Introduction to the rugarch package.
(Version 1.4-3)

Alexios Ghalanos

June 20, 2025

Contents

1 Introduction

The pioneering work of Box et al. (1994) in the area of autoregressive moving average models paved the way for related work in the area of volatility modelling with the introduction of ARCH and then GARCH models by Engle (1982) and Bollerslev (1986), respectively. In terms of the statistical framework, these models provide motion dynamics for the dependency in the conditional time variation of the distributional parameters of the mean and variance, in an attempt to capture such phenomena as autocorrelation in returns and squared returns. Extensions to these models have included more sophisticated dynamics such as threshold models to capture the asymmetry in the news impact, as well as distributions other than the normal to account for the skewness and excess kurtosis observed in practice. In a further extension, Hansen (1994) generalized the `GARCH` models to capture time variation in the full density parameters, with the Autoregressive Conditional Density Model[1], relaxing the assumption that the conditional distribution of the standardized innovations is independent of the conditioning information.

The `rugarch` package aims to provide for a comprehensive set of methods for modelling univariate `GARCH` processes, including fitting, filtering, forecasting, simulation as well as diagnostic tools including plots and various tests. Additional methods such as rolling estimation, bootstrap forecasting and simulated parameter density to evaluate model uncertainty provide a rich environment for the modelling of these processes. This document discusses the finer details of the included models and conditional distributions and how they are implemented in the package with numerous examples.

The `rugarch` package is available on CRAN (`http://cran.r-project.org/web/packages/rugarch/index.html`) and the development version on bitbucket (`https://bitbucket.org/alexiosg`). Some online examples and demos are available on my website (`http://www.unstarched.net`).

The package is provided AS IS, without any implied warranty as to its accuracy or suitability. A lot of time and effort has gone into the development of this package, and it is offered under the GPL-3 license in the spirit of open knowledge sharing and dissemination. If you do use the model in published work DO remember to cite the package and author (type `citation("rugarch")` for the appropriate BibTeX entry) , and if you have used it and found it useful, drop me a note and let me know.

USE THE R-SIG-FINANCE MAILING LIST FOR QUESTIONS.

A section on FAQ is included at the end of this document.

2 Model Specification

This section discusses the key step in the modelling process, namely that of the specification. This is defined via a call to the `ugarchspec` function,

```
> args(ugarchspec)

function (variance.model = list(model = "sGARCH", garchOrder = c(1,
    1), submodel = NULL, external.regressors = NULL, variance.targeting = FALSE),
    mean.model = list(armaOrder = c(1, 1), include.mean = TRUE,
        archm = FALSE, archpow = 1, arfima = FALSE, external.regressors = NULL,
        archex = FALSE), distribution.model = "norm", start.pars = list(),
    fixed.pars = list(), ...)
```

Thus a model, in the `rugarch` package, may be described by the dynamics of the conditional mean and variance, and the distribution to which they belong, which determines any additional

[1] The **racd** package is now available from my bitbucket repository.

parameters. The following sub-sections will outline the background and details of the dynamics and distributions implemented in the package.

2.1 Univariate ARFIMAX Models

The univariate GARCH specification allows to define dynamics for the conditional mean from the general ARFIMAX model with the addition of ARCH-in-mean effects introduced in Engle et al. (1987). The ARFIMAX-ARCH-in-mean specification may be formally defined as,

$$\Phi(L)(1-L)^d(y_t-\mu_t)=\Theta(L)\varepsilon_t, \tag{1}$$

with the left hand side denoting the Fractional AR specification on the demeaned data and the right hand side the MA specification on the residuals. (L) is the lag operator, $(1-L)^d$ the long memory fractional process with $0<d<1$, and equivalent to the Hurst Exponent H - 0.5, and μ_t defined as,

$$\mu_t=\mu+\sum_{i=1}^{m-n}\delta_i x_{i,t}+\sum_{i=m-n+1}^{m}\delta_i x_{i,t}\sigma_t+\xi\sigma_t^k, \tag{2}$$

where we allow for `m` external regressors x of which n (last n of m) may optionally be multiplied by the conditional standard deviation σ_t, and ARCH-in-mean on either the conditional standard deviation, $k=1$ or conditional variance $k=2$. These options can all be passed via the arguments in the `mean.model` list in the `ugarchspec` function,

- armaOrder (default = (1,1). The order of the ARMA model.)
- include.mean (default = TRUE. Whether the mean is modelled.)
- archm (default = FALSE. The ARCH-in-mean parameter.)
- archpow (default = 1 for standard deviation, else 2 for variance.)
- arfima (default = FALSE. Whether to use fractional differencing.)
- external.regressors (default = NULL. A matrix of external regressors of the same length as the data.)
- archex (default = FALSE. Either FALSE or integer denoting the number of external regressors from the end of the matrix to multiply by the conditional standard deviation.).

Since the specification allows for both fixed and starting parameters to be passed, it is useful to provide the naming convention for these here,

- AR parameters are 'ar1', 'ar2', ...,
- MA parameters are 'ma1', 'ma2', ...,
- mean parameter is 'mu'
- archm parameter is 'archm'
- the arfima parameter is 'arfima'
- the external regressor parameters are 'mxreg1', 'mxreg2', ...,

Note that estimation of the mean and variance equations in the maximization of the likelihood is carried out jointly in a single step. While it is perfectly possible and consistent to perform a 2-step estimation, the one step approach results in greater efficiency, particularly for smaller datasets.

2.2 Univariate GARCH Models

In GARCH models, the density function is usually written in terms of the location and scale parameters, normalized to give zero mean and unit variance,

$$\alpha_t = (\mu_t, \sigma_t, \omega), \tag{3}$$

where the conditional mean is given by

$$\mu_t = \mu(\theta, x_t) = E(y_t|x_t), \tag{4}$$

and the conditional variance is,

$$\sigma_t^2 = \sigma^2(\theta, x_t) = E((y_t - \mu_t)^2|x_t), \tag{5}$$

with $\omega = \omega(\theta, x_t)$ denoting the remaining parameters of the distribution, perhaps a shape and skew parameter. The conditional mean and variance are used to scale the innovations,

$$z_t(\theta) = \frac{y_t - \mu(\theta, x_t)}{\sigma(\theta, x_t)}, \tag{6}$$

having conditional density which may be written as,

$$g(z|\omega) = \frac{d}{dz}P(z_t < z|\omega), \tag{7}$$

and related to $f(y|\alpha)$ by,

$$f(y_t|\mu_t, \sigma_t^2, \omega) = \frac{1}{\sigma_t}g(z_t|\omega). \tag{8}$$

The `rugarch` package implements a rich set of univariate GARCH models and allows for the inclusion of external regressors in the variance equation as well as the possibility of using variance targeting as in Engle and Mezrich (1995). These options can all be passed via the arguments in the `variance.model` list in the `ugarchspec` function,

- model (default = 'sGARCH' (vanilla GARCH). Valid models are 'iGARCH', 'gjrGARCH', 'eGARCH', 'apARCH','fGARCH','csGARCH' and 'mcsGARCH').
- garchOrder (default = c(1,1). The order of the GARCH model.)
- submodel (default = NULL. In the case of the 'fGARCH' omnibus model, valid choices are 'GARCH', 'TGARCH', 'GJRGARCH', 'AVGARCH', 'NGARCH', 'NAGARCH', 'APARCH' and 'ALLGARCH')
- external.regressors (default = NULL. A matrix of external regressors of the same length as the data).
- variance.targeting (default = FALSE. Whether to include variance targeting. It is also possible to pass a numeric value instead of a logical, in which case it is used for the calculation instead of the variance of the conditional mean equation residuals).

The rest of this section discusses the various flavors of GARCH implemented in the package, while Section 2.3 discusses the distributions implemented and their standardization for use in GARCH processes.

2.2.1 The standard GARCH model ('sGARCH')

The standard GARCH model (Bollerslev (1986)) may be written as:

$$\sigma_t^2 = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{9}$$

with σ_t^2 denoting the conditional variance, ω the intercept and ε_t^2 the residuals from the mean filtration process discussed previously. The GARCH order is defined by (q, p) (ARCH, GARCH), with possibly `m` external regressors v_j which are passed *pre-lagged*. If variance targeting is used, then ω is replaced by,

$$\bar{\sigma}^2 \left(1 - \hat{P}\right) - \sum_{j=1}^{m} \zeta_j \bar{v}_j \tag{10}$$

where $\bar{\sigma}^2$ is the unconditional variance of ε^2 which is consistently estimated by its sample counterpart at every iteration of the solver following the mean equation filtration, and $\bar{v}_j$ represents the sample mean of the j^{th} external regressors in the variance equation (assuming stationarity), and $\hat{P}$ is the persistence and defined below. If a numeric value was provided to the *variance.targeting* option in the specification (instead of logical), this will be used instead of $\bar{\sigma}^2$ for the calculation.[2] One of the key features of the observed behavior of financial data which GARCH models capture is volatility clustering which may be quantified in the persistence parameter $\hat{P}$. For the 'sGARCH' model this may be calculated as,

$$\hat{P} = \sum_{j=1}^{q} \alpha_j + \sum_{j=1}^{p} \beta_j. \tag{11}$$

Related to this measure is the 'half-life' (call it $h2l$) defined as the number of days it takes for half of the expected reversion back towards $E\left(\sigma^2\right)$ to occur,

$$h2l = \frac{-\log_e 2}{\log_e \hat{P}}. \tag{12}$$

Finally, the unconditional variance of the model $\hat{\sigma}^2$, and related to its persistence, is,

$$\hat{\sigma}^2 = \frac{\hat{\omega}}{1 - \hat{P}}, \tag{13}$$

where $\hat{\omega}$ is the estimated value of the intercept from the GARCH model. The naming conventions for passing fixed or starting parameters for this model are:

- ARCH(q) parameters are 'alpha1', 'alpha2', ...,
- GARCH(p) parameters are 'beta1', 'beta2', ...,
- variance intercept parameter is 'omega'
- the external regressor parameters are 'vxreg1', 'vxreg2', ...,

[2] Note that this should represent a value related to the variance in the plain vanilla GARCH model. In more general models such as the APARCH, this is a value related to σ^δ, which may not be obvious since δ is not known prior to estimation, and therefore care should be taken in those cases. Finally, if scaling is used in the estimation (via the fit.control option), this value will also be automatically scale adjusted by the routine.

2.2.2 The integrated GARCH model ('iGARCH')

The integrated GARCH model (see Engle and Bollerslev (1986)) assumes that the persistence $\hat{P} = 1$, and imposes this during the estimation procedure. Because of unit persistence, none of the other results can be calculated (i.e. unconditional variance, half life etc). The stationarity of the model has been established in the literature, but one should investigate the possibility of omitted structural breaks before adopting the iGARCH as the model of choice. The way the package enforces the sum of the ARCH and GARCH parameters to be 1, is by subtracting $1 - \sum_{i=1}^{q} \alpha_i - \sum_{i>1}^{p} \beta_i$, so that the last beta is never estimated but instead calculated.

2.2.3 The exponential GARCH model

The exponential model of Nelson (1991) is defined as,

$$\log_e\left(\sigma_t^2\right) = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \left(\alpha_j z_{t-j} + \gamma_j \left(|z_{t-j}| - E\left|z_{t-j}\right|\right)\right) + \sum_{j=1}^{p} \beta_j \log_e\left(\sigma_{t-j}^2\right) \tag{14}$$

where the coefficient α_j captures the sign effect and γ_j the size effect. The expected value of the absolute standardized innovation, z_t is,

$$E\left|z_t\right| = \int_{-\infty}^{\infty} |z| f\left(z, 0, 1, ...\right) dz \tag{15}$$

The persistence $\hat{P}$ is given by,

$$\hat{P} = \sum_{j=1}^{p} \beta_j. \tag{16}$$

If variance targeting is used, then ω is replaced by,

$$\log_e\left(\bar{\sigma}^2\right)\left(1 - \hat{P}\right) - \sum_{j=1}^{m} \zeta_j \bar{v}_j \tag{17}$$

The unconditional variance and half life follow from the persistence parameter and are calculated as in Section 2.2.1.

2.2.4 The GJR-GARCH model ('gjrGARCH')

The GJR GARCH model of Glosten et al. (1993) models positive and negative shocks on the conditional variance asymmetrically via the use of the indicator function I,

$$\sigma_t^2 = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \left(\alpha_j \varepsilon_{t-j}^2 + \gamma_j I_{t-j} \varepsilon_{t-j}^2\right) + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{18}$$

where γ_j now represents the 'leverage' term. The indicator function I takes on value of 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 used. The persistence of the model $\hat{P}$ is,

$$\hat{P} = \sum_{j=1}^{q} \alpha_j + \sum_{j=1}^{p} \beta_j + \sum_{j=1}^{q} \gamma_j \kappa, \tag{19}$$

where κ is the expected value of the standardized residuals z_t below zero (effectively the probability of being below zero),

$$\kappa = E\left[I_{t-j} z_{t-j}^2\right] = \int_{-\infty}^{0} f\left(z, 0, 1, ...\right) dz \tag{20}$$

where f is the standardized conditional density with any additional skew and shape parameters $(\dots)$. In the case of symmetric distributions the value of κ is simply equal to 0.5. The variance targeting, half-life and unconditional variance follow from the persistence parameter and are calculated as in Section 2.2.1. The naming conventions for passing fixed or starting parameters for this model are:

- ARCH(q) parameters are 'alpha1', 'alpha2', ...,
- Leverage(q) parameters are 'gamma1', 'gamma2', ...,
- GARCH(p) parameters are 'beta1', 'beta2', ...,
- variance intercept parameter is 'omega'
- the external regressor parameters are 'vxreg1', 'vxreg2', ...,

Note that the Leverage parameter follows the order of the ARCH parameter.

2.2.5 The asymmetric power ARCH model ('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.

$$\sigma_t^{\delta} = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \alpha_j (|\varepsilon_{t-j}| - \gamma_j \varepsilon_{t-j})^{\delta} + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^{\delta} \tag{21}$$

where $\delta \in \mathbb{R}^+$, being a Box-Cox transformation of σ_t, and γ_j the coefficient in the leverage term. Various submodels arise from this model:

- The simple GARCH model of Bollerslev (1986) when $\delta = 2$ and $\gamma_j = 0$.
- The Absolute Value GARCH (AVGARCH) model of Taylor (1986) and Schwert (1990) when $\delta = 1$ and $\gamma_j = 0$.
- The GJR GARCH (GJRGARCH) model of Glosten et al. (1993) when $\delta = 2$.
- The Threshold GARCH (TGARCH) model of Zakoian (1994) when $\delta = 1$.
- The Nonlinear ARCH model of Higgins et al. (1992) when $\gamma_j = 0$ and $\beta_j = 0$.
- The Log ARCH model of Geweke (1986) and Pantula (1986) when $\delta \to 0$.

The persistence of the model is given by,

$$\hat{P} = \sum_{j=1}^{p} \beta_j + \sum_{j=1}^{q} \alpha_j \kappa_j \tag{22}$$

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 γ_j,

$$\kappa_j = E(|z| - \gamma_j z)^\delta = \int_{-\infty}^{\infty} (|z| - \gamma_j z)^\delta f(z, 0, 1, ...)\, dz \tag{23}$$

If variance targeting is used, then ω is replaced by,

$$\bar{\sigma}^\delta \left(1 - \hat{P}\right) - \sum_{j=1}^{m} \zeta_j \bar{v}_j. \tag{24}$$

Finally, the unconditional variance of the model $\hat{\sigma}^2$ is,

$$\hat{\sigma}^2 = \left(\frac{\hat{\omega}}{1 - \hat{P}}\right)^{2/\delta} \tag{25}$$

where $\hat{\omega}$ is the estimated value of the intercept from the GARCH model. The half-life follows from the persistence parameter and is calculated as in Section 2.2.1. The naming conventions for passing fixed or starting parameters for this model are:

- ARCH(q) parameters are 'alpha1', 'alpha2', ...,
- Leverage(q) parameters are 'gamma1', 'gamma2', ...,
- Power parameter is 'delta',
- GARCH(p) parameters are 'beta1', 'beta2', ...,
- variance intercept parameter is 'omega'
- the external regressor parameters are 'vxreg1', 'vxreg2', ...,

In particular, to obtain any of the submodels simply pass the appropriate parameters as fixed.

2.2.6 The family GARCH model ('fGARCH')

The family GARCH model of Hentschel (1995) is another omnibus model which subsumes some of the most popular GARCH models. It is similar to the apARCH model, but more general since it allows the decomposition of the residuals in the conditional variance equation to be driven by different powers for z_t and σ_t and also allowing 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 large shocks.

$$\sigma_t^\lambda = \left(\omega + \sum_{j=1}^{m} \zeta_j v_{jt}\right) + \sum_{j=1}^{q} \alpha_j \sigma_{t-j}^\lambda (|z_{t-j} - \eta_{2j}| - \eta_{1j}(z_{t-j} - \eta_{2j}))^\delta + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^\lambda \tag{26}$$

which is a Box-Cox transformation for the conditional standard deviation whose shape is determined by λ, and the parameter δ transforms the absolute value function which it subject to rotations and shifts through the η_{1j} and η_{2j} parameters respectively. Various submodels arise from this model, and are passed to the `ugarchspec` 'variance.model' list via the submodel option,

- The simple GARCH model of Bollerslev (1986) when $\lambda = \delta = 2$ and $\eta_1 j = \eta_2 j = 0$ (submodel = 'GARCH').

- The Absolute Value GARCH (AVGARCH) model of Taylor (1986) and Schwert (1990) when $\lambda = \delta = 1$ and $|\eta_{1j}| \leq 1$ (submodel = 'AVGARCH').
- The GJR GARCH (GJRGARCH) model of Glosten et al. (1993) when $\lambda = \delta = 2$ and $\eta_{2j} = 0$ (submodel = 'GJRGARCH').
- The Threshold GARCH (TGARCH) model of Zakoian (1994) when $\lambda = \delta = 1$, $\eta_{2j} = 0$ and $|\eta_{1j}| \leq 1$ (submodel = 'TGARCH').
- The Nonlinear ARCH model of Higgins et al. (1992) when $\delta = \lambda$ and $\eta_1 j = \eta_2 j = 0$ (submodel = 'NGARCH').
- The Nonlinear Asymmetric GARCH model of Engle and Ng (1993) when $\delta = \lambda = 2$ and $\eta_1 j = 0$ (submodel = 'NAGARCH').
- The Asymmetric Power ARCH model of Ding et al. (1993) when $\delta = \lambda$, $\eta_{2j} = 0$ and $|\eta_{1j}| \leq 1$ (submodel = 'APARCH').
- The Exponential GARCH model of Nelson (1991) when $\delta = 1$, $\lambda = 0$ and $\eta_{2j} = 0$ (not implemented as a submodel of fGARCH).
- The Full fGARCH model of Hentschel (1995) when $\delta = \lambda$ (submodel = 'ALLGARCH').

The persistence of the model is given by,

$$\hat{P} = \sum_{j=1}^{p} \beta_j + \sum_{j=1}^{q} \alpha_j \kappa_j \tag{27}$$

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

$$\kappa_j = E(|z_{t-j} - \eta_{2j}| - \eta_{1j}(z_{t-j} - \eta_{2j}))^{\delta} = \int_{-\infty}^{\infty} (|z - \eta_{2j}| - \eta_{1j}(z - \eta_{2j}))^{\delta} f(z, 0, 1, ...)\, dz \tag{28}$$

If variance targeting is used, then ω is replaced by,

$$\bar{\sigma}^{\lambda}\left(1 - \hat{P}\right) - \sum_{j=1}^{m} \zeta_j \bar{v}_j \tag{29}$$

Finally, the unconditional variance of the model $\hat{\sigma}^2$ is,

$$\hat{\sigma}^2 = \left(\frac{\hat{\omega}}{1 - \hat{P}}\right)^{2/\lambda} \tag{30}$$

where $\hat{\omega}$ is the estimated value of the intercept from the GARCH model. The half-life follows from the persistence parameter and is calculated as in Section 2.2.1. The naming conventions for passing fixed or starting parameters for this model are:

- ARCH(q) parameters are 'alpha1', 'alpha2', ...,
- Asymmetry1(q) - rotation - parameters are 'eta11', 'eta12', ...,
- Asymmetry2(q) - shift - parameters are 'eta21', 'eta22', ...,
- Asymmetry Power parameter is 'delta',

- Conditional Sigma Power parameter is 'lambda',
- GARCH(p) parameters are 'beta1', 'beta2', ...,
- variance intercept parameter is 'omega'
- the external regressor parameters are 'vxreg1', 'vxreg2', ...,

2.2.7 The Component sGARCH model ('csGARCH')

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. Letting q_t represent the permanent component of the conditional variance, the component model can then be written as:

$$\begin{aligned}\sigma_t^2 &= q_t + \sum_{j=1}^{q}\alpha_j\left(\varepsilon_{t-j}^2 - q_{t-j}\right) + \sum_{j=1}^{p}\beta_j\left(\sigma_{t-j}^2 - q_{t-j}\right)\\ q_t &= \omega + \rho q_{t-1} + \phi\left(\varepsilon_{t-1}^2 - \sigma_{t-1}^2\right)\end{aligned} \tag{31}$$

where effectively the intercept of the GARCH model is now time-varying following first order autoregressive type dynamics. The difference between the conditional variance and its trend, $\sigma_{t-j}^2 - q_{t-j}$ is the transitory component of the conditional variance. The conditions for the non-negativity of the conditional variance are given in Lee and Engle (1999) and imposed during estimation by the stationarity option in the fit.control list of the `ugarchfit` method, and related to the stationarity conditions that the sum of the (α,β) coefficients be less than 1 and that $\rho < 1$ (effectively the persistence of the transitory and permanent components).
The multistep, $n > 1$ ahead forecast of the conditional variance proceeds as follows:

$$\begin{aligned}E_{t-1}\left(\sigma_{t+n}^2\right) &= E_{t-1}\left(q_{t+n}\right) + \sum_{j=1}^{q}\alpha_j\left(\varepsilon_{t+n-j}^2 - q_{t+n-j}\right) + \sum_{j=1}^{p}\beta_j\left(\sigma_{t+n-j}^2 - q_{t+n-j}\right)\\ E_{t-1}\left(\sigma_{t+n}^2\right) &= E_{t-1}\left(q_{t+n}\right) + \sum_{j=1}^{q}\alpha_j E_{t-1}\left[\varepsilon_{t+n-j}^2 - q_{t+n-j}\right] + \sum_{j=1}^{p}\beta_j E_{t-1}\left[\sigma_{t+n-j}^2 - q_{t+n-j}\right]\end{aligned} \tag{32}$$

However, $E_{t-1}\left[\varepsilon_{t+n-j}^2\right] = E_{t-1}\left[\sigma_{t+n-j}^2\right]$, therefore:

$$\begin{aligned}E_{t-1}\left(\sigma_{t+n}^2\right) &= E_{t-1}\left(q_{t+n}\right) + \sum_{j=1}^{q}\alpha_j E_{t-1}\left[\sigma_{t+n-j}^2 - q_{t+n-j}\right] + \sum_{j=1}^{p}\beta_j E_{t-1}\left[\sigma_{t+n-j}^2 - q_{t+n-j}\right]\\ E_{t-1}\left(\sigma_{t+n}^2\right) &= E_{t-1}\left(q_{t+n}\right) + \left[\sum_{j=1}^{\max(p,q)}\left(\alpha_j + \beta_j\right)\right]^n\left(\sigma_t^2 - q_t\right)\end{aligned} \tag{33}$$

The permanent component forecast can be represented as:

$$\begin{aligned} E_{t-1}\left[q_{t+n}\right] &= \omega + \rho E_{t-1}\left[q_{t+n-1}\right] + \phi E_{t-1}\left[\varepsilon^2_{t+n-j} - \sigma^2_{t+n-j}\right] && (34)\\ &= \omega + \rho E_{t-1}\left[q_{t+n-1}\right] && (35)\\ &= \omega + \rho\left[\omega + \rho E_{t-1}\left[q_{t+n-2}\right]\right] && (36)\\ &= \ldots && (37)\\ &= \left(1 + p + \cdots + \rho^{n-1}\right)\omega + \rho^n q_t && (38)\\ &= \frac{1-\rho^n}{1-\rho}\omega + \rho^n q_t && (39)\\ & && (40) \end{aligned}$$

As $n \to \infty$ the unconditional variance is:

$$E_{t-1}\left[\sigma^2_{t+n}\right] = E_{t-1}\left[q_{t+n}\right] = \frac{\omega}{1-\rho} \tag{41}$$

In the `rugarch` package, the parameters ρ and ϕ are represented by η_{11}('eta11') and η_{21}('eta21') respectively.

2.2.8 The Multiplicative Component sGARCH model ('mcsGARCH')

A key problem with using GARCH models for intraday data is the seasonality in the absolute returns observed at the beginning and end of the trading session. For regularly sampled time intervals (1-min, 5-min etc.), a number of models have tried to either 'de-seasonalize' the residuals and then fit the GARCH model by using for instance a Flexible Fourier method as in Andersen and Bollerslev (1997) or incorporate seasonality directly into the model as in the Periodic GARCH model of **?**. A rather simple and parsimonious approach to the de-seasonalization was recently presented in Engle and Sokalska (2012) (henceforth ES2012). Consider the continuously compounded return $r_{t,i}$, where t denotes the day and i the regularly spaced time interval at which the return was calculated. Under this model, the conditional variance is a multiplicative product of daily, diurnal and stochastic (intraday) components, so that the return process may be represented as:

$$\begin{aligned} r_{t,i} &= \mu_{t,i} + \varepsilon_{t,i}\\ \varepsilon_{t,i} &= \left(q_{t,i}\sigma_t s_i\right) z_{t,i} \end{aligned} \tag{42}$$

where $q_{t,i}$ is the stochastic intraday volatility, σ_t a daily exogenously determined forecast volatility, s_i the diurnal volatility in each regularly spaced interval i, $z_{t,i}$ the i.i.d (0,1) standardized innovation which conditional follows some appropriately chosen distribution. In ES2012, the forecast volatility σ_t is derived from a multifactor risk model externally, but it is just as possible to generate such forecasts from a daily GARCH model. The seasonal (diurnal) part of the process is defined as:

$$s_i = \frac{1}{T}\sum_{t=1}^{T}\left(\varepsilon^2_{t,i}/\sigma^2_t\right). \tag{43}$$

Dividing the residuals by the diurnal and daily volatility gives the normalized residuals ($\bar{\varepsilon}$):

$$\bar{\varepsilon}_{t,i} = \varepsilon_{t,i}/\left(\sigma_t s_i\right) \tag{44}$$

which may then be used to generate the stochastic component of volatility $q_{t,i}$ which assuming a simple vanilla GARCH model has the following motion dynamics:

$$q^2_{t,i} = \left(\omega + \sum_{j=1}^{m}\zeta_j v_{jt}\right) + \sum_{j=1}^{p}\alpha_j \bar{\varepsilon}^2_{t-j} + \sum_{j=1}^{q}\beta_j q^2_{t-j} \tag{45}$$

see Section 2.2.1 for details. In the `rugarch` package, unlike the paper of ES2012, the conditional mean and variance equations (and hence the diurnal component on the residuals from the conditional mean filtration) are estimated jointly. Furthermore, and unlike ES2012, it is possible to include ARMAX dynamics in the conditional mean, though because of the complexity of the model and its use of time indices, ARCH-m is not currently allowed, but may be included once the xts package is fully translated to Rcpp. Finally, as an additional point of departure from ES2012, the diurnal component calculation uses the median instead of the mean which was found to provide for a more robust alternative, particularly given the type and size of datasets typically used [changed in version 1.2-3]. As currently stands, the model has methods for estimation (ugarchfit), filtering (ugarchfilter), forecast from fit (ugarchforecast) but not from spec (secondary dispatch method), simulation from fit (ugarchsim) but not from spec (ugarchpath), rolling estimation (ugarchroll) but not the bootstrap (ugarchboot). Some of the plots, which depend on the xts package will not render nicely since plot.xts does not play well with intraday data. Some plots however such as VaRplot have been amended to properly display intraday data coming from an xts object, and more may be added as time allows. An extensive example of the model may be found on `http://www.unstarched.net`. The paper by ES2012 is currently freely available here: `http://jfec.oxfordjournals.org/content/10/1/54.full`.

2.2.9 The realized GARCH model ('realGARCH')

While the previous section discussed the modelling of intraday returns using a multiplicative component GARCH model, this section deals with the inclusion of realized measures of volatility in a GARCH modelling setup. The realized GARCH (realGARCH) model of Hansen et al. (2012) (henceforth HHS2012) provides for an excellent framework for the joint modelling of returns and realized measures of volatility. Unlike the naive augmentation of GARCH processes by a realized measures, the realGARCH model relates the observed realized measure to the latent volatility via a measurement equation, which also includes asymmetric reaction to shocks, making for a very flexible and rich representation. Formally, let:

$$\begin{aligned} y_t &= \mu_t + \sigma_t z_t,\ z_t \sim i.i.d\,(0,1) \\ \log \sigma_t^2 &= \omega + \sum_{i=1}^{q} \alpha_i \log r_{t-i} + \sum_{i=1}^{p} \beta_i \log \sigma_{t-i}^2 \\ \log r_t &= \xi + \delta \log \sigma_t^2 + \tau\,(z_t) + u_t,\ u_t \sim N\,(0,\lambda) \end{aligned} \tag{46}$$

where we have defined the dynamics for the returns (y_t), the log of the conditional variance (σ_t^2) and the log of the realized measure (r_t).[3] The asymmetric reaction to shocks comes via the $\tau\,(.)$ function which is based on the Hermite polynomials and truncated at the second level to give a simple quadratic form:

$$\tau\,(z_t) = \eta_1 z_t + \eta_2\left(z_t^2 - 1\right) \tag{47}$$

[3] In the original paper by HHS2012, the notation is slightly different as I have chosen to re-use some of the symbols/variables already in the rugarch specification. For completeness, the differences are noted:

- y_t (rugarch) = r_t (HHS2012)
- α (rugarch) = γ (HHS2012)
- σ_t^2 (rugarch) = h_t (HHS2012)
- r_t (rugarch) = x_t (HHS2012)
- δ (rugarch) = φ (HHS2012)
- η_1 (rugarch) = τ_1 (HHS2012)
- η_2 (rugarch) = τ_2 (HHS2012)
- λ (rugarch) = σ_u (HHS2012)

which has the very convenient property that $E\tau(z_t) = 0$. The function also forms the basis for the creation of a type of news impact curve $\nu(z)$, defined as:

$$\nu(z) = E[\log \sigma_t \,|z_{t-1} = z\,] - E[\log \sigma_t] = \delta\tau(z) \tag{48}$$

so that $100 \times \nu(z)$ is the percent change in volatility as a function of the standardized innovations.

A key feature of this model is that it preserves the ARMA structure which characterize many standard GARCH models and adapted here from Proposition 1 of the HHS2012 paper:

$$\begin{aligned}
\log \sigma_t^2 &= \mu_\sigma + \sum_{i=1}^{p\vee q} (\delta\alpha_i + \beta_i) \log \sigma_{t-1}^2 + \sum_{j=1}^{q} \alpha_j w_{t-j} \\
\log r_t &= \mu_r + \sum_{i=1}^{p\vee q} (\delta\alpha_i + \beta_i) \log r_{t-1} + w_t - \sum_{j=1}^{p\vee q} \beta_j w_{t-j} \\
\mu_\sigma &= \omega + \xi \sum_{i=1}^{q} \alpha_i, \mu_r = \delta\omega + \left(1 - \sum_{i=1}^{p} \beta_i\right) \xi
\end{aligned} \tag{49}$$

where $w_t = \tau(z_t) + u_t$, $\mu_\sigma = \omega + \xi \sum_{i=1}^{q} \alpha_i$ and $\mu_r = \delta\omega + \left(1 - \sum_{i=1}^{p} \beta_i\right) \xi$, and the convention $\beta_i = \alpha_j = 0$ for $i > p$ and $j < p$. It is therefore a simple matter to show that the persistence ($\hat{P}$) of the process is given by:

$$\hat{P} = \sum_{i=1}^{p} \beta_i + \delta \sum_{i=1}^{q} \alpha_i \tag{50}$$

while the unconditional (long-run) variance may be written as:

$$\hat{\sigma}^2 = \frac{\omega + \xi \sum_{i=1}^{q} \alpha_i}{1 - \hat{P}} \tag{51}$$

The joint likelihood of the model, made up of the returns and measurement equation, is given by:

$$\log L\left(\{y_t, r_t\}_{t=1}^{T}; \theta\right) = \sum_{t=1}^{T} \log f(y_t, r_t \,|\mathcal{F}_{t-1}) \tag{52}$$

While not standard for MLE, the independence of z_t and u_t means that we can factorize the joint density into:

$$\log f(y_t, r_t \,|\mathcal{F}_{t-1}) = \underbrace{\log(y_t \,|\mathcal{F}_{t-1})}_{\log l(y)} + \underbrace{(r_t \,|y_t, \mathcal{F}_{t-1})}_{\log l(r|y)} \tag{53}$$

which makes comparison with other GARCH models possible (using $\log l(y)$). Finally, multi-

period forecasts have a nice VARMA type representation $Y_t = AY_{t-1} + b + \varepsilon_t$, where:

$$Y_t = \begin{bmatrix} \log \sigma_t^2 \\ . \\ . \\ . \\ \log \sigma_{t-p+1}^2 \\ \log r_t \\ . \\ . \\ \log r_{t-q+1} \end{bmatrix}, A = \begin{pmatrix} (\beta_1, ..., \beta_p) & (\alpha_1, ..., \alpha_q) \\ (I_{p-1\times p-1}, 0_{p-1\times 1}) & 0_{p-1\times q} \\ \delta(\beta_1, ..., \beta_p) & \delta(\alpha_1, ..., \alpha_q) \\ 0_{q-1\times p} & (I_{q-1\times q-1}, 0_{q-1\times 1}) \end{pmatrix}, b = \begin{pmatrix} \omega \\ 0_{p-1\times 1} \\ \xi + \delta\omega \\ 0_{q-1\times 1} \end{pmatrix}$$
$$\varepsilon_t = \begin{pmatrix} 0_{p\times 1} \\ \tau(z_t) + u_t \\ 0_{q\times 1} \end{pmatrix} \tag{54}$$

so that $Y_{t+k} = A^k Y_t + \sum_{j=0}^{k-1} A^j (b + \varepsilon_{t+k-j})$, and it is understood that the superscripts denote matrix power, with $[.]^0$ the identity matrix.[4]

In the `rugarch` package, all the methods, from estimation, to filtering, forecasting and simulation have been included with key parts of the code written in C for speed (as elsewhere). For the forecast routine, some additional arguments are made available as a result of the generation of the conditional variance density (rather than point forecasts, although these are returned based on the average of the simulated density values). Consult the documentation and online examples for more details.

2.2.10 The fractionally integrated GARCH model ('fiGARCH')

Motivated by the developments in long memory processes, and in particular the ARFIMA type models (see section 2.1), Baillie et al. (1996) proposed the fractionally integrated generalized autoregressive conditional heteroscedasticity, or FIGARCH, model to capture long memory (in essence hyperbolic memory). Unlike the standard GARCH where shocks decay at an exponential rate, or the integrated GARCH model where shocks persist forever, in the FIGARCH model shocks decay at a slower hyperbolic rate. Consider the standard GARCH equation:

$$\sigma_t^2 = \omega + \alpha(L)\varepsilon_t^2 + \beta(L)\sigma_t^2 \tag{55}$$

where L is the lag operator, such that $a(L) = \sum_{i=1}^{q} \alpha_i L^i$ and $\beta(L) = \sum_{j=1}^{p} \beta_j L^j$. Re-arranging to give the ARMA in squares representation:

$$[1 - \alpha(L) - \beta(L)]\varepsilon_t^2 = \omega + [1 - \beta(L)]v_t \tag{56}$$

where $v_t = \varepsilon_t^2 - \sigma_t^2$, and condensing the left hand side:

$$(1 - L)\phi(L)\varepsilon_t^2 = \omega + [1 - \beta(L)]v_t \tag{57}$$

with $\phi(L) = \sum_{i=1}^{m-1} \phi_i L^i$ and $m = \max\{p, q\}$. In the fractionally integrated model, $(1 - L)$ is replaced by :

$$(1 - L)^d = \sum_{k=0}^{\infty} \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)} L^k = 1 - \sum_{k=1}^{\infty} \pi_k L^k \tag{58}$$

[4]The original paper of HHS2012 has some small typos in the notation for the horizon k which is at times represented by the symbol h (which is reserved in the paper for the conditional variance notation).

where $\pi_i = \prod_{1\leq k\leq i} \frac{k-1-d}{k}$. The expansion[5] is usually truncated to some large number, such as 1000. Rearranging again, we obtain the following representation of the FIGARCH model:

$$\begin{aligned}
\sigma_t^2 &= \omega[1-\beta(L)]^{-1} + \left\{1 - [1-\beta(L)^{-1}\phi(L)(1-L)^d\right\}\varepsilon_t^2 \\
&= \omega^* + \lambda(L)\varepsilon_t^2 \\
&= \omega^* + \sum_{j=1}^{\infty}\lambda_i L^i \varepsilon_t^2
\end{aligned} \tag{59}$$

where $\lambda_1 = \phi_1 - \beta_1 + d$ and $\lambda_k = \beta_1\lambda_{k-1} + \left(\frac{k-1-d}{k} - \phi_1\right)\pi_{d,k-1}$. For the FIGARCH(1,d,1) model, sufficient conditions to ensure positivity of the conditional variance are $\omega > 0$,$\beta_1 - d \leq \phi_1 \leq \left(\frac{2-d}{2}\right)$, and $d\left(\phi_1 - \frac{(1-d)}{2}\right) \leq \beta_1(\phi_1 - \beta_1 + d)$. In keeping with the formulation used for GARCH models in this package, we re-write equation 59 as follows, setting $\phi(L) \equiv (1-\alpha(L))$:

$$\begin{aligned}
\phi(L)(1-L)^d\varepsilon_t^2 &= \omega + (1-\beta(L))\left(\varepsilon_t^2 - \sigma_t^2\right) \\
\phi(L)(1-L)^d\varepsilon_t^2 &= \omega - \sigma_t^2 + \varepsilon_t^2 + \beta(L)\sigma_t^2 - \beta(L)\varepsilon_t^2 \\
\sigma_t^2 &= \omega + \varepsilon_t^2 + \beta(L)\sigma_t^2 - \beta(L)\varepsilon_t^2 - \phi(L)(1-L)^d\varepsilon_t^2 \\
\sigma_t^2 &= \omega + \left\{1 - \beta(L) - \phi(L)(1-L)^d\right\}\varepsilon_t^2 + \beta(L)\sigma_t^2 \\
\sigma_t^2 &= \omega + \left\{1 - \beta(L) - (1-\alpha(L))(1-L)^d\right\}\varepsilon_t^2 + \beta(L)\sigma_t^2
\end{aligned} \tag{60}$$

Truncating the expansion to 1000 lags and setting $(1-L)^d\varepsilon_t^2 = \varepsilon_t^2 + \left(\sum_{k=1}^{1000}\pi_k L^k\right)\varepsilon_t^2 = \varepsilon_t^2 + \bar{\varepsilon}_t^2$, we can rewrite the equation as:

$$\begin{aligned}
\sigma_t^2 &= \omega + \left\{\varepsilon_t^2 - \beta(L)\varepsilon_t^2 - (1-L)^d\varepsilon_t^2 + \alpha(L)(1-L)^d\varepsilon_t^2\right\} + \beta(L)\sigma_t^2 \\
\sigma_t^2 &= \omega + \left\{\varepsilon_t^2 - \beta(L)\varepsilon_t^2 - \left(\varepsilon_t^2 + \bar{\varepsilon}_t^2\right) + \alpha(L)\left(\varepsilon_t^2 + \bar{\varepsilon}_t^2\right)\right\} + \beta(L)\sigma_t^2 \\
\sigma_t^2 &= \omega + \varepsilon_t^2 - \beta(L)\varepsilon_t^2 - \left(\varepsilon_t^2 + \bar{\varepsilon}_t^2\right) + \alpha(L)\left(\varepsilon_t^2 + \bar{\varepsilon}_t^2\right) + \beta(L)\sigma_t^2 \\
\sigma_t^2 &= \omega - \bar{\varepsilon}_t^2 - \beta(L)\varepsilon_t^2 + \alpha(L)\left(\varepsilon_t^2 + \bar{\varepsilon}_t^2\right) + \beta(L)\sigma_t^2 \\
\sigma_t^2 &= \left(\omega - \bar{\varepsilon}_t^2\right) - \sum_{j=1}^{p}\beta_j\varepsilon_{t-j}^2 + \sum_{j=1}^{q}\alpha_j\varepsilon_{t-j}^2 + \sum_{j=1}^{q}\alpha_j\bar{\varepsilon}_{t-j}^2 + \sum_{j=1}^{p}\beta_j\sigma_{t-j}^2 \\
\sigma_t^2 &= \left(\omega - \bar{\varepsilon}_t^2\right) + \sum_{j=1}^{q}\alpha_j\left(\varepsilon_{t-j}^2 + \bar{\varepsilon}_{t-j}^2\right) + \sum_{j=1}^{p}\beta_j\left(\sigma_{t-j}^2 - \varepsilon_{t-j}^2\right)
\end{aligned} \tag{61}$$

Contrary to the case of the ARFIMA model, the degree of persistence in the FIGARCH model operates in the oppposite direction, so that as the fractional differencing parameter d gets closer to one, the memory of the FIGARCH process increases, a direct result of the parameter acting on the squared errors rather than the conditional variance. When $d = 0$ the FIGARCH collapses to the vanilla GARCH model and when $d = 1$ to the integrated GARCH model. The question of the stationarity of a FIGARCH(q,d,p) model is open and there is no general proof of this at present. As such, the stationarity argument in the estimation function is used interchangeably for positivity conditions. Baillie et al. (1996) provided a set of sufficient conditions for the FIGARCH(1,d,1) case which may be restrictive in practice, which is why Conrad and Haag (2006) provide a more general set of sufficient conditions for the FIGARCH(q,d,p). Equations (7)-(9) and Corollary 1 of their paper provide the conditions for the positivity in the case of the FIGARCH(1,d,1) case which the `rugarch` package implements.[6] Therefore, while it is possible to estimate any order desired, only conditions for the (1,d,1) are checked and imposed during estimation when the stationarity[7] flag is set to TRUE.

[5] This is the hypergeometric function expansion.

[6] At present, only the (1,d,1) case is allowed.

[7] Which is used here to denote a positivity constraint

Numerous alternatives and extensions have been proposed in the literature since the FI-GARCH model was published. The model of Karanasos et al. (2004) models the squared residuals as deviations from ω so that it specifies a covariance stationary process (although the question of strict stationary still remains open). Augmenting the EGARCH model with the fractional operator appears to provide for a natural way to deal with the positivity issue since the process is always strictly positive (see Bollerslev and Mikkelsen (1996)). This may be included in a future version.

2.3 Conditional Distributions

The `rugarch` package supports a range of univariate distributions including the Normal ('norm'), Generalized Error ('ged'), Student ('std') and their skew variants ('snorm', 'sged' and 'sstd') based on the transformations described in Fernandez and Steel (1998) and Ferreira and Steel (2006).[8] Additionally, the Generalized Hyperbolic ('ghyp'), Normal Inverse Gaussian ('nig') and GH Skew-Student ('ghst')[9] distributions are also implemented as is Johnson's reparametrized SU ('jsu') distribution[10] The choice of distribution is entered via the 'distribution.model' option of the `ugarchspec` method. The package also implements a set of functions to work with the parameters of these distributions. These are:

- ddist(distribution = "norm", y, mu = 0, sigma = 1, lambda = -0.5, skew = 1, shape = 5). The density (d*) function.
- pdist(distribution = "norm", q, mu = 0, sigma = 1, lambda = -0.5, skew = 1, shape = 5). The distribution (p*) function.
- qdist(distribution = "norm", p, mu = 0, sigma = 1, lambda = -0.5, skew = 1, shape = 5). The quantile (q*) function.
- rdist(distribution = "norm", n, mu = 0, sigma = 1, lambda = -0.5, skew = 1, shape = 5). The sampling (q*) function.
- fitdist(distribution = "norm", x, control = list()). A function for fitting data using any of the included distributions.
- dskewness(distribution = "norm", skew = 1, shape = 5, lambda = -0.5). The distribution skewness (analytical where possible else by quadrature integration).
- dkurtosis(distribution = "norm", skew = 1, shape = 5, lambda = -0.5). The distribution excess kurtosis (analytical where it exists else by quadrature integration).

This section provides a dry but comprehensive exposition of the required standardization of these distributions for use in GARCH modelling.

The conditional distribution in GARCH processes should be self-decomposable which is a key requirement for any autoregressive type process, while possessing the linear transformation property is required to center $(x_t - \mu_t)$ and scale (ε_t/σ_t) the innovations, after which the modelling is carried out directly using the zero-mean, unit variance, distribution of the standardized variable z_t which is a scaled version of the same conditional distribution of x_t, as described in Equations 6, 7 and 8.

[8] These were originally taken from the fBasics package but have been adapted and re-written in C for the likelihood estimation.

[9] Since version 1.0-8.

[10] From the gamlss package.

2.3.1 The Normal Distribution

The Normal Distribution is a spherical distribution described completely by it first two moments, the mean and variance. Formally, the random variable x is said to be normally distributed with mean μ and variance σ^2 (both of which may be time varying), with density given by,

$$f(x) = \frac{e^{\frac{-0.5(x-\mu)^2}{\sigma^2}}}{\sigma\sqrt{2\pi}}. \tag{62}$$

Following a mean filtration or whitening process, the residuals ε, standardized by σ yield the standard normal density given by,

$$f\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{\sigma}f(z) = \frac{1}{\sigma}\left(\frac{e^{-0.5z^2}}{\sqrt{2\pi}}\right). \tag{63}$$

To obtain the conditional likelihood of the GARCH process at each point in time (LL_t), the conditional standard deviation σ_t from the GARCH motion dynamics, acts as a scaling factor on the density, so that:

$$LL_t(z_t;\sigma_t) = \frac{1}{\sigma_t}f(z_t) \tag{64}$$

which illustrates the importance of the scaling property. Finally, the normal distribution has zero skewness and zero excess kurtosis.

2.3.2 The Student Distribution

The GARCH-Student model was first used described in Bollerslev (1987) as an alternative to the Normal distribution for fitting the standardized innovations. It is described completely by a shape parameter ν, but for standardization we proceed by using its 3 parameter representation as follows:

$$f(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\beta\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{(x-\alpha)^2}{\beta\nu}\right)^{-\left(\frac{\nu+1}{2}\right)} \tag{65}$$

where α, β, and ν are the location, scale[11] and shape parameters respectively, and Γ is the Gamma function. Similar to the GED distribution described later, this is a unimodal and symmetric distribution where the location parameter α is the mean (and mode) of the distribution while the variance is:

$$Var(x) = \frac{\beta\nu}{(\nu-2)}. \tag{66}$$

For the purposes of standardization we require that:

$$\begin{aligned} Var(x) &= \frac{\beta\nu}{(\nu-2)} = 1 \\ \therefore \beta &= \frac{\nu-2}{\nu} \end{aligned} \tag{67}$$

Substituting $\frac{(\nu-2)}{\nu}$ into 65 we obtain the standardized Student's distribution:

$$f\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{\sigma}f(z) = \frac{1}{\sigma}\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{(\nu-2)\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{z^2}{(\nu-2)}\right)^{-\left(\frac{\nu+1}{2}\right)}. \tag{68}$$

[11] In some representations, mostly Bayesian, this is represented in its inverse form to denote the precision.

In terms of R's standard implementation of the Student density ('dt'), and including a scaling by the standard deviation, this can be represented as:

$$\frac{dt\left(\frac{\varepsilon_t}{\sigma\sqrt{(v-2)/\nu}},\nu\right)}{\sigma\sqrt{(v-2)/\nu}} \tag{69}$$

The Student distribution has zero skewness and excess kurtosis equal to $6/(\nu-4)$ for $\nu > 4$.

2.3.3 The Generalized Error Distribution

The Generalized Error Distribution (GED) is a 3 parameter distribution belonging to the exponential family with conditional density given by,

$$f(x) = \frac{\kappa e^{-0.5\left|\frac{x-\alpha}{\beta}\right|^{\kappa}}}{2^{1+\kappa^{-1}}\beta\Gamma\left(\kappa^{-1}\right)} \tag{70}$$

with α, β and κ representing the location, scale and shape parameters. Since the distribution is symmetric and unimodal the location parameter is also the mode, median and mean of the distribution (i.e. μ). By symmetry, all odd moments beyond the mean are zero. The variance and kurtosis are given by,

$$\begin{aligned} Var(x) &= \beta^2 2^{2/\kappa}\frac{\Gamma\left(3\kappa^{-1}\right)}{\Gamma\left(\kappa^{-1}\right)} \\ Ku(x) &= \frac{\Gamma\left(5\kappa^{-1}\right)\Gamma\left(\kappa^{-1}\right)}{\Gamma\left(3\kappa^{-1}\right)\Gamma\left(3\kappa^{-1}\right)} \end{aligned} \tag{71}$$

As κ decreases the density gets flatter and flatter while in the limit as $\kappa \to \infty$, the distribution tends towards the uniform. Special cases are the Normal when $\kappa = 2$, the Laplace when $\kappa = 1$. Standardization is simple and involves rescaling the density to have unit standard deviation:

$$\begin{aligned} Var(x) &= \beta^2 2^{2/\kappa}\frac{\Gamma\left(3\kappa^{-1}\right)}{\Gamma\left(\kappa^{-1}\right)} = 1 \\ \therefore \beta &= \sqrt{2^{-2/\kappa}\frac{\Gamma\left(\kappa^{-1}\right)}{\Gamma\left(3\kappa^{-1}\right)}} \end{aligned} \tag{72}$$

Finally, substituting into the scaled density of z:

$$f\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{\sigma}f(z) = \frac{1}{\sigma}\frac{\kappa e^{-0.5\left|\sqrt{2^{-2/\kappa}\frac{\Gamma\left(\kappa^{-1}\right)}{\Gamma\left(3\kappa^{-1}\right)}}z\right|^{\kappa}}}{\sqrt{2^{-2/\kappa}\frac{\Gamma\left(\kappa^{-1}\right)}{\Gamma\left(3\kappa^{-1}\right)}}2^{1+\kappa^{-1}}\Gamma\left(\kappa^{-1}\right)} \tag{73}$$

2.3.4 Skewed Distributions by Inverse Scale Factors

Fernandez and Steel (1998) proposed introducing skewness into unimodal and symmetric distributions by introducing inverse scale factors in the positive and negative real half lines. Given a skew parameter, ξ[12], the density of a random variable z can be represented as:

$$f(z|\xi) = \frac{2}{\xi+\xi^{-1}}\left[f(\xi z)H(-z) + f\left(\xi^{-1}z\right)H(z)\right] \tag{74}$$

[12]When $\xi = 1$, the distribution is symmetric.

where $\xi \in \mathbb{R}^+$ and $H(.)$ is the Heaviside function. The absolute moments, required for deriving the central moments, are generated from the following function:

$$M_r = 2\int_0^\infty z^r f(z)\, dz. \tag{75}$$

The mean and variance are then defined as:

$$\begin{aligned} E(z) &= M_1\left(\xi - \xi^{-1}\right) \\ Var(z) &= \left(M_2 - M_1^2\right)\left(\xi^2 + \xi^{-2}\right) + 2M_1^2 - M_2 \end{aligned} \tag{76}$$

The Normal, Student and GED distributions have skew variants which have been standardized to zero mean, unit variance by making use of the moment conditions given above.

2.3.5 The Generalized Hyperbolic Distribution and Sub-Families

In distributions where the expected moments are functions of all the parameters, it is not immediately obvious how to perform such a transformation. In the case of the GHYP distribution, because of the existence of location and scale invariant parameterizations and the possibility of expressing the variance in terms of one of those parametrization, namely the (ζ, ρ), the task of standardizing and estimating the density can be broken down to one of estimating those 2 parameters, representing a combination of shape and skewness, followed by a series of transformation steps to demean, scale and then translate the parameters into the $(\alpha, \beta, \delta, \mu)$ parametrization for which standard formulae exist for the likelihood function. The (ξ, χ) parametrization, which is a simple transformation of the (ζ, ρ), could also be used in the first step and then transformed into the latter before proceeding further. The only difference is the kind of 'immediate' inference one can make from the different parameterizations, each providing a different direct insight into the kind of dynamics produced and their place in the overall GHYP family particularly with regards to the limit cases.
The `rugarch` package performs estimation using the (ζ, ρ) parametrization[13], after which a series of steps transform those parameters into the $(\alpha, \beta, \delta, \mu)$ while at the same time including the necessary recursive substitution of parameters in order to standardize the resulting distribution.

Proof 1 *The Standardized Generalized Hyperbolic Distribution. Let ε_t be a r.v. with mean* (0) *and variance* (σ^2) *distributed as $GHYP(\zeta, \rho)$, and let z be a scaled version of the r.v. ε with variance* (1) *and also distributed as $GHYP(\zeta, \rho)$.*[14] *The density $f(.)$ of z can be expressed as*

$$f(\frac{\varepsilon_t}{\sigma}; \zeta, \rho) = \frac{1}{\sigma} f_t(z; \zeta, \rho) = \frac{1}{\sigma} f_t(z; \tilde{\alpha}, \tilde{\beta}, \tilde{\delta}, \tilde{\mu}), \tag{77}$$

where we make use of the $(\alpha, \beta, \delta, \mu)$ parametrization since we can only naturally express the density in that parametrization. The steps to transforming from the (ζ, ρ) to the $(\alpha, \beta, \delta, \mu)$ parametrization, while at the same time standardizing for zero mean and unit variance are given henceforth.
Let

$$\zeta = \delta\sqrt{\alpha^2 - \beta^2} \tag{78}$$

$$\rho = \frac{\beta}{\alpha}, \tag{79}$$

[13] Credit is due to Diethelm Wurtz for his original implementation in the fBasics package of the transformation and standardization function.
[14] The parameters ζ and ρ do not change as a result of being location and scale invariant

which after some substitution may be also written in terms of α and β as,

$$\alpha = \frac{\zeta}{\delta\sqrt{(1-\rho^2)}}, \tag{80}$$

$$\beta = \alpha\rho. \tag{81}$$

For standardization we require that,

$$\begin{aligned} E(X) &= \mu + \frac{\beta\delta}{\sqrt{\alpha^2-\beta^2}}\frac{K_{\lambda+1}(\zeta)}{K_\lambda(\zeta)} = \mu + \frac{\beta\delta^2}{\zeta}\frac{K_{\lambda+1}(\zeta)}{K_\lambda(\zeta)} = 0 \\ \therefore \mu &= -\frac{\beta\delta^2}{\zeta}\frac{K_{\lambda+1}(\zeta)}{K_\lambda(\zeta)} \end{aligned} \tag{82}$$

$$\begin{aligned} Var(X) &= \delta^2\left(\frac{K_{\lambda+1}(\zeta)}{\zeta K_\lambda(\zeta)} + \frac{\beta^2}{\alpha^2-\beta^2}\left(\frac{K_{\lambda+2}(\zeta)}{K_\lambda(\zeta)} - \left(\frac{K_{\lambda+1}(\zeta)}{K_\lambda(\zeta)}\right)^2\right)\right) = 1 \\ \therefore \delta &= \left(\frac{K_{\lambda+1}(\zeta)}{\zeta K_\lambda(\zeta)} + \frac{\beta^2}{\alpha^2-\beta^2}\left(\frac{K_{\lambda+2}(\zeta)}{K_\lambda(\zeta)} - \left(\frac{K_{\lambda+1}(\zeta)}{K_\lambda(\zeta)}\right)^2\right)\right)^{-0.5} \end{aligned} \tag{83}$$

Since we can express, $\beta^2/\left(\alpha^2-\beta^2\right)$ as,

$$\frac{\beta^2}{\alpha^2-\beta^2} = \frac{\alpha^2\rho^2}{a^2-\alpha^2\rho^2} = \frac{\alpha^2\rho^2}{a^2(1-\rho^2)} = \frac{\rho^2}{(1-\rho^2)}, \tag{84}$$

then we can re-write the formula for δ in terms of the estimated parameters $\hat{\zeta}$ and $\hat{\rho}$ as,

$$\delta = \left(\frac{K_{\lambda+1}\left(\hat{\zeta}\right)}{\hat{\zeta}K_\lambda\left(\hat{\zeta}\right)} + \frac{\hat{\rho}^2}{(1-\hat{\rho}^2)}\left(\frac{K_{\lambda+2}\left(\hat{\zeta}\right)}{K_\lambda\left(\hat{\zeta}\right)} - \left(\frac{K_{\lambda+1}\left(\hat{\zeta}\right)}{K_\lambda\left(\hat{\zeta}\right)}\right)^2\right)\right)^{-0.5} \tag{85}$$

Transforming into the $(\tilde{\alpha}, \tilde{\beta}, \tilde{\delta}, \tilde{\mu})$ parametrization proceeds by first substituting 85 into 80 and simplifying,

$$\begin{aligned} \tilde{\alpha} &= \frac{\hat{\zeta}\left(\frac{K_{\lambda+1}(\hat{\zeta})}{\hat{\zeta}K_\lambda(\hat{\zeta})} + \frac{\hat{\rho}^2\left(\frac{K_{\lambda+2}(\hat{\zeta})}{K_\lambda(\hat{\zeta})} - \frac{(K_{\lambda+1}(\hat{\zeta}))^2}{(K_\lambda(\hat{\zeta}))^2}\right)}{(1-\hat{\rho}^2)}\right)^{0.5}}{\sqrt{(1-\hat{\rho}^2)}}, \\ &= \frac{\left(\frac{\hat{\zeta}K_{\lambda+1}(\hat{\zeta})}{K_\lambda(\hat{\zeta})} + \frac{\hat{\zeta}^2\hat{\rho}^2\left(\frac{K_{\lambda+2}(\hat{\zeta})}{K_\lambda(\hat{\zeta})} - \frac{(K_{\lambda+1}(\hat{\zeta}))^2}{(K_\lambda(\hat{\zeta}))^2}\right)}{(1-\hat{\rho}^2)}\right)^{0.5}}{\sqrt{(1-\hat{\rho}^2)}}, \\ &= \left(\frac{\frac{\hat{\zeta}K_{\lambda+1}(\hat{\zeta})}{K_\lambda(\hat{\zeta})}}{(1-\hat{\rho}^2)} + \frac{\hat{\zeta}^2\hat{\rho}^2\left(\frac{K_{\lambda+2}(\hat{\zeta})}{K_{\lambda+1}(\hat{\zeta})}\frac{K_{\lambda+1}(\hat{\zeta})}{K_\lambda(\hat{\zeta})} - \frac{(K_{\lambda+1}(\hat{\zeta}))^2}{(K_\lambda(\hat{\zeta}))^2}\right)}{(1-\hat{\rho}^2)^2}\right)^{0.5}, \\ &= \left(\frac{\hat{\zeta}K_{\lambda+1}(\hat{\zeta})}{K_\lambda(\hat{\zeta})}\right)^{0.5}\left(\frac{1}{(1-\hat{\rho}^2)}\left(1 + \frac{\hat{\zeta}\hat{\rho}^2\left(\frac{K_{\lambda+2}(\hat{\zeta})}{K_{\lambda+1}(\hat{\zeta})} - \frac{K_{\lambda+1}(\hat{\zeta})}{K_\lambda(\hat{\zeta})}\right)}{(1-\hat{\rho}^2)}\right)\right)^{0.5}. \end{aligned} \tag{86}$$

Finally, the rest of the parameters are derived recursively from $\tilde{\alpha}$ and the previous results,

$$\tilde{\beta} = \tilde{\alpha}\hat{\rho}, \tag{87}$$

$$\tilde{\delta} = \frac{\hat{\zeta}}{\tilde{\alpha}\sqrt{1-\hat{\rho}^2}}, \tag{88}$$

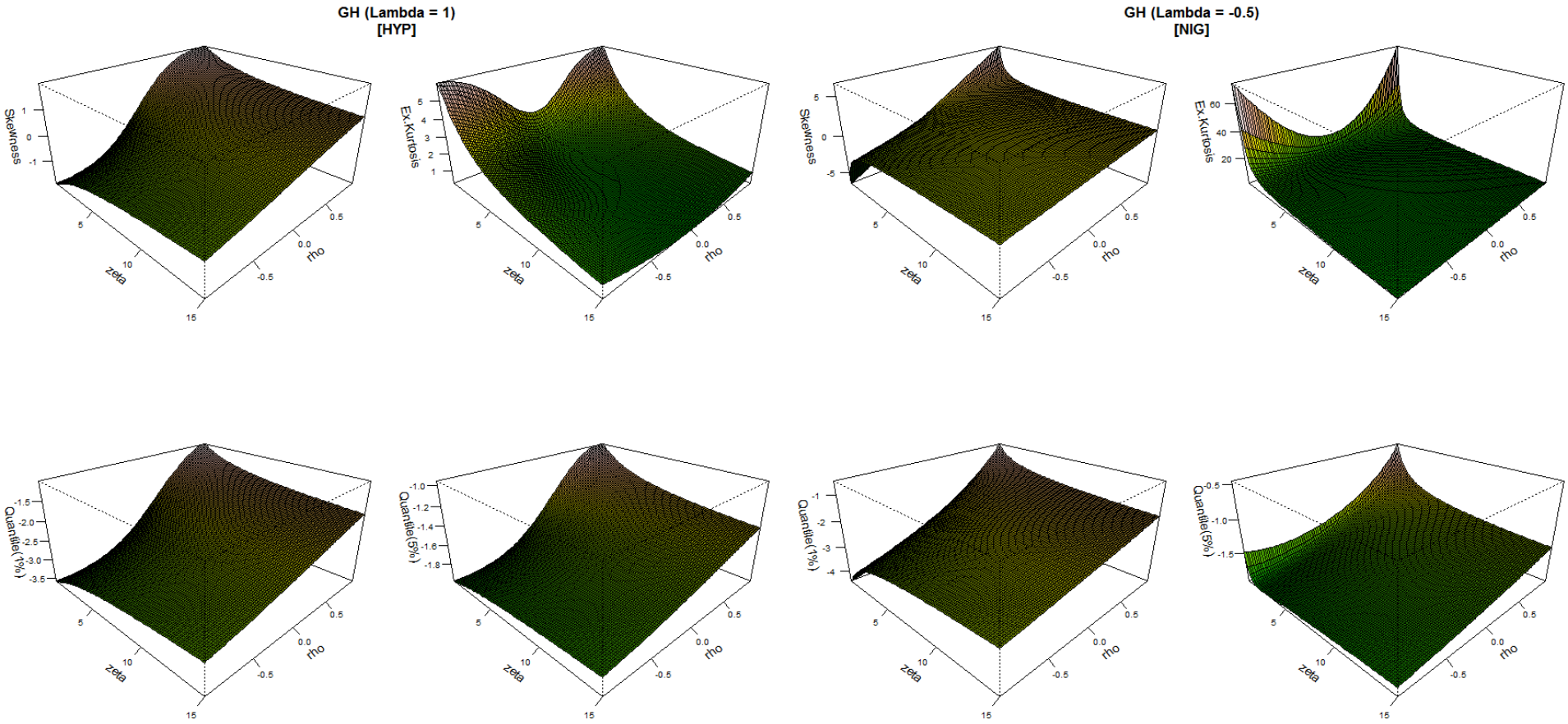
$$\tilde{\mu} = \frac{-\tilde{\beta}\tilde{\delta}^2 K_{\lambda+1}\left(\hat{\zeta}\right)}{\hat{\zeta}K_{\lambda}\left(\hat{\zeta}\right)}. \tag{89}$$

For the use of the (ξ, χ) parametrization in estimation, the additional preliminary steps of converting to the (ζ, ρ) are,

$$\zeta = \frac{1}{\hat{\xi}^2} - 1, \tag{90}$$

$$\rho = \frac{\hat{\chi}}{\hat{\xi}}. \tag{91}$$

Particular care should be exercised when choosing the GH distribution in GARCH models since allowing the GIG λ parameter to vary is quite troublesome in practice and may lead to identification problems since different combinations of the 2 shape (λ, ζ) and 1 skew (ρ) parameters may lead to the same or close likelihood. In addition, large sections of the likelihood surface for some combinations of the distribution parameters is quite flat. Figure 1 shows the skewness, kurtosis and 2 quantiles surfaces for different combinations of the (ρ, ζ) parameters for two popular choices of λ.



(a) $\lambda = 1$(HYP)

(b) $\lambda = -0.5$(NIG)

Figure 1: GH Distribution Skewness, Kurtosis and Quantile Surfaces

2.3.6 The Generalized Hyperbolic Skew Student Distribution

The GH Skew-Student distribution was popularized by Aas and Haff (2006) because of its uniqueness in the GH family in having one tail with polynomial and one with exponential behavior. This distribution is a limiting case of the GH when $\alpha \to |\beta|$ and $\lambda = -\nu/2$, where ν is the shape parameter of the Student distribution. The domain of variation of the parameters is $\beta \in \mathbb{R}$ and $\nu > 0$, but for the variance to be finite $\nu > 4$, while for the existence of skewness and kurtosis, $\nu > 6$ and $\nu > 8$ respectively. The density of the random variable x is then given by:

$$f(x) = \frac{2^{(1-\nu)/2}\delta^{\nu}|\beta|^{(\nu+1)/2}K_{(\nu+1)/2}\left(\sqrt{\beta^2\left(\delta^2+(x-\mu)^2\right)}\right)\exp\left(\beta\left(x-\mu\right)\right)}{\Gamma\left(\nu/2\right)\sqrt{\pi}\left(\sqrt{\delta^2+(x-\mu)^2}\right)^{(\nu+1)/2}} \tag{92}$$

To standardize the distribution to have zero mean and unit variance, I make use of the first two moment conditions for the distribution which are:

$$\begin{aligned} E(x) &= \mu + \frac{\beta\delta^2}{\nu-2} \\ Var(x) &= \frac{2\beta^2\delta^4}{(\nu-2)^2(\nu-4)} + \frac{\delta^2}{\nu-2} \end{aligned} \tag{93}$$

We require that $Var(x) = 1$, thus:

$$\delta = \left(\frac{2\bar{\beta}^2}{(\nu-2)^2(\nu-4)} + \frac{1}{\nu-2}\right)^{-1/2} \tag{94}$$

where I have made use of the 4^{th} parametrization of the GH distribution given in Prause (1999) where $\hat{\beta} = \beta\delta$. The location parameter is then rescaled by substituting into the first moment formula δ so that it has zero mean:

$$\bar{\mu} = -\frac{\beta\delta^2}{\nu-2} \tag{95}$$

Therefore, we model the GH Skew-Student using the location-scale invariant parametrization $(\bar{\beta}, \nu)$ and then translate the parameters into the usual GH distribution's $(\alpha, \beta, \delta, \mu)$, setting $\alpha = abs(\beta) + 1e-12$. As of version 1.2-8, the quantile function (via **qdist**) is calculated using the SkewHyperbolic package of Scott and Grimson using the spline method (for speed), as is the distribution function (via **pdist**).

2.3.7 Johnson's Reparametrized SU Distribution

The reparameterized Johnson SU distribution, discussed in Rigby and Stasinopoulos (2005), is a four parameter distribution denoted by $JSU(\mu, \sigma, \nu, \tau)$, with mean μ and standard deviation σ for all values of the skew and shape parameters ν and τ respectively. The implementation is taken from the GAMLSS package of Stasinopoulos et al. (2009) and the reader is referred there for further details.

3 Fitting

Once a `uGARCHspec` has been defined, the `ugarchfit` method takes the following arguments:

```
> args(ugarchfit)
```

```
function (spec, data, out.sample = 0, solver = "solnp", solver.control = list(),
    fit.control = list(stationarity = 1, fixed.se = 0, scale = 0,
        rec.init = "all"), ...)
```

The out.sample option controls how many data points from the end to keep for out of sample forecasting, while the solver.control and fit.control provide additional options to the fitting routine. Importantly, the *stationarity* option controls whether to impose a stationarity constraint during estimation, which is usually closely tied to the persistence of the process. The *fixed.se* controls whether, for those values which are fixed, numerical standard errors should be calculated. The *scale* option controls whether the data should be scaled prior to estimation by its standard deviation (scaling sometimes facilitates the estimation process). The option *rec.init*, introduced in version 1.0-14 allows to set the type of method for the conditional recursion initialization, with default value 'all' indicating that all the data is used to calculate the mean of the squared residuals from the conditional mean filtration. To use the first 'n' points for the calculation, a positive integer greater than or equal to one (and less than the total estimation datapoints) can instead be provided. If instead a positive numeric value less than 1 is provided, this is taken as the weighting in an exponential smoothing backcast method for calculating the initial recursion value.
Currently, 5 solvers [15] are supported, with the main one being the augmented Lagrange solver solnp of Ye (1997) implemented in R by Ghalanos and Theussl (2011). The main functionality, namely the GARCH dynamics and conditional likelihood calculations are done in C for speed. For reference, there is a benchmark routine called `ugarchbench` which provides a comparison of `rugarch` against 2 published GARCH models with analytic standard errors, and a small scale comparison with a commercial GARCH implementation. The fitted object is of class `uGARCHfit` which can be passed to a variety of other methods such as show (summary), plot, ugarchsim, ugarchforecast etc. The following example illustrates its use, but the interested reader should consult the documentation on the methods available for the returned class.

```
> spec = ugarchspec()
> data(sp500ret)
> fit = ugarchfit(spec = spec, data = sp500ret)
> show(fit)

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model        : sGARCH(1,1)
Mean Model         : ARFIMA(1,0,1)
Distribution       : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000522    0.000087   5.9873  0.00000
ar1     0.870609    0.071909  12.1070  0.00000
```

[15] Since version $1.0-8$ the 'nlopt' solver of Johnson (interfaced to R by Jelmer Ypma in the 'nloptr' package) has been added, greatly expanding the range of possibilities available via its numerous subsolver options - see documentation.

```
ma1    -0.897805    0.064324 -13.9576  0.00000
omega   0.000001    0.000001   1.3912  0.16418
alpha1  0.087714    0.013705   6.4001  0.00000
beta1   0.904955    0.013750  65.8136  0.00000

Robust Standard Errors:
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.000522    0.000130   4.020100 0.000058
ar1     0.870609    0.087878   9.907060 0.000000
ma1    -0.897805    0.079989 -11.224134 0.000000
omega   0.000001    0.000014   0.093412 0.925576
alpha1  0.087714    0.186497   0.470322 0.638125
beta1   0.904955    0.191986   4.713649 0.000002

LogLikelihood : 17902.41

Information Criteria
------------------------------------

Akaike       -6.4807
Bayes        -6.4735
Shibata      -6.4807
Hannan-Quinn -6.4782

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic   p-value
Lag[1]                      5.548 1.850e-02
Lag[2*(p+q)+(p+q)-1][5]     6.437 1.263e-05
Lag[4*(p+q)+(p+q)-1][9]     7.191 1.108e-01
d.o.f=2
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                      1.103  0.2935
Lag[2*(p+q)+(p+q)-1][5]     1.497  0.7407
Lag[4*(p+q)+(p+q)-1][9]     1.956  0.9102
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]   0.01965 0.500 2.000  0.8885
ARCH Lag[5]   0.17504 1.440 1.667  0.9713
ARCH Lag[7]   0.53718 2.315 1.543  0.9749

Nyblom stability test
------------------------------------
```

```
Joint Statistic:  174.6662
Individual Statistics:
mu      0.2090
ar1     0.1488
ma1     0.1057
omega  21.3780
alpha1  0.1345
beta1   0.1130

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:              1.49 1.68 2.12
Individual Statistic:         0.35 0.47 0.75

Sign Bias Test
------------------------------------
                    t-value      prob sig
Sign Bias           0.4298 6.674e-01
Negative Sign Bias  2.9478 3.214e-03 ***
Positive Sign Bias  2.3929 1.675e-02  **
Joint Effect       28.9794 2.262e-06 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     179.0    4.951e-28
2    30     188.1    3.195e-25
3    40     218.6    7.737e-27
4    50     227.6    7.927e-25


Elapsed time : 1.295898
```

3.1 Fit Diagnostics

The summary method for the `uGARCHfit` object provides the parameters and their standard errors (and a robust version), together with a variety of tests which can also be called individually. The robust standard errors are based on the method of White (1982) which produces asymptotically valid confidence intervals by calculating the covariance (V) of the parameters (θ) as:

$$\hat{V} = (-A)^{-1} B (-A)^{-1} \tag{96}$$

where,

$$\begin{aligned} A &= L'' \left(\hat{\theta} \right) \\ B &= \sum_{i=1}^{n} g_i \left(x_i \left| \hat{\theta} \right. \right)^T g_i \left(x_i \left| \hat{\theta} \right. \right) \end{aligned} \tag{97}$$

which is the Hessian and covariance of the scores at the optimum. The robust standard errors are the square roots of the diagonal of V.
The `inforcriteria` method on a fitted or filtered object returns the Akaike (AIC), Bayesian

(BIC), Hannan-Quinn (HQIC) and Shibata (SIC) information criteria to enable model selection by penalizing overfitting at different rates. Formally, they may be defined as:

$$\begin{aligned} AIC &= \frac{-2LL}{N} + \frac{2m}{N} \\ BIC &= \frac{-2LL}{N} + \frac{m\log_e(N)}{N} \\ HQIC &= \frac{-2LL}{N} + \frac{(2m\log_e(\log_e(N)))}{N} \\ SIC &= \frac{-2LL}{N} + \log_e\left(\frac{(N+2m)}{N}\right) \end{aligned} \tag{98}$$

were any parameters fixed during estimation are excluded from the calculation. Since version 1.3-1, the Q-statistics and ARCH-LM test have been replaced with the Weighted Ljung-Box and ARCH-LM statistics of Fisher and Gallagher (2012) which better account for the distribution of the statistics of the values from the estimated models. The ARCH-LM test is now a weighted portmanteau test for testing the null hypothesis of adequately fitted ARCH process, whilst the Ljung-Box is another portmanteau test with null the adequacy of the ARMA fit. The `signbias` calculates the Sign Bias Test of Engle and Ng (1993), and is also displayed in the summary. This tests the presence of leverage effects in the standardized residuals (to capture possible misspecification of the GARCH model), by regressing the squared standardized residuals on lagged negative and positive shocks as follows:

$$\hat{z}_t^2 = c_0 + c_1 I_{\hat{\varepsilon}_{t-1}<0} + c_2 I_{\hat{\varepsilon}_{t-1}<0}\hat{\varepsilon}_{t-1} + c_3 I_{\hat{\varepsilon}_{t-1}\geqslant 0}\hat{\varepsilon}_{t-1} + u_t \tag{99}$$

where I is the indicator function and $\hat{\varepsilon}_t$ the estimated residuals from the GARCH process. The Null Hypotheses are $H_0 : c_i = 0$ (for $i = 1, 2, 3$), and that jointly $H_0 : c_1 = c_2 = c_3 = 0$. As can be inferred from the summary of the previous fit, there is significant Negative and Positive reaction to shocks. Using instead a model such as the apARCH would likely alleviate these effects.

The `gof` calculates the chi-squared goodness of fit test, which compares the empirical distribution of the standardized residuals with the theoretical ones from the chosen density. The implementation is based on the test of Palm (1996) which adjusts the tests in the presence on non-i.i.d. observations by reclassifying the standardized residuals not according to their value (as in the standard test), but instead on their magnitude, calculating the probability of observing a value smaller than the standardized residual, which should be identically standard uniform distributed. The function must take 2 arguments, the fitted object as well as the number of bins to classify the values. In the summary to the fit, a choice of $(20, 30, 40, 50)$ bins is used, and from the summary of the previous example it is clear that the Normal distribution does not adequately capture the empirical distribution based on this test.

The `nymblom` test calculates the parameter stability test of Nyblom (1989), as well as the joint test. Critical values against which to compare the results are displayed, but this is not available for the joint test in the case of more than 20 parameters.

Finally, some informative plots can be drawn either interactively(which = 'ask'), individually (which = 1:12) else all at once (which = 'all') as in Figure 2.

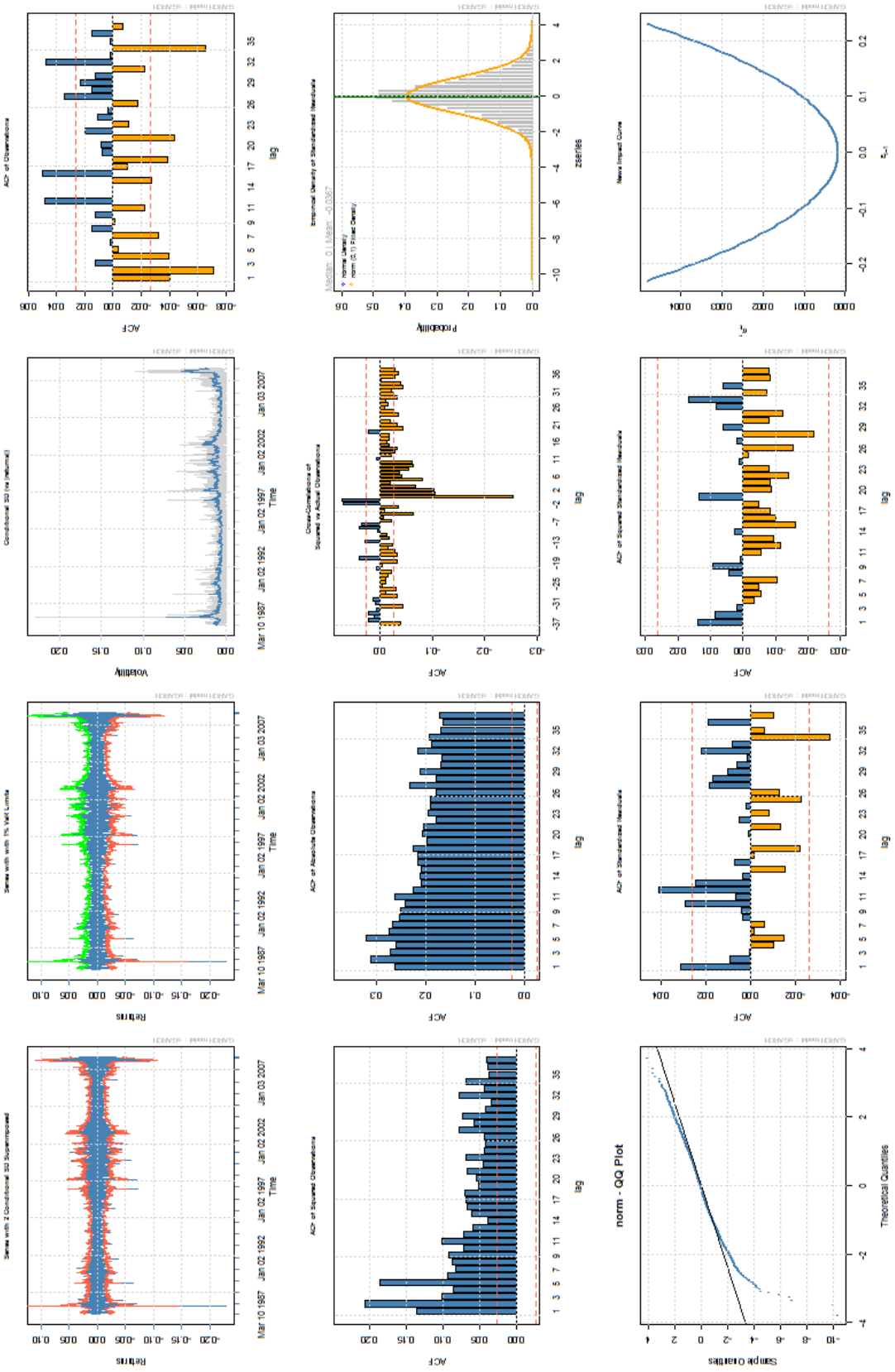


Figure 2: uGARCHfit Plots

4 Filtering

Sometimes it is desirable to simply filter a set of data with a predefined set of parameters. This may for example be the case when new data has arrived and one might not wish to re-fit. The `ugarchfilter` method does exactly that, taking a `uGARCHspec` object with fixed parameters. Setting fixed or starting parameters on the GARCH spec object may be done either through the `ugarchspec` function when it is called via the fixed.pars arguments to the function, else by using the `setfixed<-` method on the spec object. The example which follows explains how:

```
> data(sp500ret)
> spec = ugarchspec(variance.model = list(model = "apARCH"), distribution.model = "std")
> setfixed(spec) <- list(mu = 0.01, ma1 = 0.2, ar1 = 0.5, omega = 1e-05,
+     alpha1 = 0.03, beta1 = 0.9, gamma1 = 0.01, delta = 1, shape = 5)
> filt = ugarchfilter(spec = spec, data = sp500ret)
> show(filt)

*----------------------------------*
*          GARCH Model Filter      *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model        : apARCH(1,1)
Mean Model        : ARFIMA(1,0,1)
Distribution        : std

Filter Parameters
-------------------------------------

mu     1e-02
ar1    5e-01
ma1    2e-01
omega  1e-05
alpha1 3e-02
beta1  9e-01
gamma1 1e-02
delta  1e+00
shape  5e+00

LogLikelihood : 5627.392

Information Criteria
-------------------------------------

Akaike       -2.0378
Bayes        -2.0378
Shibata      -2.0378
Hannan-Quinn -2.0378

Weighted Ljung-Box Test on Standardized Residuals
-------------------------------------
```

```
                        statistic p-value
Lag[1]                       1178       0
Lag[2*(p+q)+(p+q)-1][5]      1212       0
Lag[4*(p+q)+(p+q)-1][9]      1217       0
d.o.f=2
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                      170.4       0
Lag[2*(p+q)+(p+q)-1][5]     173.8       0
Lag[4*(p+q)+(p+q)-1][9]     175.8       0
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]     3.915 0.500 2.000 0.04785
ARCH Lag[5]     4.207 1.440 1.667 0.15612
ARCH Lag[7]     5.237 2.315 1.543 0.20163


Sign Bias Test
------------------------------------
                   t-value      prob sig
Sign Bias           8.3144 1.150e-16 ***
Negative Sign Bias  5.6999 1.261e-08 ***
Positive Sign Bias  0.5621 5.741e-01
Joint Effect       95.8328 1.223e-20 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     86296            0
2    30    128891            0
3    40    169014            0
4    50    207506            0
```

The returned object is of class `uGARCHfilter` and shares many of the methods as the `uGARCHfit` class. Additional arguments to the function are explained in the documentation. Note that the information criteria shown here are based on zero estimated parameters (they are all fixed), and the same goes for the *infocriteria* method on a `uGARCHfilter` object.

5 Forecasting and the GARCH Bootstrap

There are 2 types of forecasts available with the package. A rolling method, whereby consecutive 1-ahead forecasts are created based on the out.sample option set in the fitting routine, and an unconditional method for n>1 ahead forecasts. (and it is also possible to combine the 2 creating

a rather complicated object). In the latter case, it is also possible to make use of the GARCH bootstrap, described in Pascual et al. (2006) and implemented in the function `ugarchboot`, with the added innovation of an optional extra step of fitting either a kernel or semi-parametric density (SPD) to the standardized residuals prior to sampling in order to provide for (possibly) more robustness in the presence of limited data. To understand what the GARCH bootstrap does, consider that there are two main sources of uncertainty about n.ahead forecasting from GARCH models: that arising from the form of the predictive density and that due to parameter uncertainty. The bootstrap method in the `rugarch` package is based on resampling standardized residuals from the empirical distribution of the fitted model to generate future realizations of the series and sigma. Two methods are implemented: one takes into account parameter uncertainty by building a simulated distribution of the parameters through simulation and refitting, and one which only considers distributional uncertainty and hence avoids the expensive and lengthy parameter distribution estimation. In the latter case, prediction intervals for the 1-ahead sigma forecast will not be available since only the parameter uncertainty is relevant in GARCH type models in this case. The following example provides for a brief look at the partial method, but the interested reader should consult the more comprehensive examples in the inst folder of the package.

```
> data(sp500ret)
> spec = ugarchspec(variance.model=list(model="csGARCH"), distribution="std")
> fit = ugarchfit(spec, sp500ret)
> bootp = ugarchboot(fit, method = c("Partial", "Full")[1],
+ n.ahead = 500, n.bootpred = 500)
> show(bootp)

*----------------------------------*
*     GARCH Bootstrap Forecast     *
*----------------------------------*
Model : csGARCH
n.ahead : 500
Bootstrap method:  partial
Date (T[0]): 2009-01-30

Series (summary):
          min      q.25      mean     q.75      max forecast[analytic]
t+1  -0.10855 -0.013343  0.000668 0.016285 0.090454           0.001944
t+2  -0.11365 -0.010796  0.001632 0.015721 0.085783           0.001707
t+3  -0.28139 -0.013203 -0.000378 0.015496 0.082250           0.001512
t+4  -0.10459 -0.014830  0.000346 0.015602 0.109223           0.001352
t+5  -0.21915 -0.012494  0.001196 0.016627 0.098003           0.001220
t+6  -0.11029 -0.012119  0.001008 0.015000 0.083469           0.001112
t+7  -0.22818 -0.013280  0.000398 0.015250 0.094184           0.001023
t+8  -0.25722 -0.014854 -0.001401 0.016074 0.088067           0.000949
t+9  -0.34629 -0.017681 -0.004484 0.012847 0.154058           0.000889
t+10 -0.11328 -0.013566  0.000957 0.018291 0.140734           0.000840
.....................

Sigma (summary):
          min    q0.25     mean    q0.75      max forecast[analytic]
t+1  0.026387 0.026387 0.026387 0.026387 0.026387           0.026387
t+2  0.025518 0.025564 0.026345 0.026493 0.038492           0.026577
```

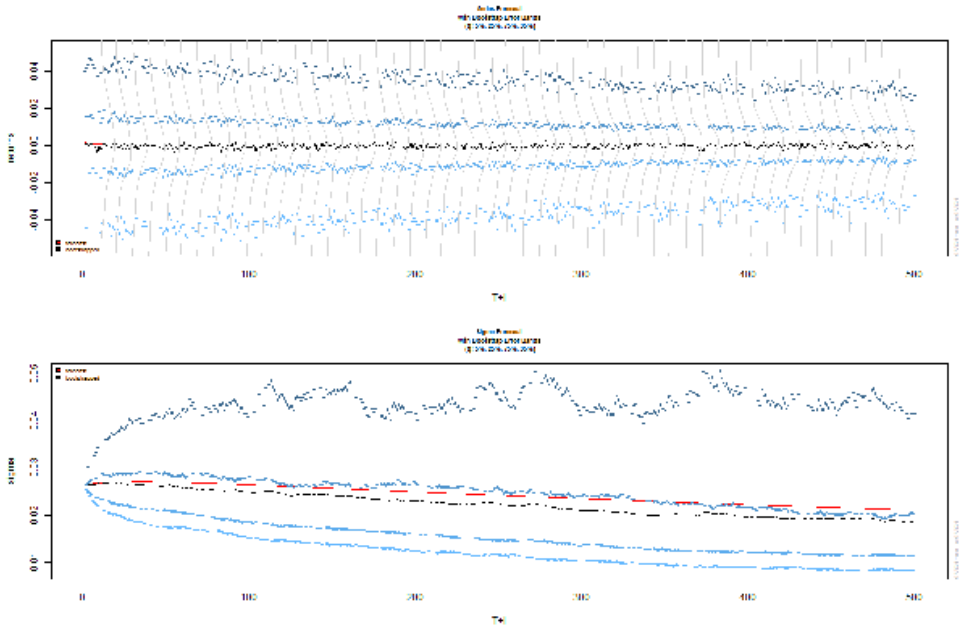


Figure 3: GARCH Bootstrap Forecast Plots

```
t+3  0.024698 0.025021 0.026332 0.026768 0.039903          0.026614
t+4  0.023925 0.024682 0.026440 0.027087 0.077525          0.026649
t+5  0.023259 0.024456 0.026474 0.027507 0.074919          0.026682
t+6  0.022622 0.024173 0.026498 0.027687 0.072876          0.026714
t+7  0.021988 0.023855 0.026463 0.027700 0.069719          0.026744
t+8  0.021503 0.023684 0.026424 0.027785 0.070750          0.026772
t+9  0.021044 0.023677 0.026690 0.028065 0.071725          0.026799
t+10 0.020560 0.023589 0.027050 0.028363 0.095243          0.026824
.....................
```

The 'recipe' for the full GARCH bootstrap is summarized below:

1. Extract the standardized residuals from the estimated object. If it is a specification with fixed parameters, first filter using the supplied dataset and then extract the standardized residuals from the filtered object.

2. Sample n.bootfit sets of size N (original dataset less any out of sample periods) from either the raw standardized residuals, using the spd or kernel based methods.

3. Simulate n.bootfit paths of size N, using as innovations the sampled standardized residuals. The simulation is initiated with the last values from the dataset at point N (T_0 in simulation time).

4. The n.bootfit simulated series are then estimated with the same specification used by the originally supplied object in order to generate a set of coefficients representing the parameter uncertainty.

5. Filter the original dataset with the n.bootfit set of estimated coefficients.

6. Use the last values of the filtered conditional sigma (and if it is the csGARCH model, then also the permanent component q) and residuals from the previous step to initialize a

new simulation with horizon n.ahead and m.sim=n.bootpred, using again the standardized residuals sampled as in step 2 and the new set of estimated coefficients. The simulation now contains uncertainty about the conditional n-ahead density as well as parameter uncertainty.

6 Simulation

Simulation may be carried out either directly on a fitted object (`ugarchsim`) else on a GARCH spec with fixed parameters (`ugarchpath`). The `ugarchsim` method takes the following arguments:

```
> args(ugarchsim)

function (fit, n.sim = 1000, n.start = 0, m.sim = 1, startMethod = c("unconditional",
    "sample"), presigma = NA, prereturns = NA, preresiduals = NA,
    rseed = NA, custom.dist = list(name = NA, distfit = NA),
    mexsimdata = NULL, vexsimdata = NULL, ...)
```

where the n.sim indicates the length of the simulation while m.sim the number of independent simulations. For reasons of speed, when n.sim is large relative to m.sim, the simulation code is executed in C, while for large m.sim a special purpose C++ code (using Rcpp and RcppArmadillo) is used which was found to lead to significant speed increase. Key to replicating results is the rseed argument which is used to pass a user seed to initialize the random number generator, else one will be assigned by the program. In any case, the returned object, of class `uGARCHsim` (or `uGARCHpath`) contains a slot with the seed(s) used.

7 Rolling Estimation

The `ugarchroll` method allows to perform a rolling estimation and forecasting of a model/dataset combination, optionally returning the VaR at specified levels. More importantly, it returns the distributional forecast parameters necessary to calculate any required measure on the forecasted density. The following example illustrates the use of the method where use is also made of the parallel functionality and run on 10 cores.[16] Figure 4 is generated by calling the plot function on the returned `uGARCHroll` object. Additional methods, and more importantly extractor functions can be found in the documentation. **Note that only n.ahead=1 is allowed at present (more complicated rolling forecasts can be created by the user with the ugarchfit and ugarchforecast functions).** Finally, there is a new method called *resume* which allows resumption of estimation of an object which had non-converged windows, optionally supplying a different solver and solver control combination.

```
> data(sp500ret)
> library(parallel)
> cl = makePSOCKcluster(10)
> spec = ugarchspec(variance.model = list(model = "eGARCH"), distribution.model = "jsu")
> roll = ugarchroll(spec, sp500ret, n.start = 1000, refit.every = 100,
refit.window = "moving", solver = "hybrid", calculate.VaR = TRUE,
VaR.alpha = c(0.01, 0.05), cluster = cl, keep.coef = TRUE)
>show(roll)
>stopCluster(cl)
```

[16] **Since version 1.0-14 the parallel functionality is based on the paralllel package and it is upto the user to initialize a cluster object and pass it to the function, and then terminate it once it is no longer required. Eventually, this approach to the parallel usage will filter through to all the functions in rugarch and rmgarch.**

```
*-------------------------------------*
*              GARCH Roll             *
*-------------------------------------*
No.Refits       : 46
Refit Horizon   : 100
No.Forecasts    : 4523
GARCH Model     : eGARCH(1,1)
Distribution    : jsu

Forecast Density:
              Mu  Sigma    Skew  Shape Shape(GIG) Realized
1991-02-21 4e-04 0.0102 -0.2586 1.5065          0  -0.0005
1991-02-22 2e-04 0.0099 -0.2586 1.5065          0   0.0019
1991-02-25 4e-04 0.0095 -0.2586 1.5065          0   0.0044
1991-02-26 3e-04 0.0093 -0.2586 1.5065          0  -0.0122
1991-02-27 1e-04 0.0101 -0.2586 1.5065          0   0.0135
1991-02-28 7e-04 0.0099 -0.2586 1.5065          0  -0.0018

.........................
                Mu  Sigma  Skew Shape Shape(GIG) Realized
2009-01-23  0.0015 0.0259 -0.87 2.133          0   0.0054
2009-01-26  0.0005 0.0243 -0.87 2.133          0   0.0055
2009-01-27 -0.0002 0.0228 -0.87 2.133          0   0.0109
2009-01-28 -0.0011 0.0212 -0.87 2.133          0   0.0330
2009-01-29 -0.0039 0.0191 -0.87 2.133          0  -0.0337
2009-01-30  0.0009 0.0220 -0.87 2.133          0  -0.0231

Elapsed: 12.97949 secs

> report(roll, type = "VaR", VaR.alpha = 0.05, conf.level = 0.95)

VaR Backtest Report
===========================================
Model:                  eGARCH-jsu
Backtest Length:        4523
===========================================
alpha:                  5%
Expected Exceed:        226.2
Actual VaR Exceed:      253
Actual %:               5.6%

Unconditional Coverage (Kupiec)
Null-Hypothesis:        Correct Exceedances
LR.uc Statistic:        0
LR.uc Critical:         3.841
LR.uc p-value:          1
Reject Null:            NO

Conditional Coverage (Christoffersen)
Null-Hypothesis:        Correct Exceedances and
                        Independence of Failures
```

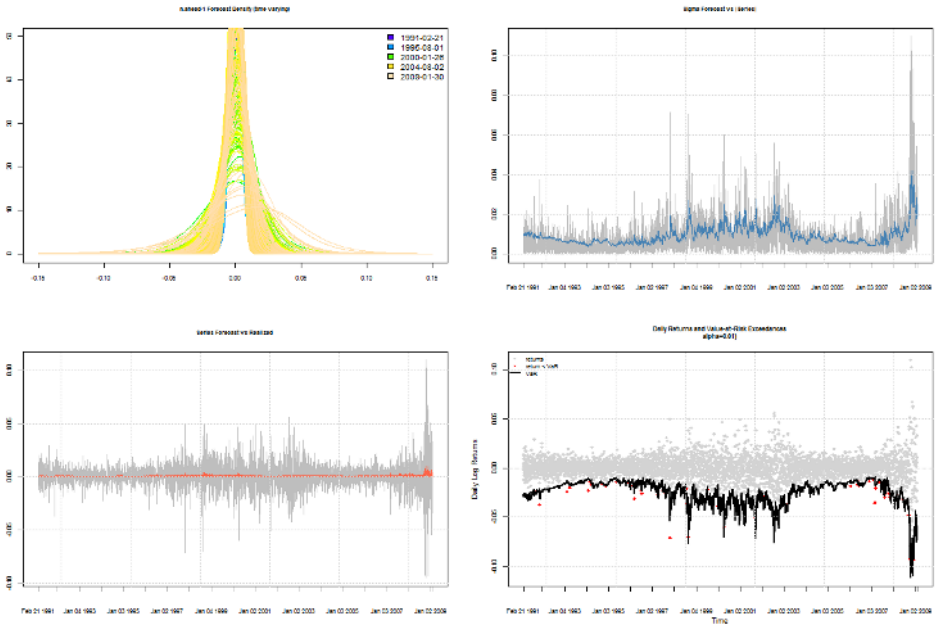


Figure 4: eGARCH Rolling Forecast Plots

```
LR.cc Statistic:          0
LR.cc Critical:           5.991
LR.cc p-value:            1
Reject Null:              NO
```

8 Simulated Parameter Distribution and RMSE

It is sometimes instructive to be able to investigate the underlying density of the estimated parameters under different models. The `ugarchdistribution` method performs a monte carlo experiment by simulating and fitting a model multiple times and for different 'window' sizes. This allows to obtain some insight on the consistency of the parameter estimates as the data window increases by looking at the rate of decrease of the Root Mean Squared Error and whether we have $\sqrt{N}$ consistency. This is a computationally expensive exercise and as such should only be undertaken in the presence of ample computing power and RAM. As in other functions, parallel functionality is enabled if available. The example which follows illustrates an instance of this test on one model and one set of parameters. Figures 5 and 6 complete this example.

```
> spec = ugarchspec(variance.model = list(model = "gjrGARCH"),
+     distribution.model = "ged")
> print(persistence(pars = unlist(list(mu = 0.001, ar1 = 0.4, ma1 = -0.1,
+     omega = 1e-06, alpha1 = 0.05, beta1 = 0.9, gamma1 = 0.05,
+     shape = 1.5)), distribution = "ged", model = "gjrGARCH"))

persistence
      0.975

> library(parallel)
> cl = makePSOCKcluster(10)
> setfixed(spec) <- list(mu = 0.001, ar1 = 0.4, ma1 = -0.1, omega = 1e-06,
+     alpha1 = 0.05, beta1 = 0.9, gamma1 = 0.05, shape = 1.5)
> dist = ugarchdistribution(fitORspec = spec, n.sim = 2000, n.start = 1,
+     m.sim = 100, recursive = TRUE, recursive.length = 6000, recursive.window = 1000,
+     rseed = 1066, solver = "solnp", solver.control = list(trace = 0),
+     cluster = cl)
> stopCluster(cl)
> show(dist)

*-----------------------------------*
*    GARCH Parameter Distribution    *
*-----------------------------------*
Model : gjrGARCH
No. Paths (m.sim) : 100
Length of Paths (n.sim) : 2000
Recursive : TRUE
Recursive Length : 6000
Recursive Window : 1000

Coefficients: True vs Simulation Mean (Window-n)
                    mu     ar1       ma1      omega   alpha1   beta1   gamma1  shape
true-coef   0.00100000 0.40000 -0.100000 1.0000e-06 0.050000 0.90000 0.050000 1.5000
window-2000 0.00097099 0.39679 -0.097585 1.0411e-06 0.046161 0.90177 0.051709 1.4945
window-3000 0.00101520 0.38907 -0.089494 9.9344e-07 0.046741 0.90192 0.052902 1.4957
window-4000 0.00099876 0.39329 -0.094523 9.9556e-07 0.049385 0.90179 0.047515 1.4924
window-5000 0.00099491 0.40223 -0.103767 9.7545e-07 0.048651 0.90220 0.049442 1.4900
window-6000 0.00098876 0.39540 -0.096411 9.8975e-07 0.048753 0.90096 0.050907 1.4901
```

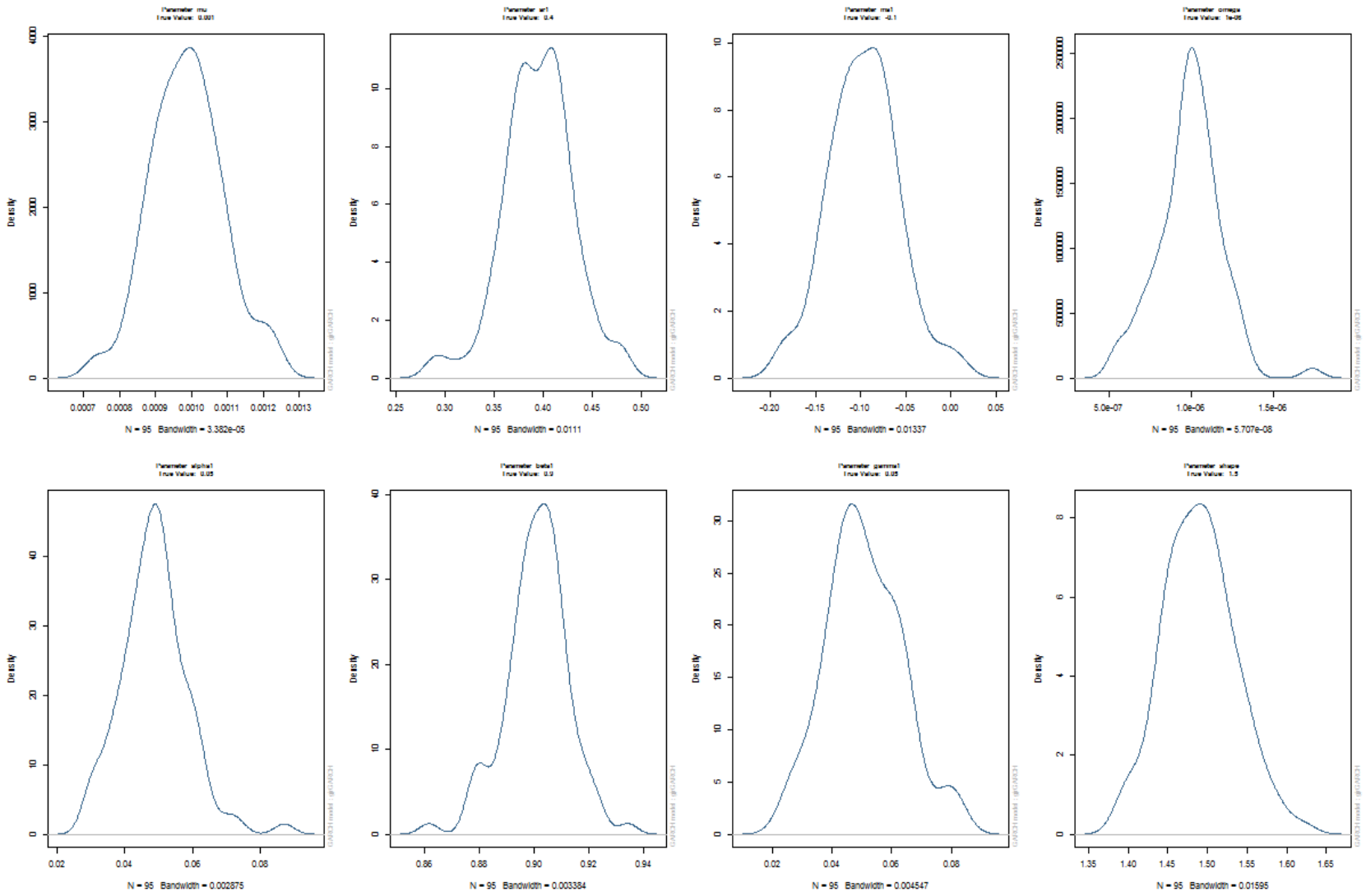


Figure 5: Simulated Parameter Density

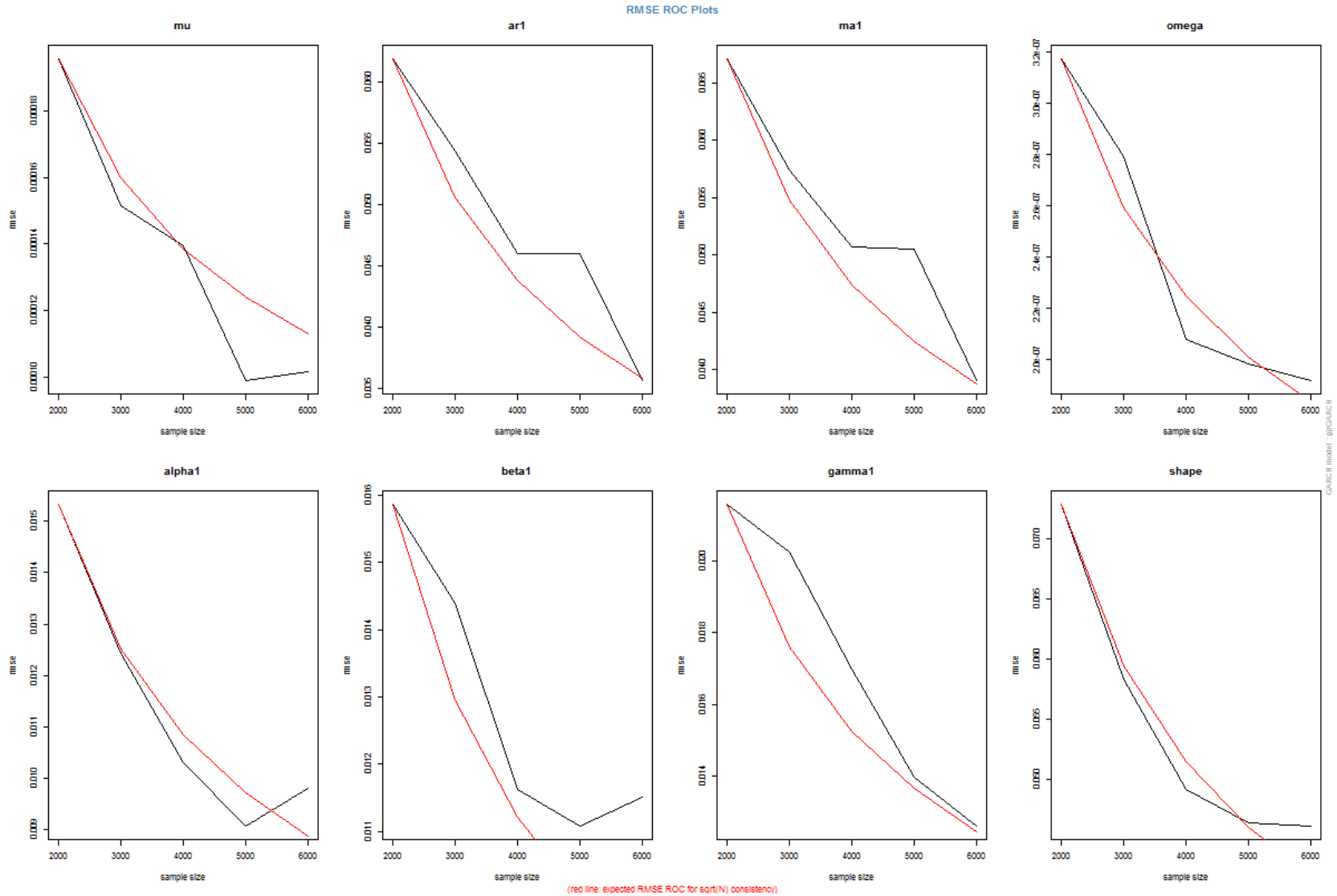
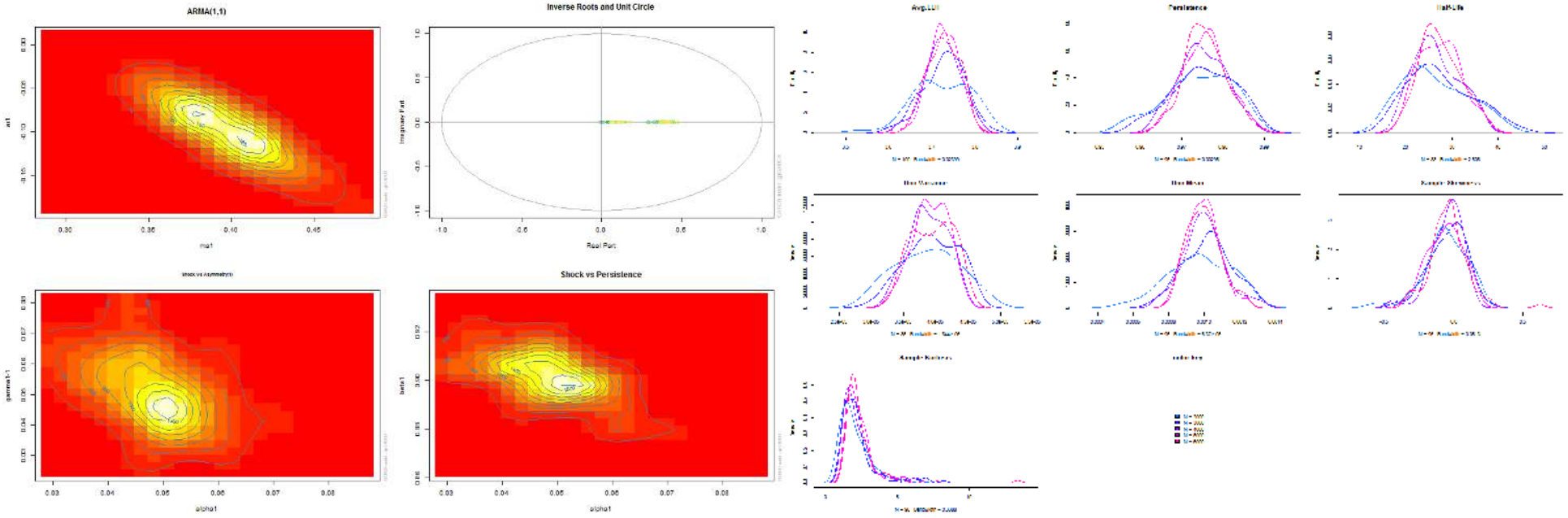


Figure 6: RMSE Rate of Change



(a) Bivariate Parameter Plots

(b) GARCH Stat Plots

Figure 7: GARCH Simulated Parameters Density

9 The ARFIMAX Model with constant variance

The `rugarch` package implements an additional set of methods and classes, mirroring those of the GARCH specification, for modelling ARFIMAX processes with constant variance via Maximum Likelihood. With the exception of plots, the functionality is very similar to that covered so far for GARCH methods. The main functions are `arfimaspec`, `arfimafit`, `arfimaforecast`, `arfimasim`, `arfimapath`, `arfimadistirbution` and `arfimaroll`. The usual extractor, inference and summary methods are replicated for all the ARFIMA classes and the user should consult the documentation for further details.

10 Mispecification and Other Tests

Apart from the usual tests presented in the summary to the fit object, a number of other interesting and useful tests are uniquely implemented in the `rugarch` package, and described in this section.

10.1 The GMM Orthogonality Test

The GMM type moment (orthogonality) tests of Hansen (1982) have been applied to test the adequacy of model in a variety of setups. Under a correctly specified model, certain population moment conditions should be satisfied and hold in the sample using the standardized residuals. The moment conditions can be tested both individually using a t-test or jointly using a Wald test. Formally, the following moment conditions are tested:

$$\begin{array}{lcl} M_1 & E\left[z_t\right] & = 0 \\ M_2 & E\left[z_t^2-1\right] & = 0 \\ M_3 & E\left[z_t^3\right] & = 0 \\ M_4 & E\left[z_t^4-3\right] & = 0 \\ Q_2 & E\left[\left(z_t^2-1\right)\left(z_{t-j}^2-1\right)\right] & = 0 \\ Q_3 & E\left[\left(z_t^3\right)\left(z_{t-j}^3\right)\right] & = 0 \\ Q_4 & E\left[\left(z_t^4-3\right)\left(z_{t-j}^4-3\right)\right] & = 0 \end{array} \tag{100}$$

where $j = 1\ldots,p$ is the lag (defaults to 4 in the function), M_1 to M_4 denotes the individual moment conditions (t-test), and Q_2 to Q_4 the joint conditional moment conditions (variance, skewness and kurtosis) which are distributed χ^2 with p d.o.f. All the moment conditions can also be tested jointly using a Wald test distributed χ^2 with $3p+4$ d.o.f. The test is implemented under the name: `GMMTest`.

10.2 Parametric and Non-Parametric Density Tests

A novel method to analyze how well a conditional density fits the underlying data is through the probability integral transformation (PIT) discussed in Rosenblatt (1952) and defined as:

$$x_t = \int_{-\infty}^{y_t} \hat{f}\left(u\right) du = \hat{F}\left(y_t\right) \tag{101}$$

which transforms the data y_t, using the estimated distribution $\hat{F}$ into i.i.d. $U(0,1)$ under the correctly specified model. Based on this transformation, Tay et al. (1998) provide for a visual assessment test, while Berkowitz (2001) provides a more formal test, implemented in the package

under the name `BerkowitzTest`. Because of the difficulty in testing a $U(0,1)$ sequence, the PIT data is transformed into $N(0,1)$ by Berkowitz using the normal quantile function, and tested using a Lagrange Multiplier (LM) test for any residual autocorrelation given a specified number of lags. In addition, a tail test based on the censored Normal is also provided, under the Null that the standardized tail data has mean zero and unit variance. More recently, Hong and Li (2005) introduced a nonparametric portmanteau test, building on the work of Ait-Sahalia (1996), which tests the joint hypothesis of i.i.d *AND* $U(0,1)$ for the sequence x_t. As noted by the authors, testing x_t using a standard goodness-of-fit test (such as the Kolmogorov-Smirnov) would only check the $U(0,1)$ assumption under i.i.d. and not the joint assumption of $U(0,1)$ and i.i.d. Their approach is to compare a kernel estimator $\hat{g}_j(x_1,x_2)$ for the joint density $g_j(x_1,x_2)$ of the pair $\{x_t, x_{t-j}\}$ (where j is the lag order) with unity, the product of two $U(0,1)$ densities. Given a sample size n and lag order $j>0$, their joint density estimator is:

$$\hat{g}_j(x_1,x_2) \equiv (n-j)^{-1} \sum_{t=j+1}^{n} K_h\left(x_1, \hat{X}_t\right) K_h\left(x_2, \hat{X}_{t-j}\right) \tag{102}$$

where $\hat{X}_t = X_t\left(\hat{\theta}\right)$, and $\hat{\theta}$ is a $\sqrt{n}$ consistent estimator of θ_0. The function K_h is a boundary modified kernel defined as:

$$K_h(x,y) \equiv \begin{cases} h^{-1}k\left(\frac{x-y}{h}\right) \Big/ \int_{-(x/h)}^{1} k(u)\,du, & \text{if} x \in [0,h), \\ h^{-1}k\left(\frac{x-y}{h}\right), & \text{if} x \in [h,1-h], \\ h^{-1}k\left(\frac{x-y}{h}\right) \Big/ \int_{-1}^{(1-x)/h} k(u)\,du, & \text{if} x \in (1-h,1], \end{cases} \tag{103}$$

where $h \equiv h(n)$ is a bandwidth such that $h \to 0$ as $n \to \infty$, and the kernel $k(.)$ is a pre-specified symmetric probability density, which is implemented as suggested by the authors using a quartic kernel,

$$k(u) = \frac{15}{16}\left(1-u^2\right)^2 \mathbf{1}(|u| \leq 1), \tag{104}$$

where $\mathbf{1}(.)$ is the indicator function. Their portmanteau test statistic is defined as:

$$\hat{W}(p) \equiv p^{-1/2} \sum_{j=1}^{p} \hat{Q}(j), \tag{105}$$

where

$$\hat{Q}(j) \equiv \left[(n-j)\, h\hat{M}(j) - A_h^0\right] \Big/ V_0^{1/2}, \tag{106}$$

and

$$\hat{M}(j) \equiv \int_0^1 \int_0^1 \left[\hat{g}_j(x_1,x_2) - 1\right]^2 dx_1 dx_2. \tag{107}$$

The centering and scaling factors A_h^0 and V_0 are defined as:

$$\begin{aligned} A_h^0 &\equiv \left[\left(h^{-1}-2\right) \int_{-1}^{1} k^2(u)\,du + 2\int_0^1 \int_{-1}^{b} k_b^2(u)\,du db\right]^2 - 1 \\ V_0 &\equiv 2\left[\int_{-1}^{1}\left[\int_{-1}^{1} k(u+v)\,k(v)\,dv\right]^2 du\right]^2 \end{aligned} \tag{108}$$

where,

$$k_b(.) \equiv k(.) \Big/ \int_{-1}^{b} k(v)\,dv. \tag{109}$$

Under the correct model specification, the authors show that $\hat{W}(p) \to N(0,1)$ in distribution. Because negative values of the test statistic only occur under the Null Hypothesis of a correctly

specified model, the authors indicate that only upper tail critical values need be considered. The test is quite robust to model misspecification as parameter uncertainty has no impact on the asymptotic distribution of the test statistic as long as the parameters are $\sqrt{n}$ consistent. Finally, in order to explore possible causes of misspecification when the statistic rejects a model, the authors develop the following test statistic:

$$M(m,l) \equiv \left[\sum_{j=1}^{n-1} w^2(j/p)(n-j)\hat{\rho}_{ml}^2(j) - \sum_{j=1}^{n-1} w^2(j/p)\right] \Bigg/ \left[2\sum_{j=1}^{n-2} w^4(j/p)\right]^{1/2} \tag{110}$$

where $\hat{\rho}_{ml}(j)$ is the sample cross-correlation between $\hat{X}_t^m$ and $\hat{X}_{t-|j|}^l$, and $w(.)$ is a weighting function of lag order j, and as suggested by the authors implemented as the Bartlett kernel. As in the $\hat{W}(p)$ statistic, the asymptotic distribution of $M(m,l)$ is $N(0,1)$ and upper critical values should be considered. As an experiment, Table 1 considers the cost of fitting a GARCH-Normal model when the true model is GARCH-Student, using the `HLTest` on simulated data using the `ugarchsim` function. The results are clear: At low levels of the shape parameter ν, representing a very high excess kurtosis, the model is overwhelmingly rejected by the test, and as that parameter increases to the point where the Student approximates the Normal, the rejections begin to reverse. Also of interest, but not surprising, the strength of the rejection is somewhat weaker for smaller datasets ($N = 500, 1000$). For example, in the case of using only 500 data points and a shape parameter of 4.1 (representing an excess kurtosis of 60!), 5% of the time, in this simulation, the test failed to reject the GARCH-Normal.

Table 1: GARCH-Student: Misspecification Exercise.

	ν[4.1]	**ν[4.5]**	**ν[5]**	**ν[5.5]**	**ν[6]**	**ν[6.5]**	**ν[7]**	**ν[7.5]**	**ν[8]**	**ν[10]**	**ν[15]**	**ν[20]**	**ν[25]**	**ν[30]**
N_{500}														
stat	10.10	6.70	5.08	4.07	2.64	2.22	1.47	1.46	1.05	0.19	-0.34	-0.36	-0.54	-0.71
%reject	*95*	*89*	*82*	*76*	*59*	*54*	*42*	*41*	*34*	*22*	*12*	*13*	*6*	*8*
N_{1000}														
stat	18.54	13.46	9.46	7.64	6.16	5.14	4.17	2.95	3.03	1.31	0.28	-0.15	-0.48	-0.47
%reject	*100*	*100*	*98*	*97*	*90*	*86*	*79*	*64*	*69*	*39*	*24*	*11*	*7*	*12*
N_{2000}														
stat	32.99	26.46	19.41	15.53	12.41	10.35	7.76	6.79	5.79	3.20	0.87	0.09	0.03	-0.21
%reject	*100*	*100*	*100*	*100*	*100*	*99*	*95*	*94*	*92*	*71*	*32*	*22*	*22*	*16*
N_{3000}														
stat	47.87	37.03	27.38	21.67	17.85	14.22	11.46	9.73	7.99	5.12	1.60	0.35	0.10	-0.09
%reject	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*99*	*96*	*85*	*46*	*27*	*22*	*15*

Note: The table presents the average test statistic of Hong and Li (2005) and number of rejections at the 95% confidence level for fitting a GARCH(1,1)-Normal model to a GARCH(1,1)-Student model for different values of the shape parameter ν, and sample size (N). For each sample of size N, 250 simulated series were created from a GARCH student model with parameters $(\mu, \omega, \alpha, \beta) = (5.2334e-04, 4.3655e-06, 5.898e-02, 9.2348e-01)$, and ν in the range of $[4.1, 30]$, and fitted using a GARCH(1,1)-Normal model. The standardized residuals of the fitted model where then transformed via the normal distribution function into $U(0,1)$ series and evaluated using the test of Hong and Li (2005).

10.3 Directional Accuracy Tests

High and significant Directional Accuracy (DA) could imply either an ability to predict the sign of the mean forecast or could merely be the result of volatility dependence in the absence of mean predictability as argued by Christoffersen and Diebold (2006). In either case, the function `DACTest` provides 2 tests for determining the significance of sign predictability and mean predictability. The Directional Accuracy (DA) Test of Pesaran and Timmermann (1992) and the Excess Profitability (EP) Test of Anatolyev and Gerko (2005), both of which are Hausmann type tests. The EP test statistic is formally defined as:

$$EP = \frac{A_T - B_T}{\sqrt{\hat{V}_{EP}}} \tag{111}$$

with,

$$\begin{aligned} A_T &= \frac{1}{T}\sum_t r_t \\ B_T &= \left(\frac{1}{T}\sum_t \operatorname{sgn}(\hat{y}_t)\right)\left(\frac{1}{T}\sum_t y_t\right) \end{aligned} \tag{112}$$

with $\hat{y}$ being the forecast of y and $r_t = \operatorname{sgn}(\hat{y}_t)(y_t)$. According to the authors of the test, the estimated variance of EP, $\hat{V}_{EP}$ may be estimated as:

$$\hat{V}_{EP} = \frac{4}{T^2}\hat{p}_{\hat{y}}\left(1-\hat{p}_{\hat{y}}\right)\sum\left(y_t-\bar{y}\right)^2 \tag{113}$$

where $\hat{p}_{\hat{y}} = \frac{1}{2}\left(1+\frac{1}{T}\sum_t \operatorname{sgn}(\hat{y}_t)\right)$. The EP statistic is asymptotically distributed as $N(0,1)$. For the DA test the interested reader can consult the relevant literature for more details.

10.4 VaR and Expected Shortfall Tests

The unconditional coverage, or proportion of failures, test of Kupiec (1995) allows to test whether the observed frequency of VaR exceedances is consistent with the expected exceedances, given the chosen quantile and a confidence level. Under the Null hypothesis of a correctly specified model, the number of exceedances X follows a binomial distribution. A probability below a given significance level leads to a rejection of the Null hypothesis. The test is usually conducted as a likelihood ratio test, with the statistic taking the form,

$$LR_{uc} = -2\ln\left(\frac{(1-p)^{N-X}p^X}{\left(1-\frac{X}{N}\right)^{N-X}\left(\frac{X}{N}\right)^X}\right) \tag{114}$$

where p is the probability of an exceedance for the chosen confidence level and N is the sample size. Under the Null the test statistic is asymptotically distributed as a χ^2 with 1 degree of freedom. The test does not consider any potential violation of the assumption of the independence of the number of exceedances. The conditional coverage test of Christoffersen et al. (2001) corrects this by jointly testing the frequency as well as the independence of exceedances, assuming that the VaR violation is modelled with a first order Markov chain. The test is a likelihood ratio, asymptotically distributed as χ^2 with 2 degrees of freedom, where the Null is that the conditional and unconditional coverage are equal to α. The test is implemented under the name `VaRTest`.
In a further paper, Christoffersen and Pelletier (2004) considers the duration between VaR violations as a stronger test of the adequacy of a risk model. The duration of time between VaR violations (no-hits) should ideally be independent and not cluster. Under the Null hypothesis of a correctly specified risk model, the no-hit duration should have no memory. Since the only continuous distribution which is memory free is the exponential, the test can conducted on any distribution which embeds the exponential as a restricted case, and a likelihood ratio test then conducted to see whether the restriction holds. Following Christoffersen and Pelletier (2004), the Weibull distribution is used with parameter $b = 1$ representing the case of the exponential. The test is implemented under the name `VaRDurTest`.
Because VaR tests deal with the occurrences of hits, they are by definition rather crude measures to compare how well one model has done versus another, particularly with short data sets. The expected shortfall test of McNeil and Frey (2000) measures the mean of the shortfall violations which should be zero under the Null of a correctly specified risk model. The test is implemented in the function `ESTest` which also provides for the option of bootstrapping the distribution of the p-value, hence avoiding any strong assumptions about the underlying distribution of the excess

shortfall residuals.
Finally, it is understood that these tests are applied to out-of-sample forecasts and NOT insample, for which no correction to the tests have been made to account for parameter uncertainty.

10.5 The Model Confidence Set Test

The Model Confidence Set (MCS) procedure of Hansen et al. (2011) (henceforth HLN) provides for a ranking of models given some penalized measure on their relative loss function difference. Define a set M^0 as the original model comparison set with i models and t the time index, and let $L_{i,t}(.)$ be some user specified loss function. The ranking of the models is then based on the relative difference of the pairwise loss function, $d_{ij,t}$:

$$d_{ij,t} = L_{i,t} - L_{j,t} \; \forall i, j \in M^0, \tag{115}$$

where it is assumed that $\mu_{ij} \equiv E[d_{ij,t}]$ is finite and does not depend on t, and that i is preferred to j if $\mu_{ij} \leq 0$. The set of models which can then be described as superior is defined as:

$$M^* \equiv \left\{ i \in M^0 : \mu_{ij} \leqslant 0 \; \forall j \in M^0 \right\}. \tag{116}$$

The determination of M^* is done through a sequence of significance tests with models found to be significantly inferior eliminated from the set. The null hypothesis takes the form:

$$H_{0,M} : \mu_{ij} = 0 \; \forall i, j \in M \tag{117}$$

with $M \in M^0$, and tested using an equivalence test δ_M. In case of rejection of the null, an elimination rule e_M is then used to identify the model to be removed from the set and the procedure repeated until all inferior models are eliminated. Given a significance level a, the models which are not eliminated are deemed the model confidence set $\hat{M}^*_{1-a}$ with the key theorem of the test, given a set of assumptions on δ_M and e_M, being that $\lim_{n \to +\infty} P\left(M^* \subset \hat{M}^*_{1-a}\right) \geqslant 1 - a$. The actual studentized measure used to compare models is defined as:

$$\frac{\hat{d}_i}{\sqrt{\text{var}\left(\hat{d}_i\right)}} \tag{118}$$

with $\hat{d}_i$ derived as:

$$\begin{aligned} \hat{d}_{ij} &\equiv \frac{1}{N} \sum_{t=1}^{N} d_{ij,t} \\ \hat{d}_i &\equiv \frac{1}{m-1} \sum_{j \in M} \hat{d}_{ij} \end{aligned} \tag{119}$$

where $\hat{d}_{ij}$ measures the relative performance between models, and $\hat{d}_i$ the measures the relative performance of model i to the average of all the models in M, and the variance of $\hat{d}_i$, $var(\hat{d}_i)$ may be derived by use of the bootstrap. The statistic then used to eliminate inferior models is the range statistic[17] and defined as:

$$T_R = \max_{i,j \in M} \frac{\left|\hat{d}_i\right|}{\sqrt{\text{var}\left(\hat{d}_i\right)}}. \tag{120}$$

The asymptotic distribution of T_R, and hence the p-values reported, is obtained via the bootstrap procedure, the validity of which is established in HLN.

[17] Other options are available such as the semi-quadratic statistic which is also returned by the package function.

11 Future Development

Any future extensions will likely be 'add-on' packages released in the bitbucket code repository of the package.

12 FAQs and Guidelines

This section provides for answers to some Frequently Asked Questions (Q) as well as Guidelines (G) for the use of the `rugarch` package.

Q: What is the format of the data accepted by the package?

Since version 1.01-3, only xts data is supported, or data which can be coerced to this. This is meant to simplify maintenance of the package whilst at the same time use what is a very popular and widely adopted 'format/wrapper'. Some of the extractor functions will now also return an xts formatted object.

Q: Where can I found out about changes to the package?

Read the changelog for version specific changes.

Q: Does the package support parallel computation?

Yes. Since version 1.0-14, `rugarch` makes exclusive use of the `parallel` package for all parallel computations. Certain functions take as input a user supplied cluster object (created by calling *parallel::makeCluster*), which is then used for parallel computations. It is then up to the user to terminate that cluster once it is no longer needed. Allowing a cluster object to be provided in this way was deemed the most flexible approach to the parallel computation problem across different architectures and resources.

Q: My model does not converge, what can I do?

There are several avenues to consider here. The package offers 4 different solvers, namely 'solnp', 'gosolnp', 'nlminb' and 'L-BGFS-U' (from optim). Each solver has its own merits, and control parameters which may, and should be passed, via the solver.control list in the fitting routines, depending on your particular data. For problems where neither 'solnp' nor 'nlminb' seem to work, try the 'gosolnp' solver which does a search of the parameter space based on a truncated normal distribution for the parameters and then initializes multiple restarts of the 'solnp' solver based on the best identified candidates. The numbers of randomly generated parameters (n.sim) and solver restarts (n.restarts) can be passed via the solver.control list. Additionally, in the fit.control list of the fitting routines, the option to perform scaling of the data prior to fitting usually helps, although it is not available under some setups. Finally, consider the amount of data you are using for modelling GARCH processes, which leads to another FAQ below.

Q: How much data should I use to model GARCH processes with confidence?
The distribution of the parameters varies by model, and is left to the reader to consult relevant literature on this. However, using 100 data points to try and fit a model is unlikely to be a sound approach as you are unlikely to get very efficient parameter estimates. The `rugarch` package does provide a method (`ugarchdistribution`) for simulating from a pre-specified model, data of different sizes, fitting the model to the data, and inferring the distribution of the parameters as

well as the RMSE rate of change as the data length increases. This is a very computationally expensive way to examine the distribution of the parameters (but the only way in the non-Bayesian world), and as such should be used with care and in the presence of ample computing power.

Q: Where can one find more examples?

The package has a folder called 'rugarch.tests' which contains many tests which I use for debugging and checking. The files in the folder should be 'sourced' by the user, and the 'runtests.R' file contains some wrapper functions which describe what each test does, and optionally runs chosen tests. The output will be a combination of text files (.txt) and figures (either .eps or .png) in an output directory which the user can define in the arguments to the wrapper function 'rugarch.runtests'. It is quite instructive to read and understand what each test is doing prior to running it. There are also online examples which you can find by typing rugarch in a search engine.

Q: What to do if I find an error or have questions related to the package?

Please use the R-SIG-Finance mailing list to post your questions. If you do mail me directly, do consider carefully your email, debug information you submit, and correct email etiquette (i.e. do not send me a 1 MB .csv file of your data and at no time send me an Excel file).

References

K. Aas and I.H. Haff. The generalized hyperbolic skew student's t-distribution. *Journal of Financial Econometrics*, 4(2):275–309, 2006.

Y. Ait-Sahalia. Testing continuous-time models of the spot interest rate. *Review of Financial Studies*, 9(2):385–426, 1996.

S. Anatolyev and A. Gerko. A trading approach to testing for predictability. *Journal of Business and Economic Statistics*, 23(4):455–461, 2005.

Torben G Andersen and Tim Bollerslev. Intraday periodicity and volatility persistence in financial markets. *Journal of Empirical Finance*, 4(2):115–158, 1997.

Richard T Baillie, Tim Bollerslev, and Hans Ole Mikkelsen. Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 74(1):3–30, 1996.

J. Berkowitz. Testing density forecasts, with applications to risk management. *Journal of Business and Economic Statistics*, 19(4):465–474, 2001.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.

T. Bollerslev. A conditionally heteroskedastic time series model for speculative prices and rates of return. *The Review of Economics and Statistics*, 69(3):542–547, 1987.

Tim Bollerslev and Hans Ole Mikkelsen. Modeling and pricing long memory in stock market volatility. *Journal of econometrics*, 73(1):151–184, 1996.

G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time series analysis: Forecasting and control*. Prentice Hall, 1994.

P. Christoffersen and D. Pelletier. Backtesting value-at-risk: A duration-based approach. *Journal of Financial Econometrics*, 2(1):84–108, 2004.

P. Christoffersen, J. Hahn, and A. Inoue. Testing and comparing value-at-risk measures. *Journal of Empirical Finance*, 8(3):325–342, 2001.

P.F. Christoffersen and F.X. Diebold. Financial asset returns, direction-of-change forecasting, and volatility dynamics. *Management Science*, 52(8):1273–1287, 2006.

Christian Conrad and Berthold R Haag. Inequality constraints in the fractionally integrated garch model. *Journal of Financial Econometrics*, 4(3):413–449, 2006.

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.

R.F. Engle and J. Mezrich. Grappling with garch. *Risk*, 8(9):112–117, 1995.

R.F. Engle and V.K. Ng. Measuring and testing the impact of news on volatility. *Journal of Finance*, 48(5):1749–1778, 1993.

R.F. Engle, D.M. Lilien, and R.P. Robins. Estimating time varying risk premia in the term structure: The arch-m model. *Econometrica: Journal of the Econometric Society*, 55(2):391–407, 1987.

Robert F. Engle and Magdalena E. Sokalska. Forecasting intraday volatility in the us equity market. multiplicative component garch. *Journal of Financial Econometrics*, 10(1):54–83, 2012. doi: 10.1093/jjfinec/nbr005. URL `http://jfec.oxfordjournals.org/content/10/1/54.abstract`.

C. Fernandez and M.F. Steel. On bayesian modeling of fat tails and skewness. *Journal of the American Statistical Association*, 93(441):359–371, 1998.

J.T. Ferreira and M.F. Steel. A constructive representation of univariate skewed distributions. *Journal of the American Statistical Association*, 101(474):823–829, 2006.

Thomas J Fisher and Colin M Gallagher. New weighted portmanteau statistics for time series goodness of fit testing. *Journal of the American Statistical Association*, 107(498):777–787, 2012.

J.Y.P. Geweke. Modelling the persistence of conditional variances: A comment. *Econometric Reviews*, 5:57–61, 1986.

A. Ghalanos and S. Theussl. *Rsolnp: General non-linear optimization using augmented Lagrange multiplier method.*, 1.11 edition, 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.

B.E. Hansen. Autoregressive conditional density estimation. *International Economic Review*, 35: 705–730, 1994.

L.P. Hansen. Large sample properties of generalized method of moments estimators. *Econometrica*, 50(4):1029–1054, 1982.

Peter R Hansen, Asger Lunde, and James M Nason. The model confidence set. *Econometrica*, 79(2):453–497, 2011.

Peter Reinhard Hansen, Zhuo Huang, and Howard Howan Shek. Realized garch: a joint model for returns and realized measures of volatility. *Journal of Applied Econometrics*, 27(6):877–906, 2012.

L. Hentschel. All in the family nesting symmetric and asymmetric garch models. *Journal of Financial Economics*, 39(1):71–104, 1995.

M.L. Higgins, A.K. Bera, et al. A class of nonlinear arch models. *International Economic Review*, 33(1):137–158, 1992.

Y. Hong and H. Li. Nonparametric specification testing for continuous-time models with applications to term structure of interest rates. *Review of Financial Studies*, 18(1):37–84, 2005.

Menelaos Karanasos, Zacharias Psaradakis, and Martin Sola. On the autocorrelation properties of long-memory garch processes. *Journal of Time Series Analysis*, 25(2):265–282, 2004.

P.H. Kupiec. Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2):73–84, 1995. ISSN 1074-1240.

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.

A.J. McNeil and R. Frey. Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach. *Journal of Empirical Finance*, 7(3-4):271–300, 2000.

D.B. Nelson. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, 59(2):347–70, 1991.

J. Nyblom. Testing for the constancy of parameters over time. *Journal of the American Statistical Association*, 84(405):223–230, 1989.

F.C. Palm. Garch models of volatility. *Handbook of Statistics*, 14:209–240, 1996.

S.G. Pantula. Comment: Modelling the persistence of conditional variances. *Econometric Reviews*, 5(1):71–74, 1986.

L. Pascual, J. Romo, and E. Ruiz. Bootstrap prediction for returns and volatilities in garch models. *Computational Statistics and Data Analysis*, 50(9):2293–2312, 2006.

M.H. Pesaran and A. Timmermann. A simple nonparametric test of predictive performance. *Journal of Business and Economic Statistics*, 10(4):461–465, 1992.

K. Prause. *The generalized hyperbolic model: Estimation, financial derivatives, and risk measures*. PhD thesis, University of Freiburg, 1999.

R.A. Rigby and D.M. Stasinopoulos. Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3):507–554, 2005.

M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952.

G.W. Schwert. Stock volatility and the crash of '87. *Review of Financial Studies*, 3(1):77, 1990.

David Scott and Fiona Grimson. *SkewHyperbolic: The Skew Hyperbolic Student t-Distribution*, version 0.3-3 edition. URL `http://cran.r-project.org/web/packages/SkewHyperbolic/index.html`.

D.M. Stasinopoulos, B.A. Rigby, and C. Akantziliotou. *gamlss: Generalized additive models for location, scale and shape.*, 1.11 edition, 2009.

A. Tay, F.X. Diebold, and T.A. Gunther. Evaluating density forecasts: With applications to financial risk management. *International Economic Review*, 39(4):863–883, 1998.

S.J. Taylor. *Modelling financial time series*. Wiley, 1986.

H. White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, 50(1):1–25, 1982.

Y. Ye. *Interior point algorithms: Theory and analysis*, volume 44. Wiley-Interscience, 1997.

J.M. Zakoian. Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, 18(5):931–955, 1994.