

GARCH Models

Alexios Galanos

2026-10-09

Contents

1	Introduction	1
2	GARCH Estimation	2
2.1	ARMA Mean Equation	3
2.2	Recursion Initialization (σ_0^δ)	6
2.3	Model Persistence, Long Run Variance and Half-Life	6
2.4	Variance Targeting ($\bar{\omega}$)	6
2.5	External Regressors (χ)	7
2.6	News Impact Curve (NIC)	7
2.7	Standard Errors	7
2.7.1	Direct (H)	8
2.7.2	Outer product of the gradient (OP)	8
2.7.3	Quasi-Maximum Likelihood (QML)	8
2.7.4	HAC (NW)	8
2.8	Parameter Scaling	9
2.9	GARCH Flavors	9
2.9.1	Vanilla GARCH (<i>garch</i>)	9
2.9.2	Integrated GARCH (<i>igarch</i>)	11
2.9.3	Exponentially Weighted Moving Average (<i>ewma</i>)	11
2.9.4	Exponential GARCH (<i>egarch</i>)	12
2.9.5	GJR GARCH (<i>gjrgarch</i>)	13
2.9.6	Asymmetric Power ARCH (<i>aparch</i>)	15
2.9.7	Family GARCH (<i>fgarch</i>)	16
2.9.8	Component GARCH (<i>cgarch</i>)	18
3	Package Methods	19
4	Conclusion	20
5	References	20

1 Introduction

The Generalized Auto-Regressive Conditional Heteroscedasticity (**GARCH**) model of [Bollerslev \(1986\)](#) and the numerous extensions which have followed since, is a framework for modeling the dynamics of the conditional variance. It has proved particularly popular, particularly among financial market practitioners, but also in other areas where the arrival of unexpected new information may lead to non-instantaneous decay (*persistence*) and/or asymmetric reaction to good and bad news (*news impact*).

Many programming languages have one or more implementations of GARCH, with R having no less than 3, including the `garch` function from the `tseries` package, `fGarch` and `rugarch`. A select R package review

is provided in [Hill and McCullough \(2019\)](#), and a cross language package review on asymmetric GARCH in [Charles and Darné \(2019\)](#). The suggestions and constructive feedback in these papers were taken into account when developing this new package, as were many of the user comments over the last 10+ years.

The **tsgarch** package is a partial re-implementation of [rugarch](#), by the same author, with key differences summarized below:

- does not (yet) implement all GARCH models in [rugarch](#). FIGARCH, Multiplicative Component GARCH and Realized GARCH are not currently implemented.
- does not implement joint ARFIMA-GARCH estimation, but does allow for a jointly estimated $\text{ARMA}(p, q)$ -X mean equation (see Section 2.1) in addition to the constant. With so many options for modelling the conditional mean beyond low order ARMA dynamics, many of which are available in the **tsmodels** framework, it was decided to keep this package’s mean equation support deliberately limited rather than pursue full joint estimation of arbitrary mean dynamics with the variance equation. For a fully general mean equation (e.g. ARFIMA-X, or higher order fractional dynamics), the 2 step estimation approach, whereby the residuals of the conditional mean are passed to the variance dynamics estimation, remains available and may be more flexible in what can be achieved, even if potentially less efficient for small sized datasets. Additionally, the ARCH-in-mean model is no longer available, as it was found to have very limited value within the **tsmodels** framework or in general, this author’s experience. The sibling [tsarma](#) package remains available for standalone $\text{ARMA}(p, q)$ -X modelling outside the GARCH setting.
- makes use of automatic differentiation (autodiff) during estimation, via the [TMB](#) package. This is in line with similar approaches in other models written in the **tsmodels** framework. Using autodiff allows for more confident estimation and more accurate standard errors. Autodiff is also used for the more complex inequality constraints of some models which involve evaluation of an expectation, in order to generate more accurate Jacobians of these inequalities for use during optimization.
- adopts a custom approach in optimization using parameter scaling which avoids problems with convergence and numerical stability.
- fully implements and correctly documents a number of sandwich estimators making use of the [sandwich](#) package framework (with methods for **bread** and **estfun** and **meat/meat_HAC** functions).
- makes use of S3 methods and classes, abandoning the S4 approach of **rugarch**. Additionally, while making use of standard methods from the stats package, some of the methods are based on those exported from [tsmethods](#), consistent with other packages in the **tsmodels** framework. The Appendix provides a table with a hierarchical overview of the main functions and methods currently implemented in the package.
- provides more accurate and informative output, including standard errors, confidence intervals, and a more detailed summary output including a fancy [flextable](#) output for the **summary** method.
- provides numerous fixes and enhancements to the original **rugarch** code, including the unconditional variance of the exponential GARCH(1,1) model, and better startup conditions for the initialization of the recursion.
- provides more extensive documentation and high coverage of unit tests, with a focus on the accuracy of the estimation and the correctness of the output as well as catching many corner cases.

This vignette provides a mathematical summary of each type of GARCH model used, general assumptions regarding initialization and other estimation details as well as the forecast equation. Separate vignettes are available with code demonstrations.

2 GARCH Estimation

The density function, based on the distributions available in the [tsdistributions](#) package, is expressed in terms of the location, scale, skew and shape parameters $\{\mu_t, \sigma_t, \zeta, \nu\}$, normalized to give zero mean and unit

variance:

$$z_t = \frac{y_t - \mu_t}{\sigma_t} \quad (1)$$

where $\mu_t = E(y_t|x_t)$ and $\sigma_t^2 = E((y_t - \mu_t)^2|x_t)$, with x_t the information set available at the period immediately before time t and may include both external regressors and lagged values of y . Assuming that the distribution of z_t is independent of the conditioning information set¹, then:

$$g(z_t|\zeta, \nu) = \frac{d}{dz} P(z_t < z|\zeta, \nu) \quad (2)$$

which is related to the distribution of y_t by through the scale term:

$$f(y_t|\mu_t, \sigma_t, \zeta, \nu) = \frac{1}{\sigma_t} g(z_t|\zeta, \nu) \quad (3)$$

The parameter set θ , which includes the conditional mean, conditional variance and distributional skew and shape parameters is estimated by minimizing the negative of the log-likelihood:

$$\theta_{ML} = \arg \min_{\theta} \sum_{t=1}^T -\log \left(\frac{1}{\sigma_t} g(z_t|\zeta, \nu) \right) \quad (4)$$

subject to upper and lower parameters bound constraints as well as the persistence constraint ($P < 1$) and other non-negativity constraints. Estimation is performed using the `nloptr` solver with analytic Gradient and Jacobian of the constraints² provided through autodiff in **TMB**.

2.1 ARMA Mean Equation

By default μ_t reduces to either zero or a constant μ (controlled by the `constant` argument to `garch_modelspec`), in which case $y_t - \mu_t = y_t - \mu$ is simply passed to the conditional variance equation. Setting the `arma = c(p,q)` argument instead specifies a jointly estimated ARMA(p,q) mean equation:

$$y_t - \mu = \sum_{i=1}^p \phi_i (y_{t-i} - \mu) + \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j} \quad (5)$$

so that μ retains its interpretation as the unconditional mean of y , and ε_t (rather than $y_t - \mu$) is passed on to the conditional variance equation of whichever GARCH flavor was chosen. Setting `arma = c(0,0)` (the default) reduces Equation 5 exactly to the pre-existing constant-mean behavior. The `constant` argument remains independent of `arma`: it controls only whether μ is estimated or fixed at zero.

Regressors $x_{k,t}$ ($k = 1, \dots, s$) can additionally enter the conditional mean, via the `xreg` argument to `garch_modelspec`, with coefficients τ_k (named `tau1`, `tau2`, ... in the `parmatrix`). The `xreg_type` argument selects between two conventions. The default, `xreg_type = "arma_errors"`, applies the ARMA recursion to the regressor-adjusted deviations:

$$y_t - \mu - \sum_{k=1}^s \tau_k x_{k,t} = \sum_{i=1}^p \phi_i \left(y_{t-i} - \mu - \sum_{k=1}^s \tau_k x_{k,t-i} \right) + \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j} \quad (6)$$

¹In the case when this assumption is relaxed, this will give rise to a non constant conditional distribution. See Hansen (1994).

²For some of the problems, the persistence has an easy closed form solution and therefore the Jacobian is hardcoded instead of making use of autodiff.

This is the convention of `stats::arima(xreg = )` and of the sibling **tsarma** package: the regressors are filtered by the AR polynomial, τ_k is the long-run marginal effect of $x_{k,t}$ on y_t , and $E[y_t] = \mu + \sum_k \tau_k E[x_{k,t}]$, so μ retains its interpretation as the unconditional mean of y net of the regressor contribution. The alternative, `xreg_type = "armax"`, is the **rugarch** convention:

$$y_t = \mu + \sum_{i=1}^p \phi_i (y_{t-i} - \mu) + \sum_{k=1}^s \tau_k x_{k,t} + \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j} \quad (7)$$

Here the regressors are not filtered by the AR polynomial, so τ_k is the impact (contemporaneous) effect and the long-run effect of a sustained change in $x_{k,t}$ is $\tau_k / (1 - \sum_i \phi_i)$. The two conventions differ only by the lagged-regressor term $\sum_k \tau_k x_{k,t-i}$ inside the AR feedback, and are therefore algebraically identical whenever $p = 0$; the choice of `xreg_type` only matters when the mean equation has AR dynamics. Note that **tsarma** writes these coefficients ξ_k ; that symbol is already used in this package for the variance equation regressors (see the “External Regressors” equations of each flavor below), hence τ_k here. In both conventions it is again ε_t - not $y_t - \mu$ - that is passed on to the conditional variance equation, exactly as in the pure ARMA case.

Because the recursion consumes $x_{k,t}$ at every time step, forecasting and simulation require future regressor values (`newxreg` in `predict()`, `xreg` in `simulate()`), and `tsfilter()` likewise requires `newxreg` covering the appended observations. When the model includes `xreg` but the future values are not supplied, a zero matrix is substituted with a warning.

2.1.0.1 Choosing between the two conventions Expanding both conventions for an AR(1) mean equation with a single regressor makes the difference precise. Writing the `arma_errors` form out in levels:

$$y_t = \mu(1 - \phi_1) + \phi_1 y_{t-1} + \tau_1 x_{1,t} - \phi_1 \tau_1 x_{1,t-1} + \varepsilon_t \quad (8)$$

and the `armax` form:

$$y_t = \mu(1 - \phi_1) + \phi_1 y_{t-1} + \tau_1 x_{1,t} + \varepsilon_t \quad (9)$$

Both are restrictions of the same autoregressive distributed lag model $y_t = c + \phi_1 y_{t-1} + \beta_0 x_{1,t} + \beta_1 x_{1,t-1} + \varepsilon_t$: `arma_errors` imposes the common factor restriction $\beta_1 = -\phi_1 \beta_0$, whereas `armax` imposes $\beta_1 = 0$. Neither is nested in the other, so the choice is a genuine modelling decision rather than a reparameterization.

The practical consequence is clearest in how a shock to the regressor propagates. Under `arma_errors`, since $y_t = \mu + \sum_k \tau_k x_{k,t} + w_t$ with w_t an ARMA process which does not involve x , a one period unit blip in $x_{k,t}$ moves y_t by τ_k and leaves every subsequent observation untouched: the regressor effect is static and the ARMA dynamics belong entirely to the error. Under `armax` the regressor enters the same equation as the autoregressive feedback, so the same blip moves y_t by τ_k , y_{t+1} by $\phi_1 \tau_k$, y_{t+2} by $\phi_1^2 \tau_k$ and so on, cumulating to $\tau_k / (1 - \sum_i \phi_i)$. For a sustained unit increase in the regressor, the long run shift in y is τ_k under `arma_errors` and $\tau_k / (1 - \sum_i \phi_i)$ under `armax`.

This suggests, rather than dictates, where each is appropriate. An event or calendar dummy usually has no reason to echo into subsequent periods at the same rate as the persistence of the series itself, which is what `armax` imposes, so the static reading of τ_k as the entire effect is the easier one to defend. A covariate whose influence genuinely accumulates and decays with the series, such as a policy rate or a sentiment index, fits the `armax` propagation more naturally. When the AR order is zero the two coincide exactly, and when the regressor is close to serially uncorrelated the distinction is weak; it is with a persistent regressor and a persistent mean equation that the two part company.

Choosing wrongly is costly, and the cost does not fall where intuition might suggest. In a Monte Carlo experiment with 250 replications per cell ($n = 1500$, $\mu = 0.5$, $\phi_1 = 0.6$, $\tau_1 = 1.5$, Gaussian errors of constant variance), fitting both conventions to data generated by each, for an independent and for an AR(0.9) regressor, the average biases are:

generated by	regressor	bias in τ_1	bias in ϕ_1
arma_errors	independent	-0.003	-0.355
arma_errors	AR(0.9)	-0.329	-0.355
armax	independent	-0.420	+0.088
armax	AR(0.9)	-0.074	+0.356

each row reporting the fit under the *other* convention; the correctly specified fit is unbiased to within 0.005 in every cell.

Two things follow. Whichever convention is the wrong one, the autoregressive coefficient is the parameter most reliably damaged: the bias in ϕ_1 is roughly ± 0.36 on a true value of 0.6 in three of the four misspecified cells, which is easily enough to change the implied half life and the rate at which mean forecasts revert by a factor of several. And how the damage divides between τ_1 and ϕ_1 turns on the persistence of the regressor rather than on the direction of the error: with an independent regressor and **armax** data, the **arma_errors** fit misses τ_1 by -0.42 while recovering ϕ_1 tolerably, whereas with a persistent regressor the same misspecification leaves τ_1 nearly right and puts the error into ϕ_1 instead. Inspecting only the regressor coefficient can therefore be reassuring and wrong at the same time.

The data are, however, decisive about which convention applies. Across all 1000 comparisons in that experiment the correctly specified convention had the higher likelihood every single time, with mean gaps of 190 to 632 log points, so the choice need not rest on judgement alone: estimate both and compare.

The restriction is also testable. Since **armax** with both $x_{k,t}$ and $x_{k,t-1}$ supplied as separate columns of **xreg** spans the unrestricted distributed lag model, estimating that model and comparing $\hat{\beta}_1$ against $-\hat{\phi}_1\hat{\beta}_0$ and against zero indicates which of the two restrictions the data prefer.

Stationarity of the AR polynomial and invertibility of the MA polynomial are required for ϕ and θ to be well defined, but rather than impose these as nonlinear constraints on ϕ_i and θ_j directly (which would require either finite-difference Jacobians or an autodiff-aware eigendecomposition of a companion matrix, e.g. via **RTMB**), the raw parameters optimized over are instead a Durbin-Levinson / Jones partial autocorrelation (PACF) reparameterization $r_i^{ar}, r_j^{ma} \in (-1, 1)$, from which ϕ (guaranteed to lie in the stationarity region) and θ (guaranteed to lie in the invertibility region, via the well-known AR/MA duality) are recovered by a smooth, exact-autodiff- compatible recursion. This means only simple box constraints on r_i^{ar}, r_j^{ma} are required during optimization, with exact (rather than finite-difference) Jacobians of the mean equation available throughout via **TMB**. The raw PACF parameters (**arpacf/mapacf** in the model's **parmatrix**) are not directly interpretable; the transformed ϕ/θ coefficients are what **summary()** displays, and are also available via **arma_coefficients()**.

When **arma** is active, **fitted()** returns the (now genuinely time-varying) conditional mean $\mu_t = \mu + \sum_i \phi_i (y_{t-i} - \mu) + \sum_j \theta_j \varepsilon_{t-j}$ rather than the constant μ , and **residuals()** returns the true ARMA innovations $\varepsilon_t = y_t - \mu_t$ rather than $y_t - \mu$. Both **predict()** and **simulate()** account for the ARMA dynamics in the mean forecast/simulated paths, and **tsfilter()** continues the ARMA recursion when incrementally filtering newly appended observations. The combined pre-sample/burn-in length used internally is $\max(\text{GARCH order}, \text{ARMA order})$, so that e.g. an ARMA(2,2) mean equation with a GARCH(1,1) variance equation initializes correctly. ARMA support is available for all GARCH flavors described in Section 2.9, including **igarch** and **ewma**.

An entire AR and/or MA polynomial can instead be fixed at target coefficients rather than estimated, by setting **value** to the desired ar/ma coefficients directly (not r_i^{ar}/r_j^{ma}) and **estimate** = 0 on every **arpacf/mapacf** row of the relevant polynomial in the spec's **parmatrix**; the corresponding raw pacf values are computed automatically. Fixing only some (not all) of the lags of a given polynomial is not supported - since ϕ_i depends jointly on $r_1^{ar}, \dots, r_p^{ar}$ (and similarly for θ_j), there is no way to hold one lag's coefficient fixed while leaving others free without either giving up the stationarity/invertibility guarantee or reintroducing a nonlinear constraint - and raises an error, as does supplying a fixed target that is not itself stationary (AR) or invertible (MA).

2.2 Recursion Initialization (σ_0^δ)

When estimating a GARCH model, we need to initialize the variance to some value in order to start the recursion. In the **tsgarch** package we set all values of $\sigma_t^2, \forall t \leq 0$ to the sample variance or some other estimator which the user can choose from. Three options are available:

1. **unconditional:** $\left(\frac{1}{T} \sum_{j=1}^T \varepsilon_j^2\right)^{(\delta/2)}$
2. **sample:** $\left(\frac{1}{S} \sum_{j=1}^S \varepsilon_j^2\right)^{(\delta/2)}$, where S is the length of the sample to use.
3. **backcasting:** $\left(\lambda^T \hat{\sigma}^2 + (1 - \lambda) \sum_{j=0}^{T-1} \lambda^j \varepsilon^2 1 + j\right)^{\delta/2}$, $\lambda \in \{0, 1\}$

where : $\hat{\sigma}^2 = \frac{1}{T-1} \sum_{j=1}^T \varepsilon_j^2$

For the **backasting** method, setting $\lambda = 1$ we obtain the sample variance, else values of λ less than one and greater than zero yields the exponential smoothing backcast estimator. The exponent δ is for the power ARCH type models, otherwise it is set to 2 (variance). One popular commercial econometric software package defaults to backasting using a value of 0.7 for λ .

In addition to the variance, we also need to initialize the value of the ARCH component for $t \leq j$, by taking the sample average of the equation or part of the equation. Details of this initialization is provided for each model. Note that for all models, the initialization values of the ARCH(q) components will follow the user choice for the variance initialization.

2.3 Model Persistence, Long Run Variance and Half-Life

The persistence (P) of a GARCH model is a measure which quantifies the degree of volatility clustering and rate of decay. It also forms a bounding constraint (< 1) on the model dynamics in order to ensure stationarity. Another way to think about persistence is by looking at the unconditional variance of a GARCH model which is defined as

$$\hat{\sigma}^2 = \frac{\omega + \sum_{k=1}^m \xi_k \bar{\chi}_k}{1 - P} \quad (10)$$

where $\bar{\chi}_k$ is the sample mean of any external regressors.

Equation (10) illustrates that positivity of the unconditional variance requires $P \leq 1$, whilst existence of this value requires $P < 1$, which is not the case for the integrated GARCH model where $P = 1$ by design. The form that P takes will depend on the type of model, with the formulas provided in Section 2.9. Closely related to the persistence is the half-life measure which is defined as the number of periods it takes for a shock to revert half way back to the long run variance, and defined as $-\log_e(2)/\log_e(P)$.

A special note is warranted for the half-life of the Component GARCH (**cgarch**) model which is composed of a permanent and transitory component, each of which have a persistence (see (64)). The permanent component half-life, based on the estimate of ρ , measures the time taken for the long-run influence of a shock in volatility to revert by half towards its long run unconditional value, whereas the transitory component half-life accounts for the time taken for a shock's influence to revert to its long-run rate.

2.4 Variance Targeting ($\bar{\omega}$)

Variance targeting sets the value of the GARCH intercept (ω) to its long run estimate as :

$$\bar{\omega} = \hat{\sigma}^2 (1 - P) - \sum_{k=1}^m \xi_k \bar{\chi}_k \quad (11)$$

where $\hat{\sigma}^2$ is the unconditional variance of ε^2 , consistently estimated by its sample counterpart, P is the model persistence and $\bar{\chi}_k$ the mean of the external variance regressors (if present). A review of variance targeting

can be found in [Francq et al. \(2011\)](#). In this author's experience, more than 90% of model estimation problems come from trying to estimate ω as a result of parameter scaling issues. In **tsgarch**, despite attempts to apply some scaling rules during optimization, failure to converge will sometimes happen, in which case the model will be automatically re-estimated with variance targeting (the output will indicate when this has happened).

Unlike initialization estimators for σ_0^2 and ε_0^2 , the value of $\hat{\sigma}^2$ and $\hat{v}_j$ is based on the full sample.

2.5 External Regressors (χ)

Inclusion of additive external regressors has always been a little tricky when it comes to variance modelling due to the non-negativity constraint. This is an issue in all GARCH flavors with the exception of the exponential model. One way to deal with this is to constrain coefficients and regressors to be strictly positive which is not ideal. Another option, which is now offered in **tsgarch** is to have multiplicative regressors where the intercept is now calculated as follows:

$$\omega_t = \exp \left(\omega + \sum_{k=1}^m \xi_k \chi_{k,t} \right) \quad (12)$$

which does not require any bound constraints on either the constant ω or the regressors.

2.6 News Impact Curve (NIC)

[Engle and Ng \(1993\)](#) defined the news impact curve as a way to analyze the effect of news on the conditional variance by keeping constant the information dated $t - 2$ and earlier. Therefore, the long run variance of the model is used in place of σ_{t-1} and a range of values for ε_t are chosen to show how news of different sign and size impact the current conditional variance. A most interesting example of this found in the Family GARCH model of [Hentschel \(1995\)](#) which accommodates both shifts and rotations in the news impact curve, with the shift factor being the the main source of asymmetry for small shocks, and rotation driving larger shocks. A more detailed exposition, and compact representation of the news impact can be found in [Caporin and Costola \(2019\)](#) who define asymmetry of a GARCH model, for all shocks θ , as the case where:

$$\text{NIC}(\theta) \neq \text{NIC}(-\theta) \quad (13)$$

2.7 Standard Errors

The **tsgarch** package makes use of the methods available in the [sandwich](#) package to deliver a number of different estimators for the parameter covariance matrix (S).

Define the objective function (Ψ) as the log-likelihood of the GARCH model with respect to the data and parameters:

$$\Psi(y, x, \theta) = \sum_{t=1}^T \log \left(\frac{1}{\sigma_t} g(z_t | \zeta, \nu) \right) \quad (14)$$

where $\frac{1}{\sigma_t} g(z_t | \zeta, \nu)$ is defined in Section 2. The estimating (or score) function of the objective function is then:

$$\psi(y, x, \theta) = \frac{\partial \Psi(y, x, \theta)}{\partial \theta} \quad (15)$$

Inference about the parameter set θ relies on a central limit theorem (CLT) with $\sqrt{n}$ consistency:

$$\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{d} N(0, S(\theta)) \quad (16)$$

where $\xrightarrow{d}$ indicates convergence in distribution. The **sandwich** package defines the sandwich estimator $S(\theta)$ as:

$$S(\theta) = B(\theta) M(\theta) B(\theta) \quad (17)$$

where the meat (**M**) of the sandwich is the variance of the estimating function:

$$M(\theta) = \text{VAR}[\psi(y, x, \theta)] \quad (18)$$

and the bread (**B**) is the inverse of the expectation of its first derivative (ψ'):

$$B(\theta) = \left(E \left[-\psi'(y, x, \theta) \right] \right)^{-1} \quad (19)$$

In **tsgarch**, the following 4 estimators for the covariance matrix are defined:

2.7.1 Direct (H)

$$S(\theta) = B(\theta) = -H^{-1} \quad (20)$$

This makes use of the analytic hessian (H) at the optimal solution.

2.7.2 Outer product of the gradient (OP)

$$\begin{aligned} S(\theta) &= M(\theta) \\ &= \frac{1}{T} \sum_{t=1}^T \psi(y_t, x_t, \hat{\theta}) \psi(y_t, x_t, \hat{\theta})' \end{aligned} \quad (21)$$

The estimating function is essentially the Jacobian of the likelihood at each time step with respect to the parameters. Currently, this is based on the **jacobian** function from **numDeriv**, but will be replaced in a future version by the analytic solution from TMB once I figure out how to do this.

2.7.3 Quasi-Maximum Likelihood (QML)

$$\begin{aligned} S(\theta) &= B(\theta) M(\theta) B(\theta) \\ &= H^{-1} \left(\psi(y_t, x_t, \hat{\theta}) \psi(y_t, x_t, \hat{\theta})' \right) H^{-1} \end{aligned} \quad (22)$$

2.7.4 HAC (NW)

In the presence of residual autocorrelation, the HAC estimator is based on the weighted empirical autocorrelations of the empirical estimating functions:

$$\begin{aligned} M_{HAC} &= \frac{1}{T} \sum_{i,j=1}^T w_{|i-j|} \psi(y_t, x_t, \hat{\theta}) \psi(y_t, x_t, \hat{\theta})' \\ S(\theta) &= B(\theta) M_{HAC}(\theta) B(\theta) \end{aligned} \quad (23)$$

where the weights w can either be fixed or estimated using an appropriate choice of strategies. In the **tsgarch** package, the bandwidth of [Newey and West \(1994\)](#) is used to automatically calculate the lag and then the weights, based on the **bwNeweyWest** function from the **sandwich** package.

2.8 Parameter Scaling

The estimation strategy involves 2 passes through the optimizer. During the first pass, the parameters are first estimated using no-scaling. In the second pass, the problem is reformulated based on re-scaling the n parameters and their bounds by the vector $s = \sqrt{\{s_{1,1}^{-1}, s_{2,2}^{-1}, \dots, s_{n,n}^{-1}\}}$, where $s_{i,i}$ are the diagonals of the hessian at the previous solution.

The rescaled parameters and their bounds are passed to the optimizer, whilst the underlying C++ TMB code scales them back in order to perform the likelihood calculations and correctly adjust the analytic derivatives in the presence of this scaling vector.

The reason for performing this extra step, is to avoid some bad solutions as a result of large differences in scaling (particularly with respect to the constant ω), and to avoid issues with the derivatives. The optimal hessian and scores (the estimating function) in the output object are based on the scaled versions from the second pass which are then re-scaled back. This approach was found to offer substantial stability at a relatively low cost. Some justification for this method can be found in [Yang and Lee \(2010\)](#) and [Rao \(2019\)](#).

2.9 GARCH Flavors

The **tsgarch** package implements a number of different flavors of GARCH, including the option of 10 different distributions. The next subsections provide details of each model's formulation. For each model, the dynamics of the conditional mean take the following form:

$$\begin{aligned}\mu_t &= \mu \\ y_t &= \mu_t + \varepsilon_t, \quad \varepsilon_t \sim D(0, \sigma_t, \zeta, \nu, \lambda)\end{aligned}\tag{24}$$

where D is one of the available distributions from **tsdistributions**, ζ the skew parameter, ν the shape parameter and λ an additional shape parameter used in the Generalized Hyperbolic distribution. These additional distributional parameters may be present in some combination or not at all depending on the distribution. It is also possible to set μ to zero and pass a series of pre-filtered residual values from some other model as discussed in Section 1. This is the mean equation used by default (**arma** = **c(0,0)**); when an $\text{ARMA}(p, q)$ mean equation is instead requested, μ_t becomes the time-varying conditional mean of Equation 5 (Section 2.1) for every flavor below, with ε_t unchanged in its role as the innovation driving each flavor's conditional variance equation.

Note that the parameter symbols for the model equations presented in the following subsections will be exactly the same as those output by the package, with the **summary** method on estimated objects having the option to replace the names of parameters with symbols when transforming to a **flextable** object (**as_flextable** method on summary object).

2.9.1 Vanilla GARCH (*garch*)

The vanilla GARCH model of [Bollerslev \(1986\)](#) extended the ARCH model of [Engle \(1982\)](#) to include a moving average term making it more closely aligned with the literature on ARMA processes, and allowing for a wider range of behavior and more persistent volatility.

2.9.1.1 Equation

$$\begin{aligned}\omega_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} \\ \sigma_t^2 &= \omega_t + \sum_{j=1}^q \alpha_j \varepsilon_{t-j}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2\end{aligned}\tag{25}$$

In the presence of variance regressors, the choice to use **multiplicative** type intercept means that:

$$\omega_t = \exp \left(\omega + \sum_{k=1}^m \xi_k \chi_{k,t} \right) \quad (26)$$

in which case the bounds on ω and ξ_k are mostly free and we don't have to worry about the positivity of the variance.

2.9.1.2 Initialization

$$\begin{aligned} \sigma_{t-j}^2 &= \frac{1}{T} \sum_{t=1}^T \varepsilon_t^2, \quad (t-j) \leq 0 \\ \varepsilon_{t-j}^2 &= \frac{1}{T} \sum_{t=1}^T \varepsilon_t^2, \quad (t-j) \leq 0 \end{aligned} \quad (27)$$

where the first equation is used to initialize the GARCH recursion and the second one (identical to the first) is used for the ARCH recursion initialization.

This is the default choice, but other choices such as less than full *sample* and *backcasting* are also available and described in Section 2.2.

2.9.1.3 Persistence

$$P = \sum_{j=1}^q \alpha_j + \sum_{j=1}^p \beta_j \quad (28)$$

2.9.1.4 News Impact

$$\sigma_t^2 = \omega + \alpha \varepsilon_t^2 + \beta \bar{\sigma}^2 \quad (29)$$

where $\bar{\sigma}^2 = \frac{\omega + \sum_{k=1}^n \xi_k \bar{\chi}_k}{1-P}$ ³ and represents the long run unconditional variance in the (optional) presence of **m** variance regressors χ with coefficients ξ (see Section 2.4).

2.9.1.5 Model Constraints

- $1 > \alpha_j > 0$
- $1 > \beta_j > 0$
- $\omega > 0$
- $P < 1$

For higher order models, it is suggested that the constraints on α_j be relaxed to allow for non positive values. This can be achieved by changing the **lower** value in the **parmatrix** slot of the specification.

2.9.1.6 Forecast

$$\sigma_{t+h}^2 = \left(\omega + \sum_{k=1}^m \xi_k \chi_{k,t+h} \right) + \sum_{j=1}^q \alpha_j \sigma_{t+h-j}^2 + \sum_{j=1}^p \beta_j \sigma_{t+h-j}^2, \quad (h-j) > 0 \quad (30)$$

³If using a multiplicative intercept choice, then $\bar{\sigma}^2 = \frac{\exp \left(\omega + \sum_{k=1}^n \xi_k \bar{\chi}_k \right)}{1-P}$.

2.9.2 Integrated GARCH (*igarch*)

The integrated GARCH model of Engle and Bollerslev (1986) assumes that the persistence $P = 1$, hence shocks are permanent and the unconditional variance infinite. The motivation behind this model was to capture the long memory behavior observed in some financial time series.⁴ However, Nelson (1990) showed that the IGARCH process with no drift ($\omega = 0$) would converge to zero with probability one. In the presence of a drift term ($\omega > 0$) the process is neither covariance stationary nor does it have well-defined unconditional variance, though it still remains strictly stationary and ergodic. For truly long memory processes, other GARCH models should be considered such as the Fractionally Integrated GARCH (FIGARCH) or Hyperbolic GARCH (HYGARCH) which may be included at a later time.

2.9.2.1 Equation

$$\begin{aligned}\omega_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} \\ \sigma_t^2 &= \omega_t + \sum_{j=1}^q \alpha_j \varepsilon_{t-j}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2\end{aligned}\tag{31}$$

2.9.2.2 Persistence The persistence in the *igarch* model is set to 1 and forms a binding constraint on the parameters.

$$P = \sum_{j=1}^q \alpha_j + \sum_{j=1}^p \beta_j = 1\tag{32}$$

2.9.2.3 News Impact Not defined.

2.9.2.4 Model Constraints

- $1 > \alpha_j > 0$
- $1 > \beta_j > 0$
- $\omega > 0$
- $P = 1$

2.9.2.5 Forecast Same as the the vanilla GARCH forecast in Section 2.9.1.

2.9.3 Exponentially Weighted Moving Average (*ewma*)

The Exponentially Weighted Moving Average (*ewma*) model is a restricted *igarch* model where the drift term (ω) is set to zero. It has proven popular among some practitioners for it's simplicity and speed, with coefficients most often hard coded rather than estimated, based on prior knowledge. However, as mentioned in Section 2.9.2 the variance will converge to zero in a finite number of steps so it is unlikely to be a good model for anything but very short term forecasting.

2.9.3.1 Equation The *ewma* equation is usually written as $\sigma_t^2 = (1 - \lambda) \varepsilon_{t-1}^2 + \lambda \sigma_{t-1}^2$, but we present below the more general model which shows that it is a restricted *igarch* model with no drift (although it is always possible to include regressors).

$$\begin{aligned}\omega_t &= \sum_{k=1}^m \xi_k \chi_{k,t} \\ \sigma_t^2 &= \omega_t + \sum_{j=1}^q \alpha_j \varepsilon_{t-j}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2\end{aligned}\tag{33}$$

⁴Though many times it is also possible that this is the result of omitted structural breaks

2.9.3.2 Persistence Since this is simply a restricted `igarch` model, the persistence is set to 1 and forms a binding constraint on the parameters as in (32).

2.9.3.3 News Impact Not defined.

2.9.3.4 Forecast Same as the the vanilla GARCH forecast in Section 2.9.1 with ω set to zero.

2.9.4 Exponential GARCH (*egarch*)

The exponential GARCH model of Nelson (1991) allows for asymmetric effects between positive and negative returns, and does not require specific parameter restrictions to ensure positivity of the variance since the modelling is performed on the log variance.

2.9.4.1 Equation

$$\begin{aligned}\omega_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} \\ \log_e(\sigma_t^2) &= \omega_t + \sum_{j=1}^q (\alpha_j z_{t-j} + \gamma_j (|z_{t-j}| - E|z_{t-j}|)) + \sum_{j=1}^p \beta_j \log_e(\sigma_{t-j}^2)\end{aligned}\tag{34}$$

where $z_t = \frac{\varepsilon_t}{\sigma_t}$, with expectation of the absolute moment given by:

$$E|z_{t-j}| = \int_{-\infty}^{\infty} |z| D(z, 0, 1, \zeta, \nu, \lambda) dz\tag{35}$$

For symmetric distributions, the absolute moment is usually available in closed form (such as in the Gaussian case where it is $\sqrt{\frac{2}{\pi}}$). For other distributions this is calculated using Gauss-Kronrod quadrature in the C++ TMB code so that it is forms part of the autodiff tape.

2.9.4.2 Initialization

$$\begin{aligned}\log_e \sigma_{t-j}^2 &= \log_e \left(\frac{1}{T} \sum_{t=1}^T \varepsilon_t^2 \right), \quad (t-j) \leq 0 \\ z_{t-j}^2 &= 0, \quad (t-j) \leq 0 \\ (|z_{t-j}| - E|z_{t-j}|) &= 0, \quad (t-j) \leq 0\end{aligned}\tag{36}$$

where the first equation is used in the initialization of the GARCH recursion, whilst the second and third equations are for the ARCH initialization.

This is the default choice, but other choices such as less than full *sample* and *backcasting* are also available and described in Section 2.2.

2.9.4.3 Persistence The persistence has a rather simple form based on the sum of the moving average coefficients.

$$P = \sum_{j=1}^p \beta_j\tag{37}$$

2.9.4.4 Unconditional Variance The unconditional variance of an EGARCH(1,1) model, is given by the following equation:

$$\bar{\sigma}^2 = \exp \left(\frac{\omega + \sum_{k=1}^m \xi_k \bar{\chi}_k}{1 - \beta} \right) \prod_{i=1}^{\infty} E [\exp (\beta^{i-1} g(z_t))] \quad (38)$$

where $g(z_t) = \alpha z_t + \gamma (|z_t| - E|z_t|)$ (see [He et al. \(2002\)](#)), and $E [\exp (\beta^{i-1} g(z_t))]$ is calculated by numerical quadrature:

$$\int_{-\infty}^{\infty} \exp (\beta_1^{i-1} g(x)) D(x, 0, 1, \zeta, \nu, \lambda) dx \quad (39)$$

We approximate the infinite product by truncating i to 1000 which we found is more than sufficient for convergence.

For higher order GARCH model, i.e. $p, q > 1$, the unconditional variance is approximated via simulation, averaging the variance over 100 simulations of length 25000.

2.9.4.5 News Impact

$$\sigma_i^2 = \exp \left(\omega + \sum_{k=1}^m \xi_k \bar{\chi}_k + \alpha z_i + \gamma (|z_i| - E|z_i|) + \beta \log_e (\sigma^2) \right) \quad (40)$$

where $\bar{\sigma}^2$, represents the unconditional variance as given in Equation (38).

2.9.4.6 Model Constraints

- $P < 1$

2.9.4.7 Forecast For the $\max(p, q) \leq 1$, the forecast for $t + h, h > 1$ is:

$$\sigma_{t+h}^2 = (\sigma_{t+1}^2)^{\beta^{h-1}} \exp \left(\frac{1 - \beta^{h-1}}{1 - \beta} \left(\omega + \sum_{k=1}^m \xi_k \chi_{k,t+h} \right) \right) \prod_{i=1}^{\infty} E [\exp (\beta^{i-1} g(z_t))] \quad (41)$$

where $g(x) = (\alpha_1 x + \gamma_1 (|x| - \kappa))$, and κ as in (35). We approximate the infinite product in the equation by truncating it to 1000 terms. For higher order models, the forecast is approximated via simulation.

2.9.5 GJR GARCH (*gjrgarch*)

The GJR GARCH model of [Glosten et al. \(1993\)](#) models positive and negative shocks on the conditional variance asymmetrically using a leverage term for past squared, negative innovations via the indicator function I .

2.9.5.1 Equation

$$\begin{aligned} \omega_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} \\ \sigma_t^2 &= \omega_t + \sum_{j=1}^q (\alpha_j \varepsilon_{t-j}^2 + \gamma_j I_{[\varepsilon_{t-j} \leq 0]} \varepsilon_{t-j}^2) + \sum_{j=1}^p \beta_j \sigma_{t-j}^2 \end{aligned} \quad (42)$$

where γ_j now represents the *leverage* term. The indicator function I takes on value 1 for $\varepsilon \leq 0$ and 0 otherwise. Because of the presence of the indicator function, the persistence of the model now crucially depends on the asymmetry of the conditional distribution.

2.9.5.2 Initialization

$$\begin{aligned}\sigma_{t-j}^2 &= \frac{1}{T} \sum_{t=1}^T \varepsilon_t^2, \quad (t-j) \leq 0 \\ \varepsilon_{t-j}^2 &= \frac{1}{T} \sum_{t=1}^T \varepsilon_t^2, \quad (t-j) \leq 0 \\ I_{[\varepsilon_{t-j} \leq 0]} \varepsilon_{t-j}^2 &= \frac{1}{T} \sum_{t=i}^T I_{[\varepsilon_t \leq 0]} \varepsilon_t^2, \quad (t-j) \leq 0\end{aligned}\tag{43}$$

2.9.5.3 Persistence

$$P = \sum_{j=1}^q \alpha_j + \sum_{j=1}^p \beta_j + \sum_{j=1}^q \gamma_j \kappa\tag{44}$$

where κ is the expected value of $\varepsilon_t \leq 0$. Since this represents the probability of being less than zero, we can work directly with the standardized innovations when using the location scale distributions⁵, z_t :

$$\begin{aligned}\kappa &= E[I_{[z_t \leq 0]}] \\ &= \int_{-\infty}^0 D(z, 0, 1, \zeta, \nu, \lambda) dz\end{aligned}\tag{45}$$

For symmetric distributions this value is always 0.5, but for skewed distributions this is calculated using Gauss-Kronrod quadrature in the C++ TMB code so that it forms part of the autodiff tape allowing to also extract the Jacobian of this function for use with the inequality constraint in the **nloptr** solver.

2.9.5.4 News Impact

$$\sigma_i^2 = \omega + \sum_{k=1}^m \xi_k \bar{\chi}_k + \alpha \varepsilon_i^2 + \gamma I_{[\varepsilon_i \leq 0]} \varepsilon_i^2 + \beta \bar{\sigma}^2\tag{46}$$

where $\bar{\sigma}^2 = \frac{\omega + \sum_{k=1}^m \xi_k \bar{\chi}_k}{1-P}$, represents the long run unconditional variance.

2.9.5.5 Model Constraints

- $1 > \alpha_j > 0$
- $1 > \beta_j > 0$
- $\alpha_j + \gamma_j > 0$
- $\omega > 0$
- $P < 1$

Note that we also constrain $\gamma_j > -1$. The Jacobian of the inequality constraints is calculated either analytically or using autodiff. For higher order models, it is suggested that the constraints on α_j and γ_j be relaxed to allow for non positive values. This can be achieved by changing the **lower** value in the **parmatrix** slot of the specification.

⁵for details see the [tsdistributions](#) package.

2.9.5.6 Forecast

$$\sigma_{t+h}^2 = \omega + \sum_{k=1}^m \xi_k \chi_{k,t+h} + \sum_{j=1}^q \alpha_j \sigma_{t+h-j}^2 + \gamma_j \kappa \sigma_{t+h-j}^2 + \sum_{j=1}^p \beta_j \sigma_{t+h-j}^2, \quad (h-j) > 0 \quad (47)$$

where κ is the expected value of $\varepsilon_t \leq 0$, see (45).

2.9.6 Asymmetric Power ARCH (*aparch*)

The asymmetric power ARCH model of [Ding et al. \(1993\)](#) allows for both leverage and the Taylor effect, named after [Taylor \(1986\)](#) who observed that the sample autocorrelation of absolute returns was usually larger than that of squared returns.

2.9.6.1 Equation

$$\begin{aligned} \omega_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} \\ \sigma_t^\delta &= \omega_t + \sum_{j=1}^q \alpha_j (|\varepsilon_{t-j}| - \gamma_j \varepsilon_{t-j})^\delta + \sum_{j=1}^p \beta_j \sigma_{t-j}^\delta \end{aligned} \quad (48)$$

where $\delta \in \mathbb{R}^+$, being a Box-Cox transformation of σ_t , and γ_j the coefficient in the leverage term.

2.9.6.2 Initialization

See [Laurent \(2004\)](#) page 52:

$$\begin{aligned} \sigma_{t-j}^\delta &= \left(\frac{1}{T} \sum_{t=1}^T \varepsilon_t^2 \right)^{(\delta/2)}, \quad (t-j) \leq 0 \\ (|\varepsilon_{t-j}| - \gamma_j \varepsilon_{t-j})^\delta &= \frac{1}{T} \sum_{t=i}^T (|\varepsilon_t| - \gamma_j \varepsilon_t)^\delta, \quad (t-j) \leq 0 \end{aligned} \quad (49)$$

2.9.6.3 Persistence

$$P = \sum_{j=1}^p \beta_j + \sum_{j=1}^q \alpha_j \kappa_j \quad (50)$$

where κ_j is the expected value of the standardized residuals z_t under the Box-Cox transformation of the term which includes the leverage coefficient:

$$\begin{aligned} \kappa_j &= E(|z_{t-j}| - \gamma_j z_{t-j})^\delta \\ &= \int_{-\infty}^{\infty} (|z| - \gamma_j z)^\delta D(z, 0, 1, \zeta, \nu, \lambda) dz \end{aligned} \quad (51)$$

For symmetric distributions there are closed form solutions for this expression, but for skewed distributions this is calculated using Gauss-Kronrod quadrature in the C++ TMB code so that it is forms part of the autodiff tape allowing to also extract the Jacobian of this function for use with the inequality constraint in the **nloptr** solver.

2.9.6.4 News Impact

$$\sigma_i^2 = \left(\omega + \sum_{k=1}^m \xi_k \bar{\chi}_k + \alpha (|\varepsilon_i| - \gamma \varepsilon_i)^\delta + \beta \bar{\sigma}^\delta \right)^{\frac{2}{\delta}} \quad (52)$$

where $\bar{\sigma}^\delta = \frac{\omega + \sum_{k=1}^m \xi_k \bar{\chi}_k}{1-P}$, represents the long run unconditional volatility raised to the power of δ .

2.9.6.5 Model Constraints

- $1 > \alpha_j > 0$
- $1 > \beta_j > 0$
- $|\gamma_j| < 1$
- $\delta > 0$
- $\omega > 0$
- $P < 1$

Non-negativity constraints on α_j may be relaxed for higher order models by changing the `lower` parameter in the `parmatrix` of the specification object prior to estimation.

2.9.6.6 Forecast

$$\sigma_{t+h}^2 = \left(\omega + \sum_{k=1}^m \xi_k \chi_{k,t+h} + \sum_{j=1}^q \alpha_j \kappa_j \sigma_{t+h-j}^\delta + \sum_{j=1}^p \beta_j \sigma_{t+h-j}^\delta \right)^{2/\delta}, \quad (h-j) > 0 \quad (53)$$

where $\kappa_j = E(|z_{t-j}| - \gamma_j z_{t-j})^\delta$, see (51).

2.9.7 Family GARCH (*fgarch*)

The family GARCH model of [Hentschel \(1995\)](#) is a large omnibus model which subsumes some of the most popular GARCH models. It allows for both shifts and rotations in the news impact curve, where the shift is the main source of asymmetry for small shocks while rotation drives larger shocks. The following restrictions in the parameters leads to specific models:

- GARCH: $\delta = 2, \eta_j = \gamma_j = 0$
- Absolute Value GARCH: $\delta = 1, |\gamma_j| \leq 1$
- GJR GARCH: $\delta = 2, \eta_j = 0$
- Threshold GARCH: $\delta = 1, \eta_j = 0, |\gamma_j| \leq 1$
- Nonlinear GARCH: $\gamma_j = \eta_j = 0$
- Nonlinear Asymmetric GARCH: $\delta = 2, \gamma_j = 0$
- APARCH: $\eta_j = 0, |\gamma_j| \leq 1$

2.9.7.1 Equation

$$\begin{aligned} \omega_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} \\ \sigma_t^\delta &= \omega_t + \sum_{j=1}^q \alpha_j \sigma_{t-j}^\delta (|z_{t-j} - \eta_j| - \gamma_j (z_{t-j} - \eta_j))^\delta + \sum_{j=1}^p \beta_j \sigma_{t-j}^\delta \end{aligned} \quad (54)$$

The original formulation, takes a slightly different form:

$$\begin{aligned}
\omega_t &= \omega + \sum_{k=1}^m \xi_j \chi_{k,t} \\
\sigma_t^\delta &= \omega_t + \sum_{j=1}^q \alpha_j \sigma_{t-j}^\delta (|z_{t-j} - \eta_j| - \gamma_j (z_{t-j} - \eta_j))^\lambda + \sum_{j=1}^p \beta_j \sigma_{t-j}^\delta
\end{aligned} \tag{55}$$

which accommodates the exponential GARCH model as well as a more flexible transformation, albeit more computationally demanding. The **tsgarch** package does not adopt this more general formulation, instead setting $\lambda = \delta$.

2.9.7.2 Initialization

$$\begin{aligned}
\sigma_{t-j}^\delta &= \left(\frac{1}{T} \sum_{t=1}^T \varepsilon_t^2 \right)^{(\delta/2)}, \quad (t-j) \leq 0 \\
(|z_{t-j} - \eta_j| - \gamma_j (z_{t-j} - \eta_j))^\delta &= \frac{1}{T} \sum_{t=i}^T (|z_t - \eta_j| - \gamma_j (z_t - \eta_j))^\delta, \quad (t-j) \leq 0
\end{aligned} \tag{56}$$

2.9.7.3 Persistence

$$P = \sum_{j=1}^p \beta_j + \sum_{j=1}^q \alpha_j \kappa_j \tag{57}$$

where κ_j is the expected value of the standardized residuals z_t under the Box-Cox transformation of the absolute value asymmetry term

$$\begin{aligned}
\kappa_j &= E(|z_{t-j} - \eta_j| - \gamma_j (z_{t-j} - \eta_j))^\delta \\
&= \int_{-\infty}^{\infty} (|z - \eta_j| - \gamma_j (z - \eta_j))^\delta D(z, 0, 1, \zeta, \nu, \lambda) dz
\end{aligned} \tag{58}$$

There are no simple closed form solutions for this expression so it is calculated using Gauss-Kronrod quadrature in the C++ TMB code so that it forms part of the autodiff tape allowing to also extract the Jacobian of this function for use with the inequality constraint in the **nloptr** solver.

2.9.7.4 News Impact

$$\sigma_i^2 = \left(\omega + \sum_{k=1}^m \xi_k \tilde{\chi}_k + \alpha \bar{\sigma}^\delta (|z_i - \eta| - \gamma (z_i - \eta))^\delta + \beta \bar{\sigma}^\delta \right)^{\frac{2}{\delta}} \tag{59}$$

where $\bar{\sigma}^\delta = \frac{\omega + \sum_{k=1}^m \xi_k \tilde{\chi}_k}{1-P}$, represents the long run unconditional volatility raised to the power of δ .

2.9.7.5 Model Constraints

- $1 > \alpha_j > 0$
- $1 > \beta_j > 0$
- $|\gamma_j| < 1$
- $-10 < |\eta_j| < 10$
- $\delta > 0$
- $\omega > 0$

- $P < 1$

Non-negativity constraints on α_j may be relaxed for higher order models by changing the **lower** parameter in the **parmatrix** of the specification object prior to estimation.

2.9.7.6 Forecast

$$\sigma_{t+h}^2 = \left(\omega + \sum_{k=1}^m \xi_k \chi_{k,t+h} + \sum_{j=1}^q \alpha_j \kappa_j \sigma_{t+h-j}^\delta + \sum_{j=1}^p \beta_j \sigma_{t+h-j}^\delta \right)^{2/\delta}, \quad (h-j) > 0 \quad (60)$$

where κ_j is the expected value of the standardized residuals z_t under the Box-Cox transformation of the absolute value asymmetry term, see (58).

2.9.8 Component GARCH (*cgarch*)

The model of Lee and Engle (1999) decomposes the conditional variance into a permanent and transitory component so as to investigate the long- and short-run movements of volatility affecting securities.

2.9.8.1 Equation

$$\begin{aligned} q_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} + \rho q_{t-1} + \phi (\varepsilon_{t-1}^2 - \sigma_{t-1}^2) \\ \sigma_t^2 &= q_t + \sum_{j=1}^q \alpha_j (\varepsilon_{t-j}^2 - q_{t-j}) + \sum_{j=1}^p \beta_j (\sigma_{t-j}^2 - q_{t-j}) \end{aligned} \quad (61)$$

The process can be re-written in an alternative form to better highlight the decomposition of the permanent and transitory components, shown below for the Component GARCH(1,1) model:

$$\begin{aligned} \text{Permanent:} \quad q_t &= \omega + \sum_{k=1}^m \xi_k \chi_{k,t} + \rho q_{t-1} + \phi (\varepsilon_{t-1}^2 - \sigma_{t-1}^2) \\ \text{Transitory:} \quad s_t &= (\alpha + \beta) s_{t-1} + \alpha (\varepsilon_{t-1}^2 - \sigma_{t-1}^2) \\ \text{Total:} \quad \sigma_t^2 &= q_t + s_t \end{aligned} \quad (62)$$

The parameters α and ϕ correspond to immediate impacts of volatility shocks $(\varepsilon_{t-j}^2 - \sigma_{t-j}^2)$ on the transitory and permanent components, whereas $(\alpha + \beta)$ and ρ measure the persistence of the transitory and permanent components, respectively. The model as currently implemented allows for higher order in the transitory component but no higher orders in the permanent component. The regressors only enter through the permanent component equation.⁶

2.9.8.2 Initialization The transitory component is initialized to 0, whilst the squared residuals, variance and permanent component are initialized to the sample variance. As a result, the initial squared residuals and variance cancel out during initialization.

$$\begin{aligned} \varepsilon_{t-j}^2, \sigma_{t-j}^2, q_{t-j} &= \frac{1}{T} \sum_{t=1}^T \varepsilon_t^2, \quad (t-j) \leq 0 \\ s_{t-j} &= 0, \quad (t-j) \leq 0 \end{aligned} \quad (63)$$

⁶Some implementations in other software packages allow for more flexibility, allowing regressors to enter into both components.

2.9.8.3 Persistence

$$\begin{aligned} \text{Transitory:} \quad P^T &= \sum_{j=1}^q \alpha_j + \sum_{j=1}^p \beta_j \\ \text{Permanent:} \quad P^P &= \rho \end{aligned} \tag{64}$$

2.9.8.4 News Impact Since the component GARCH model can be represented as a restricted GARCH(2,2) model, we derive the news impact curve using this representation to arrive at the following equation:

$$\sigma_t^2 = \omega + \sum_{k=1}^m \xi_k \bar{\chi}_k + \rho \bar{\sigma}^2 + \phi (\varepsilon_t^2 - \bar{\sigma}^2) + \alpha (\varepsilon_t^2 - \bar{\sigma}^2) \tag{65}$$

where the unconditional variance $\bar{\sigma}^2 = \frac{\omega + \sum_{k=1}^m \xi_k \bar{\chi}_k}{1-\rho}$.

2.9.8.5 Constraints

- $1 > \alpha_j > 0$
- $1 > \beta_j > 0$
- $1 > \phi > 0$
- $1 > \rho > 0$
- $\omega > 0$
- $1 > P^P > P^T > 0$
- $\beta > \phi$

2.9.8.6 Forecast

$$\begin{aligned} q_{t+h} &= \left(\frac{\omega + \sum_{k=1}^m \xi_k \chi_{k,t+h}}{1-\rho} \right) + \rho^k \left(q_t - \frac{\omega + \sum_{k=1}^m \xi_k \chi_{k,t+h}}{1-\rho} \right) \\ s_{t+h} &= P_T^h (\sigma_t^2 - q_t) \\ \sigma_{t+h}^2 &= q_{t+h} + s_{t+h} \end{aligned} \tag{66}$$

where P_T is the transitory persistence as defined in (64). As h grows larger, the forecast converges to

$$\frac{\omega + \sum_{k=1}^m \xi_k \chi_{k,t+h}}{1-\rho}.$$

3 Package Methods

The **tsgarch** package includes a number of methods which can be used with either the **tsgarch.spec** or **tsgarch.estimate** objects. These include standard methods from the **stats** package such **residuals**, **fitted**, **sigma**, **logLik**, **AIC**, **BIC**, **confint**, **vcov**, **coef**, **predict** and **simulate**. Additionally, the following methods from the **tsmethods** package are included:

- **tsfilter**: online filtering
- **tsbacktest**: backtesting
- **tsprofile**: parameter profiling through simulation/estimation
- **unconditional**: unconditional variance
- **tsequation**: model equations (used mainly by the **as_flextable** method)
- **pit**: probability integral transform

- **persistence**: model persistence

Also included from the **sandwich** package are methods for **estfun** and **bread**.

Additionally, **arma_coefficients** returns the estimated AR/MA coefficients (Section 2.1) implied by the raw partial autocorrelation parameters used internally during estimation, for models specified with a non-trivial **arma** order.

4 Conclusion

The package is still in active development. Additional models may be added, porting some of the remaining **rugarch** models, but the focus is likely to be on models for high frequency data and porting the multivariate models from **rmgarch**.

A separate package called **tstests** includes a number of tests which were previously part of **rugarch** is already on CRAN.

5 References

References

- T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.
- Massimiliano Caporin and Michele Costola. Asymmetry and leverage in garch models: A news impact curve perspective. *Applied Economics*, 51(31):3345–3364, 2019.
- Amélie Charles and Olivier Darné. The accuracy of asymmetric garch model estimation. *International Economics*, 157:179–202, 2019.
- Z. Ding, C.W.J. Granger, and R.F. Engle. A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, 1(1):83–106, 1993.
- R.F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.
- R.F. Engle and T. Bollerslev. Modelling the persistence of conditional variances. *Econometric Reviews*, 5(1): 1–50, 1986.
- Robert F Engle and Victor K Ng. Measuring and testing the impact of news on volatility. *The Journal of Finance*, 48(5):1749–1778, 1993.
- Christian Francq, Lajos Horvath, and Jean-Michel Zakoian. Merits and drawbacks of variance targeting in garch models. *Journal of Financial Econometrics*, 9(4):619–656, 2011.
- L.R. Glosten, R. Jagannathan, and D.E. Runkle. On the relation between the expected value and the volatility of the nominal excess return on stocks. *Journal of Finance*, 48(5):1779–1801, 1993.
- Bruce E Hansen. Autoregressive conditional density estimation. *International Economic Review*, pages 705–730, 1994.
- Changli He, Timo Terasvirta, and Hans Malmsten. Moment structure of a family of first-order exponential garch models. *Econometric Theory*, 18(4):868–885, 2002.
- L. Hentschel. All in the family nesting symmetric and asymmetric garch models. *Journal of Financial Economics*, 39(1):71–104, 1995.
- Chelsey Hill and BD McCullough. On the accuracy of garch estimation in r packages. *Econometric Research in Finance*, 4(2):133–156, 2019.
- Sebastien Laurent. Analytical derivatives of the aparch model. *Computational Economics*, 24:51–57, 2004.

- G.J. Lee and R.F. Engle. A permanent and transitory component model of stock return volatility. In *Cointegration Causality and Forecasting A Festschrift in Honor of Clive WJ Granger*, pages 475–497. Oxford University Press, 1999.
- Daniel B Nelson. Stationarity and persistence in the garch (1, 1) model. *Econometric theory*, 6(3):318–334, 1990.
- Daniel B Nelson. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society*, pages 347–370, 1991.
- Whitney K Newey and Kenneth D West. Automatic lag selection in covariance matrix estimation. *The Review of Economic Studies*, 61(4):631–653, 1994.
- Singiresu S Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.
- S.J. Taylor. *Modelling Financial Time Series*. Wiley, 1986.
- Kyung-won Yang and Tai-yong Lee. Heuristic scaling method for efficient parameter estimation. *Chemical Engineering Research and Design*, 88(5-6):520–528, 2010.