimated and \mathbf{Q}_t is simply replaced by the sample correlation matrix.

The `mtvgarch()` function estimates, equation by equation, the vector of conditional variances σ_{it}^2 by calling the univariate `tvrgarch()` function for each constituent time series. Then, $\hat{\boldsymbol{\rho}}$ can be obtained by quasi maximum likelihood as the solution of

$$\arg \max L_n(\boldsymbol{\rho} | \hat{\boldsymbol{\vartheta}}_1, \dots, \hat{\boldsymbol{\vartheta}}_m) = -\frac{1}{2} \sum_{t=1}^n \left(\ln |\mathbf{P}_t| + \hat{\boldsymbol{\eta}}_t' \mathbf{P}_t^{-1} \hat{\boldsymbol{\eta}}_t \right).$$

D. Testing the constancy of GARCH parameters

Before estimating a TV-GARCH model, it is important to check the hypothesis of constant GARCH parameters, i.e., that the unconditional variance is time-invariant. Rejecting this hypothesis thus provides evidence for nonstationarity meaning that a standard GARCH model with constant parameters is misspecified and so not appropriate to fit the data. The structure of g_t , i.e., the number of transitions, has to be determined and the resulting TV-GARCH model can then be estimated. A brief description of the statistical test is provided below. For further details, including finite sample simulation, we refer to [Amado and Teräsvirta \(2014, 2017\)](#).

Consider a slightly different TV(1)-GARCH(1, 1), whose GARCH component is defined by Equation 4 and the TV component has the form

$$g_t = 1 + \delta_1 G_1(\gamma_1, \mathbf{c}_1; s_t),$$

where the intercept, δ_0 , is replaced by 1. The subsets of variance parameters are defined as follows: $\boldsymbol{\vartheta}_i^h = (\omega_i, \alpha_{i1}, \beta_{i1})'$ and $\boldsymbol{\vartheta}_i^g = (\delta_{i1}, \gamma_{i1}, \mathbf{c}'_{i1})'$.

When $g_t \equiv 1$, $\phi_t = u_t$ and the model collapses into the standard GARCH(1, 1) model in which case the unconditional variance is time invariant and the conditional variance stationary. To check the constancy of the TV component, we can test either $\mathbb{H}_0 : g_t \equiv 1$ or $\mathbb{H}_0 : \gamma_1 = 0$ against $\mathbb{H}_1 : g_t = 1 + \delta_1 G_1(\gamma_1, \mathbf{c}_1; s_t)$. A fairly flexible form is assumed for $G_1(\gamma_1, \mathbf{c}_1; s_t)$ by setting $k_1 = 3$, i.e., $\mathbf{c}_1 = (c_{11}, c_{12}, c_{13})'$. Due to the inability for identifying δ_1 and \mathbf{c}_1 under \mathbb{H}_0 , the transition function is replaced by its first-order Taylor approximation; see [Luukkonen, Saikkonen, and Teräsvirta \(1988\)](#). The resulting linearized and re-parametrized deterministic component is given by

$$g_t^* = \delta_0^* + s_t \delta_1^* + s_t^2 \delta_2^* + s_t^3 \delta_3^* \quad (12)$$

where δ_0^* , δ_1^* , δ_2^* and δ_3^* are functions of the original parameters.

When \mathbb{H}_0 holds, $\delta_1^* = \delta_2^* = \delta_3^* = 0$ and the parameters in $\boldsymbol{\vartheta}^h$ remain constant. \mathbb{H}_0 is thus replaced by the auxiliary null hypothesis $\mathbb{H}_0^{aux} : \delta_1^* = \delta_2^* = \delta_3^* = 0$. If \mathbb{H}_0^{aux} is rejected, proceed to testing the following sequence of nested hypotheses in order to specify the correct number of locations in the transition function:

$$\begin{aligned} \mathbb{H}_{03} : \delta_3^* &= 0 & (13) \\ \mathbb{H}_{02} : \delta_2^* &= 0 \quad | \quad \delta_3^* = 0 \\ \mathbb{H}_{01} : \delta_1^* &= 0 \quad | \quad \delta_3^* = \delta_2^* = 0. \end{aligned}$$

These tests are easy to implement as they can be based on auxiliary regressions. In practice, the test statistics are computed as nR^2 , where n is the number of observations (the usual notation is TR^2 , where T is the number of observations) and R^2 is the coefficient of determination of an auxiliary regression; see Wooldridge (1990, 1991). Under the null hypothesis, the test statistics have an asymptotic χ_v^2 distribution, where v is the number of degrees of freedom, i.e., the number of restrictions under the null hypothesis. In the general misspecification test, $v = 3$ and in each of the sequential tests in Equation 13, $v = 1$. The test has good size and power in finite samples as shown in Amado and Teräsvirta (2017).

Testing for the necessity of an additional transition function is also possible. In particular, we could test the null hypothesis $\mathbb{H}_0^2 : g_t = \delta_0 + \delta_1 G_1(\gamma_1, \mathbf{c}_1; s_t)$ against the alternative $\mathbb{H}_1^2 : g_t = \delta_0 + \sum_{l=1}^2 \delta_{1l} G_l(\gamma_l, \mathbf{c}_l; s_t)$. We refer to Amado and Teräsvirta (2017) for further details on this type of misspecification of the TV-GARCH model. This test is not yet available in *tvgarch*. In the absence of a statistical test, information criteria from models with different TV component specifications combined with visual inspection of the time series are useful tools to select the best model.

Affiliation:

Susana Campos-Martins
Catholic University of Portugal
Palma de Cima, 1649-023 Lisbon, Portugal
E-mail: scmartins@ucp.pt
URL: <https://sites.google.com/site/susanacamposmartins>

Genaro Sucarrat
Department of Economics, BI Norwegian Business School
Nydalsveien 37, 0484 Oslo, Norway
E-mail: genaro.sucarrat@bi.no
URL: <https://www.sucarrat.net/>

Journal of Statistical Software
published by the Foundation for Open Access Statistics
March 2024, Volume 108, Issue 9
[doi:10.18637/jss.v108.i09](https://doi.org/10.18637/jss.v108.i09)

<https://www.jstatsoft.org/>
<https://www.foastat.org/>
Submitted: 2021-12-23
Accepted: 2023-07-06
